Barriers to Identity-Based Encryption in Generic Group-Action Models

Sikhar Patranabis

IBM Research India
sikhar.patranabis@ibm.com

Abstract. Identity-based encryption (IBE) is required to support a superpolynomial-size identity space using only polynomial-size public parameters. This compression requirement underlies known barriers to IBE in the generic (pairing-free) group model. Cryptographic group actions, motivated prominently by isogeny-based cryptography, support many advanced primitives beyond public-key encryption. However, no generic construction of IBE from the basic group-action interface is known. Existing generic-group barriers do not resolve this question as they do not translate to generic group-action models.

We establish the first tight barriers to IBE in two variants of the generic group-action model. In the type-safe generic group-action model, we prove that an IBE scheme with negligible correctness error and n non-origin action-set element-handles in its public parameters can support at most n identities, even for a weak notion of fully selective security. We show that this bound holds irrespective of the number of action-set element-handles in identity secret keys. We further establish that this bound is tight by demonstrating a matching construction.

We then extend this tight characterization to a natural class of IBE schemes in the random-representation generic group-action model (including against representation-sensitive adversaries) under conditions ensuring that identity keys retain no hidden action-set encodings and discarded private encodings leave no residual information. We give a natural syntactic condition for the latter property. For actions additionally equipped with an inversion-like twist, we obtain a more general upper bound (including the same tight bound for odd-order actions).

Our results provide the first formal evidence that IBE is likely to require additional algebraic structure or non-black-box techniques beyond the generic group-action interface. More broadly, we provide a new framework for proving lower bounds in generic group-action models.

Table of Contents

Barriers to Identity-Based Encryption in Generic Group-Action Models 1
Sikhar Patranabis

1 Introduction 3
1.1 Our Contributions 5
2 Technical Overview 7
2.1 Two Variants of the Generic Group Action Model 7
2.2 Impossibility of Restricted TS-GGAM-IBE 8
2.3 The Tight TS-GGAM-IBE Impossibility 13
2.4 Extending the Impossibility from TS-GGAM to RSR-GGAM 15
2.5 Extending the Impossibility to Twist-Augmented Group Actions 19
3 Preliminaries 20
3.1 Identity-Based Encryption 20
3.2 Regular Group Actions 21
3.3 The Random Set Representation Generic Group Action Model 21
3.4 The Type-Safe Generic Group Action Model 22
4 Identity-Based Encryption in Generic Group-Action Models 23
4.1 IBE in the Type-Safe GGAM 23
4.2 IBE in the RSR-GGAM 25
5 Impossibility of Restricted TS-GGAM-IBE 28
5.1 Anchored Transcripts and Gain Graphs 28
5.2 Sampling Relative-Action Relations 31
5.3 Two-Sided Transfer of Generic-Action Transcripts 33
5.4 Offline Simulation and Alternative Setup Worlds 35
5.5 Impossibility of Restricted TS-GGAM-IBE 38
6 Tight Impossibility of TS-GGAM-IBE 43
6.1 Eliminating Action-Set Handles from Secret Keys 43
6.2 General Impossibility of TS-GGAM-IBE 46
6.3 Tightness of the Public-Handle Bound 49
7 A Tight Impossibility Result for Sanitizable RSR-GGAM-IBE 51
7.1 Lazy Random-String Representations 51
7.2 Label-Synchronous Transcript Transfer 52
7.3 Forgetting Private Labels 53
7.4 Symmetry-Preserving Offline Candidate Selection 54
7.5 Impossibility of Restricted Composably Label-Sanitizable RSR-GGAM-IBE 55
7.6 Tightness of the Public-Label Bound 59
7.7 A Tight Impossibility in the Presence of Twists 60
7.8 A Syntactic Criterion for Label Sanitizability 62

1 Introduction

Identity-based Encryption. Identity-based encryption (IBE), introduced by Shamir [Sha84], allows an arbitrary identifier (such as an email address) to serve as an encryption address. A trusted authority generates compact public parameters and a master secret key; anyone can encrypt to an identity using the public parameters alone, while the authority can generate the corresponding decryption key on demand. Besides eliminating the need to distribute and certify an independent public key for every user, IBE is a fundamental building block for richer forms of access-controlled encryption.

The first realizations of IBE were given by Boneh and Franklin [BF01] from bilinear pairings and by Cocks [Coc01] from quadratic residuosity. These were followed by standard-model pairing-based constructions [BB04a, BB04b, Wat05] and, in the post-quantum setting, by lattice-based constructions [GPV08, CHKP10, ABB10] and more recent constructions from a variant of learning parity with noise [DGHM18]. Historically, obtaining IBE from a new algebraic assumption has often preceded constructions of more expressive primitives, including hierarchical and attribute-based encryption [SW05, GPSW06, AFV11, GVW13], broadcast encryption [BGW05, AY20, Wee22], and traitor tracing [BSW06, GKW18]. Understanding the kinds of algebraic structures that suffice for IBE is therefore a useful test of their broader cryptographic power.

Barriers to IBE. Non-trivial IBE demands a special form of compression: a polynomial-size public parameter must support encryption for a superpolynomial (typically, exponential) identity space. More concretely, the key-generation algorithm holding the master secret key must efficiently generate the decryption key for any identity in this identity space on demand.[1] This requirement underlies several known black-box barriers for IBE. Boneh, Papakonstantinou, Rackoff, Vahlis, and Waters [BPR+08] ruled out fully black-box constructions of IBE from trapdoor permutations and CCA-secure public-key encryption. Papakonstantinou, Rackoff, and Vahlis [PRV12] subsequently showed that IBE cannot be constructed generically from a DDH-hard group. Schul-Ganz and Segev [SGS21] made this barrier tight: if the public parameters contain n generic group elements, then a secure generic-group IBE can support at most n identities.

These results do not rule out constructions of IBE from the Diffie-Hellman assumption that use the underlying group in a non-black-box manner. In particular, starting with the seminal work of Döttling and Garg [DG17b], a long line of works [CDG+17, DG17a, BLSV18, DGHM18] have bypassed the generic-group barrier through constructions that make non-black-box usage of the underlying group via garbled circuits.

Isogenies and Cryptographic Group Actions. Isogeny-based assumptions constitute an important foundation for candidate constructions of post-quantum cryptography. Isogeny-based cryptographic constructions from class-group actions such as CSIDH [CLM+18] and its derivatives [BKV19, DFK+23, PR23, DEF+25, BCE+25, DEIV26] are naturally captured by a finite group G acting on a set X:

$$\star\colon G \times X \longrightarrow X, \qquad g \star (h \star x) = (g \circ h) \star x.$$

In the principal isogeny instantiations, an ideal class group acts on a set of suitably oriented supersingular elliptic curves. The acting group can be manipulated explicitly, but the acted-on set has no efficiently available group law. In particular, given $x, y \in X$, recovering the displacement $g \in G$ satisfying $y = g \star x$ is conjectured to be hard.

Cryptographic group actions [BY91, Cou06, ADMP20] abstract precisely this asymmetric interface. They support basic primitives such as public-key encryption and non-interactive key exchange, as well as practically efficient signatures and threshold cryptographic protocols [RS06, CLM+18, DG19, BKV19, DM20]. A long line of subsequent works [ADMP20, BKP20, AMPS21, AEK+22, KLLQ23, LR24] has shown how to build advanced cryptographic primitives from cryptographic group actions, such as smooth projective hashing, dual-mode encryption, oblivious transfer, KDM-secure encryption, password-authenticated key exchange, blind and ring signatures, and updatable encryption. Thus, despite having an algebraic interface that is markedly weaker than that of an ordinary (cyclic) group, group actions seem capable of supporting many rich cryptographic primitives that are ostensibly stronger than plain public-key encryption.

[1] IBE with polynomial-sized identity space follows trivially from public-key encryption.

IBE from Group Actions? A natural question is whether one can build IBE from cryptographic group actions. Gürler and Struck [GS25] study natural isogeny-based approaches to realize IBE by separating identity-key derivation from public-key encryption. Their analysis suggests that a black-box construction of IBE from a generic cryptographic group action may be difficult, but does not establish a formal separation.

Very recently, Agrawal, Basso, and Patranabis [ABP26] constructed the first secure IBE from isogeny-based assumptions. Their construction proceeds via batch encryption [BLSV18, DGHM18] and the CDH-with-mismatched-torsion assumption, justified in the Algebraic Isogeny Model [ABR26]. Their construction crucially exploits concrete structure of supersingular *non-oriented* curves. Moreover, following the framework of [DG17b, CDG+17, DG17a, BLSV18, DGHM18], their construction makes garbling-based, non-black-box use of the underlying isogeny-based primitives. Hence, while their work establishes the feasibility of constructing IBE from isogeny-based assumptions, it does not resolve the question of whether the group-action abstraction can be used generically to realize IBE.

Resolving the latter question is important for efficiency: constructions using only the basic action interface are generally simpler and more amenable to efficient instantiation than those relying on non-black-box techniques. Conversely, an impossibility result would indicate that such techniques are likely unavoidable. This motivates our central question:

Can identity-based encryption be realized generically [2] *using only the interface of a cryptographic group action?*

Generic Group-Action Models. The generic group action model (GGAM) provides a natural setting for this question. Following Duman et al. [DHK+23], the acting group G is represented explicitly, whereas elements of X are hidden behind generic representations and can be manipulated only through the action interface. In this paper, we consider two variants of the GGAM:

- In the random-set-representation model, which we call the *RSR-GGAM*, elements of the action-set X are encoded by uniformly random strings. These strings hide their semantic action-set elements, but an algorithm may inspect, hash, copy, or otherwise compute on their literal representations.
- In the *type-safe GGAM*, or TS-GGAM, action-set elements are instead represented by opaque typed handles that may only be copied, compared for equality, passed through typed interfaces, or supplied to the action oracle.

One may view the distinction between the TS-GGAM and the RSR-GGAM as the group-action analogue of the distinction between random-representation and Maurer-style generic-group models [Mau05, Zha22]. We note here that type-safe GGAM formulations have previously been used, for example, in proving black-box lower bounds for group-action-based signatures [BGZ23], and as a foundation for algebraic-action and explicit-isogeny models for analyzing isogeny-based oblivious transfer protocols [OZ24].

GGM vs GGAM. The known generic-group impossibility results for IBE [PRV12, SGS21] do not carry over to (either formulation of) the GGAM. This is because the GGM and the GGAM expose fundamentally different algebraic interfaces. A generic group algorithm can combine any two available generic group elements, invert them, and form arbitrary linear combinations. The GGM-based IBE lower bounds from [PRV12, SGS21] crucially exploit this structure by collecting linear relations among public, secret-key, and ciphertext elements. In a group action, there is no analogous operation on X: an explicit group element may translate one action-set element, but two independently supplied action-set elements cannot be combined or subtracted. Consequently, the known GGM-based lower bounds for IBE are not inherited by the GGAM.

Technically, the linear-relation spaces underlying the GGM-based attacks in [PRV12, SGS21] have no direct GGAM counterpart. In the GGAM, an element obtained by repeatedly acting on a public element remains tied to that same starting point. Comparing two such derived elements may reveal how their starting points are related, but the action interface does not let the algorithm combine those starting points into new elements as generic-group operations would. Consequently, the attack must learn and exploit a different kind of algebraic information.

[2] Here, "generically" means that elements of the acted-on set are accessed only through the generic action interface; we do not claim a general fully black-box separation outside the models considered below.

In the random-representation model (RSR-GGAM), there is a further obstacle: algorithms can inspect the strings representing elements and remember information about them. Two executions with the same algebraic relationships can therefore behave differently when their labels differ. Establishing an IBE barrier requires a new way to track the independent algebraic information revealed by action queries, together with a way to control information about random labels that persists across successive computations.

1.1 Our Contributions

We establish, for the first time, tight upper bounds on the number of identities that an IBE scheme can support when it accesses a group action only through generic interfaces. Our impossibility applies to arbitrary IBE schemes in the TS-GGAM and to a natural class of IBE schemes in the RSR-GGAM. In both settings, we establish the following: if the public parameters contain at most n_{pp} non-origin action-set handles or labels, then the scheme cannot securely support more than n_{pp} (efficiently indexable) identities. In other words, polynomially many public action-set representations cannot support the superpolynomial identity space required of non-trivial IBE. As is customary for information-theoretic generic-model lower bounds, our separation proof considers adversaries that are *pseudoefficient*: they make polynomially many generic-oracle queries, although their local computation may be unbounded.

Impossibility of IBE in the TS-GGAM. Our first result applies to arbitrary TS-GGAM-IBE schemes. We prove the following: if the public parameters of an IBE scheme with identity-space ID_λ contain at most $n_{\mathsf{pp}}(\lambda)$ distinct non-origin action-set handles, then full selective security and negligible correctness imply $|\mathsf{ID}_\lambda| \leq n_{\mathsf{pp}}(\lambda)$ for all sufficiently large security parameters (see Informal Theorem 2.7 and Theorem 7).

Proof Techniques. We develop a novel *anchored-transcript* framework for reasoning about generic group-action computations, which serves as a structural GGAM analogue of the linear-relation spaces exploited in GGM-based lower bounds. The key intuition underlying this framework is that a generic group action provides only limited algebraic "compression." Starting from the action-set elements contained in the public parameters of a GGAM-based scheme, an algorithm may generate arbitrarily many further elements by applying known group actions. However, these derived elements remain tied to their starting points, and comparisons among them can reveal only a bounded amount of independent information. In particular, the amount of independent algebraic information available to the computation is bounded by the number of non-origin action-set elements in the public parameters.

We exploit this observation for our TS-GGAM-IBE attack. In particular, suppose the scheme supports one more identity than the number of action-set elements in the public parameter. By the pigeonhole principle, among these identities, there must be one whose encryption-decryption computation reveals no genuinely new independent information beyond that already learned from the preceding identities. The adversary can then search offline for an alternative setup and a candidate secret key for this identity that are consistent with everything learned so far. Our main technical argument shows that, as long as neither execution reveals new information, this candidate behaves identically in the alternative and real worlds. Correctness in the alternative world therefore gives the adversary a key that can be used to attack the real challenge ciphertext.

A Two-Stage Attack. For simplicity of exposition, we initially carry out this attack against a restricted class of TS-GGAM-IBE schemes such that the secret identity keys contain no action-set handles (see Informal Theorem 2.5 and Theorem 1). We then give a compiler that removes handles from arbitrary identity keys (see Informal Theorem 2.6 and Theorem 6). The compiler preserves security exactly and loses only a controllable amount of correctness. We show that, while secret-key handles affect the compiler's sampling complexity, the final identity bound established by the attack depends only on the number of public handles in the scheme's public parameters.

From Opaque Handles to Random Representations. Our second result lifts the semantic core of the TS-GGAM attack to a natural class of RSR-GGAM-IBE schemes. Recall that, in the RSR-GGAM, elements of the action-set X are encoded by uniformly random strings. This creates two difficulties absent from the type-safe setting: an algorithm may branch on the literal bits in an action-set label string, and labels allocated in one invocation may remain hidden and influence later invocations. We address these difficulties through a

novel, representation-aware framework that ensures that the random labels used in the real and alternative executions can be consistently coordinated, without allowing hidden information about previously generated labels to distinguish the two executions. This allows us to transfer the alternative-world argument from the type-safe setting to the random-representation model.

The resulting attack applies to RSR-GGAM-IBE schemes that are *restricted key-label-oblivious* and *composably label-sanitizable*. Informally, the first condition prevents information about unavailable private label strings from being hidden inside the public parameters or extracted keys. The second requires that, when temporary labels are discarded between computations, the state kept for later computations does not retain clues about their particular string values. Even after fixing all retained information, every otherwise available assignment of strings to the forgotten labels must remain equally likely.

We show that, assuming a standard notion of *label slack* (i.e., assuming the random label space is sufficiently large that labels of unseen action-set elements cannot be guessed except with negligible probability), every fully selective-secure scheme in this class with n_{pp} designated non-origin public labels supports at most n_{pp} identities. See Informal Theorem 2.12, as well as Theorem 9 and Corollary 12 for more details.

We also give a simple syntactic condition that makes our sanitizability requirement easy to verify. At a high level, it ensures that random labels which are no longer part of the visible state do not leave behind hidden information that can influence later computations. Together with our separate key-label-obliviousness requirement, this captures a broad and natural class of RSR-GGAM schemes, including natural random-label implementations of type-safe computations. Extending our impossibility result to completely unrestricted representation-sensitive RSR-GGAM schemes remains an interesting open question.

Tightness. We prove that both identity bounds in terms of public handles or labels are exact. For every polynomially bounded n, there is a fully selectively secure IBE supporting n identities with at most n non-origin public handles in the TS-GGAM (see Informal Theorem 2.8, as well as Theorem 8 and Corollary 7), and with at most n designated non-origin public labels in the RSR-GGAM (see Informal Theorem 2.13, as well as formal Theorem 10). The construction consists of n independent copies of elementary group-action public-key encryption, one per identity. We prove its security in the TS-GGAM against arbitrary adversaries, and its security in the RSR-GGAM against arbitrary representation-sensitive adversaries. This establishes that one action-set object per identity in the IBE public parameters is both necessary and sufficient in both formulations of the GGAM considered in this work.

Twist-Augmented Actions. Finally, we extend our RSR-GGAM-based separation framework to generic group actions equipped with a twist operation $\mathsf{tw}(a \star \widetilde{x}) = a^{-1} \star \widetilde{x}$. Concretely, we enrich our framework to include orientation information for action-set elements. Under the same key-label-obliviousness, sanitizability, and label-slack hypotheses as above, we demonstrate an unconditional upper bound of $2n$ identities for an IBE scheme with n non-origin labels in its public parameters. Further, when the acting group is bijective (in particular, for odd-order groups), we establish the same tight n-identity bound (see Informal Theorem 2.14, as well as Theorem 11, Corollary 13, and Corollary 14).

Taken together, our results identify a precise limitation of the generic group-action interface in the type-safe and sanitizable random-representation settings covered here. Although this interface supports public-key encryption, it admits only limited compression: the number of independent identity-indexed decryption capabilities is bounded linearly by the number of action-set elements explicitly present in the public parameters. Accordingly, within these models, non-trivial IBE must exploit structure beyond the basic generic action interface. More broadly, our results provide formal evidence that additional isogeny-specific or non-black-box structure are likely to be necessary for realizing IBE, as is indeed used by the recent positive construction [ABP26].

A Framework for GGAM Lower Bounds. Beyond the specific IBE barriers established here, our techniques provide a broader methodology for lower bounds in generic group-action models. The central principle is *anchored provenance*: an algorithm may generate many action-set elements, yet each element still traces back to a single input element; as a result, only a bounded number of independent algebraic relationships can be learned among these starting points. Our framework shows how to exploit the resulting gap between potentially many derived elements and comparatively few independent relations to establish an impossibility result. This viewpoint provides a general template for proving limitations when the desired

cryptographic functionality requires more independent algebraic capabilities than the generic group-action interface can support.

2 Technical Overview

The complete proofs of our results require substantial technical bookkeeping and are too long to include within the main-body page limit. We therefore present a self-contained technical overview that states our results and develops the principal ideas behind their proofs, while deferring the full model formulations, intermediate statements, and rigorous proofs to the supplementary material. The overview is intended to suffice for understanding both our contributions and the techniques underlying them; the supplementary material is needed only for the complete formal treatment. Conversely, readers who prefer to begin directly with the rigorously formal description of our techniques may skip this overview and read the supplementary material independently.

Notation. We summarize here some notations used throughout the overview. We write $\lambda \in \mathbb{N}$ for the security parameter. For $k \in \mathbb{N}$, let $[k] = \{1, \ldots, k\}$. We write $x \leftarrow S$ for uniform sampling from a finite set S. We denote by $\mathsf{GA}_\lambda = (G_\lambda, X_\lambda, \star_\lambda, \widetilde{x}_\lambda)$ a finite abelian regular group action. We write the operation of G_λ multiplicatively as $\circ$ and usually omit the subscript λ. The group G is represented explicitly, whereas elements of X are accessed generically. We use $\mathrm{negl}(\lambda)$ to denote any (unspecified) negligible function.

Identity-based Encryption. An identity-based encryption (IBE) scheme consists of algorithms Setup, KeyGen, Enc, and Dec. Setup produces public parameters pp and a master secret key msk, KeyGen derives a secret key $\mathsf{sk}_{\mathsf{id}}$ for any identity id, while anyone can encrypt using Enc under id using only pp. Correctness requires that decrypting an honestly generated ciphertext using Dec with the corresponding identity key recovers the encrypted message except with negligible probability. Throughout this paper, we consider IBE schemes with bit-messages. See Definitions 1–2 (Section 3.1) for the formal exposition.

Fully Selective Security of IBE. For our impossibility results, we consider a weak, fully-selective notion of security for IBE (described formally in Definitions 3–4, Section 3.1). In the fully selective security experiment, the adversary commits before setup to a challenge identity and all identities whose keys it will request, with the challenge identity excluded from the extraction set; after receiving the public parameters, the permitted keys, and an encryption of a uniformly random bit under the challenge identity, it must determine which message was encrypted. Thus, our impossibility results also rule out IBE schemes satisfying stronger adaptive notions of security.

2.1 Two Variants of the Generic Group Action Model

We study IBE in two complementary formulations of the generic group action model: the random-set-representation GGAM (RSR-GGAM) and the type-safe GGAM (TS-GGAM). As is customary for generic-model lower bounds, we say that an adversary is *pseudoefficient* if it makes only polynomially many generic-oracle queries, although its local computation may be unbounded. We use such pseudoefficient adversaries to instantiate the fully selective security experiment for IBE described above in our GGAM-based attacks. We additionally describe all algorithms in a total finite-randomness normal form: for every security parameter and syntactic input profile, their random-tape spaces are finite and effectively enumerable, and they halt on every tape. This convention is automatic for ordinary probabilistic polynomial-time algorithms; looking ahead, this convention is needed only to make the uniform offline searches used by the pseudoefficient adversaries in our attack descriptions well-defined.

The Random-Set-Representation GGAM. The random-set-representation GGAM, which we abbreviate RSR-GGAM, is the random-label model introduced for group actions by Duman et al. [DHK+23]. Given a finite regular action $(G, X, \star, \widetilde{x})$, sample a uniformly random injection

$$\sigma_X : X \hookrightarrow \Sigma_\lambda, \qquad \Sigma_\lambda = \{0,1\}^{\ell(\lambda)}.$$

The element $x \in X$ is represented by the ordinary string $[x]_{\sigma_X} = \sigma_X(x)$. Let $\mathrm{im}(\sigma_X) \subseteq \Sigma_\lambda$ denote the set of random labels assigned by σ_X to elements of the action set X. The action oracle is

$$\mathcal{O}_\star^{\sigma_X}(g, L) = \begin{cases} \sigma_X\left(g \star \sigma_X^{-1}(L)\right), & L \in \mathrm{im}(\sigma_X), \\ \perp, & L \notin \mathrm{im}(\sigma_X). \end{cases}$$

The group G remains explicit: its elements can be sampled, multiplied, inverted, and tested for equality without an oracle. Random labeling conceals the semantic identity of x, but it does not make its label opaque. An RSR-GGAM algorithm may inspect the bits of L, hash them, branch on them, construct new strings from them, and submit arbitrary strings to the action oracle. This representation sensitivity is important for an impossibility theorem: even if two semantic executions have the same equality pattern, their algorithms may behave differently when corresponding elements receive different literal labels.

Our treatment of the RSR-GGAM makes several conventions explicit. We allow arbitrary finite abelian regular acting groups, whereas the basic formulation of [DHK+23] is presented for cyclic actions. Invalid strings are explicitly rejected, and we keep track of this behavior through a rejection set. Structured outputs identify which fields are intended to be action-set labels, and these designated fields are validated through the identity action. We also impose finite-randomness and totality conventions so that a pseudoefficient adversary can enumerate algorithmic supports offline. None of these conventions gives an algorithm an additional algebraic operation.

We additionally assume sufficient *label-slack*. Informally, even after polynomially many labels have been exposed, a polynomial-query algorithm should have negligible probability of guessing the representation of an unseen X-element. A sufficient condition is $(\mathrm{poly}(\lambda)|X|)/2^{\ell(\lambda)} = \mathrm{negl}(\lambda)$. We expand further on this assumption subsequently. See Section 3.3 for a more detailed treatment.

The Type-Safe GGAM. In the type-safe GGAM, abbreviated TS-GGAM, an action-set element is represented by an abstract handle of type $\mathsf{Handle}[X]$, rather than by an ordinary bit string. A type-safe algorithm may perform the following operations: (i) store, copy, and forward an existing handle; (ii) test two handles for equality; (iii) submit an explicit $g \in G$ and an existing handle to the action oracle; and (iv) return a handle in a designated handle-valued output position. Note that a type-safe algorithm cannot inspect a handle, cast it to an ordinary string, derive an explicit group element from it, or construct a handle from ordinary data. Equivalently, the computation is invariant under every consistent renaming of handles.

In this paper, we use an equality-explicit, equality-saturated normal form in which all relevant handle equalities are recorded in an instrumented transcript. We stress that this is primarily a proof artifact, rather than a modification to the model specification. Equality saturation changes neither the algorithm's action queries nor its outputs; it only makes every positive relation available to the proof. See Section 3.4 for a more detailed treatment.

We also remark that the TS-GGAM is not a new generic model introduced by this work; TS-GGAM interfaces already appear in [BGZ23, OZ24]. Our formulation of TS-GGAM specializes to finite regular actions, keeps the acting group explicit, permits polynomially many initial action-set handles supplied by a surrounding cryptographic experiment, and does not provide an opaque sampling interface for X. We treat equality as free; charging equality queries instead would change polynomial query bounds by at most a polynomial factor.

Relationship between the Models. A TS-GGAM computation can be embedded into the RSR-GGAM by representing each abstract handle by a random label and requiring the algorithm to treat that label opaquely. The converse fails because an RSR-GGAM algorithm may inspect and compute on literal label representations. We begin with the TS-GGAM because equality-pattern agreement there suffices for transcript transfer up to handle renaming, allowing us to better isolate the central algebraic invariant used by our impossibility result.

2.2 Impossibility of Restricted TS-GGAM-IBE

We now present an overview of our core technical result. We first consider a restricted class of IBE schemes in the TS-GGAM (referred to henceforth as restricted TS-GGAM-IBE), in which identity secret keys contain no $\mathcal{X}$-handles. For this class, we show that the number of efficiently indexable identities supported by a fully

selectively secure IBE scheme cannot exceed the number of non-origin $\mathcal{X}$-handles appearing in its public parameters, up to the explicit correctness and security losses stated below. Although the restriction simplifies the interface between key generation and decryption, the proof already contains the central technical ideas underlying our general separation. Section 2.3 subsequently removes the secret-key restriction while preserving the same impossibility.

At a high level, the restricted attack follows the following broad blueprint: choose the challenge identity at a random position among $(n_{\mathsf{pp}} + 1)$ identities, learn the dependencies revealed by the preceding identities, search offline for an alternative setup and a candidate challenge key consistent with those dependencies, and validate the candidate before applying it to the real challenge ciphertext. The novel aspect of our work is the machinery needed to carry out these steps in the GGAM, as described next.

The Restricted Syntax. A TS-GGAM-IBE public parameter has the following canonical form:

$$\mathsf{pp} = (\mathsf{pp}_{\mathsf{exp}}, \widehat{P}_1, \ldots, \widehat{P}_d), \qquad d \le n_{\mathsf{pp}},$$

where the displayed handles are the distinct non-origin handles occurring in the structured output; repetitions and occurrences of the public origin are recorded by an explicit shape. We set $P_0 = \widetilde{x}$ and call $\mathbf{P} = (P_0, P_1, \ldots, P_d)$ the semantic public-anchor tuple. We say that a TS-GGAM-IBE scheme is *restricted* if every identity secret key is entirely explicit and contains no action-set handle. Public parameters, ciphertexts, and the master secret key may still contain action-set handles. Explicit secret-key values may include arbitrary ordinary data and explicit elements of G. We refer to Section 4.1 for a more detailed exposition.

This restriction removes the most immediate obstacle to an alternative-world attack. A handle created in one simulated action world cannot be submitted to the oracle of another world. An explicit candidate key, however, is a well-typed input to decryption under both the candidate and real public parameters. Moreover, because the secret key supplies no new action-set anchor, every handle produced during the composed encryption-decryption computation descends from the public origin or one of the d public anchors. The relation rank governing the attack is therefore at most $d \le n_{\mathsf{pp}}$.

For the restricted impossibility, we use *pointwise correctness*: namely, the correctness probability is evaluated after fixing a particular setup output $(\mathsf{msk}, \mathsf{pp})$. This is stronger than standard IBE correctness, which averages also over setup. Section 2.3 later removes this strengthening.

Anchored Transcripts and Gain Graphs. We now present an overview of the formalism around anchored transcripts and gain graphs underlying all of our separation results. A more detailed treatment appears in Section 5.1. Assign the term (i, e) to the input handle representing an action-set element P_i. We write $\mathbf{P} = (P_0, \ldots, P_d)$ for the semantic tuple, where $P_0 = \widetilde{x}$ is the public origin and each $P_i \in \mathcal{X}$ is the action-set element represented by the public handle $\widehat{P}_i$, for $i \in [d]$. If a handle annotated by (i, g) is acted upon by $a \in G$, annotate the response by $(i, a \circ g)$. The semantics of an anchored term are: $[\![(i, g)]\!]_{\mathbf{P}} = g \star P_i$. Hence action queries modify only the explicit displacement g and never change the anchor index i. If occurrences annotated by (i, g) and (j, h) are equal, then

$$g \star P_i = h \star P_j \quad \Longleftrightarrow \quad P_j = (h^{-1} \circ g) \star P_i.$$

We record this as an edge from i to j with gain $h^{-1} \circ g$. All observed edges are simultaneously realized by $\mathbf{P}$, so the resulting gain graph is balanced: the product of the gains along every closed walk is the identity of G.

We write $\mathrm{Rel}_{\mathbf{P}}(\mathsf{Tr})$ for the canonical set of positive relative-action relations extracted from a transcript Tr under the anchor tuple $\mathbf{P}$. An anchor tuple $\mathbf{P}'$ *realizes* a relation set R if every labeled relation in R holds when its endpoints are interpreted using $\mathbf{P}'$. For a fixed anchor tuple, there is a unique canonical relation between every pair of vertices. Identifying each canonical relation with its underlying undirected edge gives the graphic relation matroid $\mathcal{M}_{\mathbf{P}}$. Its closure has a concrete meaning: a relation belongs to $\mathrm{cl}_{\mathbf{P}}(R)$ exactly when its endpoints are connected by R, in which case the gain is obtained by composing the relations along any connecting path. Since there are $(d + 1)$ vertices, $\mathrm{rank}_{\mathbf{P}}(R) \le d$. We now state the following (informal) lemma.

Informal Lemma 2.1 (Anchored Transcripts) *(Informal Version of Lemma 1) Every handle in a type-safe generic-action transcript is a known translate of an input anchor. Its positive equalities form a balanced*

gain graph of rank at most the number of non-origin anchors, whose closure captures all implied equalities; consequently, anchor tuples agreeing on all reached-term equalities induce executions identical up to type-preserving handle renaming.

Learning Relations by Repeated Sampling. The anchored-transcript lemma identifies a bounded-rank object that captures the algebraic information revealed by a generic-action computation. To exploit this structure, the attack must learn enough relations from computations associated with extracted identities to predict the relations that will arise for the challenge identity. It therefore accumulates the gain graphs exposed by a sequence of sampled computations and relies on the fact that a relation set escaping the current closure must increase rank (an event that can occur only a bounded number of times). We next turn this deterministic rank bound into probabilistic guarantees that a later computation reveals no relation outside the learned closure (see Section 5.2 for a more detailed treatment). We use the independent-sampling form later to eliminate action-set handles from secret keys, and the random-prefix form in the restricted attack to choose a challenge identity whose computation is already explained by the preceding identities. The first consequence concerns independent repetitions of the same computation, captured as follows.

Informal Lemma 2.2 (Repeated-Sampling Closure) *[Informal Version of Lemma 2 and Corollary 2] Let $R_1, \ldots, R_M$ be independent and identically distributed random subsets of a matroid of rank at most r. Then*

$$\Pr[R_M \not\subseteq \mathrm{cl}(R_1 \cup \cdots \cup R_{M-1})] \leq \frac{r}{M}.$$

In particular, let $\mathsf{Tr}_1, \ldots, \mathsf{Tr}_M$ be independent executions of the same generic-action computation on an anchor tuple $\mathbf{P} = (P_0, \ldots, P_d)$, and set $R_i = \mathrm{Rel}_{\mathbf{P}}(\mathsf{Tr}_i)$. Since the corresponding relation matroid has rank at most d, the probability that the final transcript exposes a positive relation not implied by the preceding transcripts is at most d/M.

Intuitively, whenever a newly sampled relation set escapes the current closure, the rank of the accumulated set must increase. Since the total rank is at most r, this can happen only r times; identical distribution and monotonicity then bound the probability that it happens at the final sample. Relations within an individual transcript may be arbitrarily correlated and adaptively generated; only the complete transcripts must be independent samples of the same computation.

The second sampling statement, which drives the restricted attack, does not require the relation sets to be independent.

Informal Lemma 2.3 (Random-Prefix Relation Learning) *[Informal Version of Lemma 3 and Corollary 3] Let $R_1, \ldots, R_k$ be arbitrarily jointly distributed subsets of a matroid of rank at most r, and let $Y \leftarrow [k]$ be chosen independently of the entire sequence. Then, with probability at least $(k-r)/(2k)$ over the random position Y and the observed prefix $R_1, \ldots, R_{Y-1}$,*

$$\Pr[R_Y \subseteq \mathrm{cl}(R_1 \cup \cdots \cup R_{Y-1}) \,|\, Y, R_1, \ldots, R_{Y-1}] \geq \frac{k-r}{2k}.$$

In other words, with noticeable probability the relations exposed at the randomly selected position are already implied by those exposed earlier. The conditional form of the statement is important because the relation set associated with a later identity may be constructed recursively using candidates obtained from the earlier relation sets. Taking $k = n_{\mathsf{pp}}+1$ and $r \leq n_{\mathsf{pp}}$ makes both the outer probability and the conditional-containment threshold at least $1/(2k)$.

Two-sided Transcript Transfer. Our attack will learn a relation graph from computations associated with extracted identities and then search offline for an alternative setup world containing a candidate key for the challenge identity. Correctness guarantees that this key decrypts successfully in its own alternative world, but the challenge ciphertext is generated using the real public parameters. We therefore need a transfer principle showing that the composed encryption-decryption computation follows the same path, and produces the same output, in the real and alternative worlds whenever both are governed by the learned relations. The random-prefix argument will ensure this condition on the real side, while the offline candidate search will enforce it on the alternative side. This is the purpose of two-sided transcript transfer: it converts containment

in the learned relation graph into agreement of the two executions, thereby allowing correctness in the alternative world to be used against the real challenge.

Suppose that $\mathbf{P}$ and $\mathbf{P}'$ are two public-anchor tuples that both realize a learned relation set R. Consider executions of the same deterministic generic-action algorithm, with the same explicit input and random tape, under $\mathbf{P}$ and $\mathbf{P}'$. It is not enough to assume that the relations observed in the first execution are implied by R. An equality that is negative under $\mathbf{P}$ might become positive under $\mathbf{P}'$, changing the control flow of the second execution. We therefore impose a two-sided closure condition: $\mathrm{Rel}_{\mathbf{P}}(\mathsf{Tr}_{\mathbf{P}}) \subseteq \mathrm{cl}_{\mathbf{P}}(R)$ and $\mathrm{Rel}_{\mathbf{P}'}(\mathsf{Tr}_{\mathbf{P}'}) \subseteq \mathrm{cl}_{\mathbf{P}'}(R)$. If a first divergence existed, it would be caused by an equality that holds in exactly one execution. Equality saturation would record the corresponding positive relation on that side; the closure condition and the fact that both anchor tuples realize R would force the same equality on the other side, a contradiction.

Informal Lemma 2.4 (Two-Sided Transfer) *[Informal Version of Lemma 4 and Corollary 5] If two anchor tuples realize the same relation set R and every positive relation in either execution lies in the appropriate closure of R, their transcripts agree up to handle renaming. Consequently, a composed encryption-decryption computation using the same explicit candidate key has the same output in both worlds.*

Offline Alternative Worlds. The two-sided transfer lemma explains how a computation can be moved from one anchor tuple to another, but it does not itself construct the alternative world or the challenge-identity key required by the attack. The adversary must find such a key without extracting the genuine challenge key and without submitting handles created in one action world to the oracle of another. We address this by exploiting the pseudoefficient setting: although the adversary makes only polynomially many real-oracle queries, its local computation may exhaustively search over offline setup and key-generation executions. The purpose of the offline model below is therefore to construct a candidate setup world consistent with the learned relation graph, find a challenge-identity key that decrypts correctly in that world, and certify the closure conditions needed to transfer this correctness back to the real world. Only the explicit candidate key leaves the offline simulation, which is precisely why the restricted secret-key syntax is important at this stage.

Observe that regularity gives every $x \in X$ a unique origin-relative coordinate $a \in G$ such that $x = a \star \widetilde{x}$. These coordinates are not available to an online generic algorithm. An offline simulator can, however, define a fresh abstract action world by maintaining an injective table $\mathsf{HT} : G \rightharpoonup \widetilde{\mathcal{H}}$ from coordinates to formal handles. Acting by g on the formal handle for a returns the formal handle for $g \circ a$, allocated deterministically when first required. Equality is coordinate equality. This exactly simulates a real type-safe execution up to handle renaming; see Lemma 5 (Section 5.4) for a rigorous formalization.

Because generic algorithms have finite, effectively enumerable random tapes, an unbounded local computation can enumerate all setup and key-generation executions under such offline oracles. It can also enumerate every encryption and decryption random tape and compute exact correctness and relation-containment probabilities. None of this requires a query to the real action oracle.

Admissible Candidates. We formalize the object sought by the offline search as an *admissible candidate*. Its conditions ensure simultaneously that the alternative world is compatible with the observed public parameters and learned relations, and that its challenge-identity key is both correct in that world and suitable for subsequent transfer to the real execution. A bit more concretely, given real public parameters and a learned relation set R, an *admissible candidate* consists of an offline setup world and an explicit challenge-identity key satisfying the following conditions: (i) the setup and key occur in the supports of the corresponding algorithms; (ii) the explicit public-parameter component and canonical shape agree with those of the real public parameters; (iii) the candidate public-anchor tuple realizes R; (iv) encryption followed by candidate-key decryption is correct with high probability in the candidate world; and (v) the relations exposed by that computation remain in $\mathrm{cl}(R)$ with high probability.

Exhaustive enumeration yields a deterministic procedure that returns an admissible candidate whenever one exists (see Lemma 6 in Section 5.4). The candidate-world handles remain purely symbolic and are never submitted to the real oracle. Only the explicit candidate secret key is exported from the offline world. This is exactly where the restricted secret-key syntax is used.

The Random-Prefix Attack. We now combine the preceding ingredients into the attack against restricted TS-GGAM-IBE. The adversary places the challenge identity at a uniformly random position in a sequence of

$(n_{\mathsf{pp}}+1)$ identities and uses the keys preceding that position to learn a cumulative relation graph. Because this graph has rank at most n_{pp}, with noticeable probability the computation for the challenge identity reveals no new independent relation. On this event, the offline search finds a suitable candidate challenge key, and two-sided transcript transfer allows the adversary to validate and use that key against the real challenge ciphertext.

Set $n = n_{\mathsf{pp}}$ and let $k = (n+1)$. Because the identity space is efficiently indexable and contains at least k identities, the adversary can, before setup and independently of the hidden action world, compute a publicly determined ordered tuple of pairwise distinct identities $\mathsf{id}_1, \ldots, \mathsf{id}_k$. The adversary samples $Y \leftarrow [k]$, declares id_Y as its challenge identity, and requests the keys for $\mathsf{id}_1, \ldots, \mathsf{id}_{Y-1}$. For every preceding identity, it repeatedly executes honest encryption-decryption computations and records their relations. The learning process is recursive: it also runs the current offline candidate, when one exists, and records the relations exposed in those candidate-key computations. Including both genuine-key and candidate-key relations is what later supplies the two sides of the transcript-transfer condition.

For each identity id_i, let E_i denote the complete block of positive relations collected at this stage: it is the union, over all sampling repetitions and both message bits, of the relations exposed by the composed encryption-decryption computations using the genuine key for id_i and, whenever one exists, the current offline candidate key. Thus, E_i is the new relation information contributed by the i-th identity, whereas $(R_0 = \varnothing, \{R_i = R_{i-1} \cup E_i\}_{i\in[Y-1]})$ represents the cumulative relation set learned through the first i identities.

A Good-Prefix Event. For the analysis, we define the complete hypothetical sequence $(E_1, R_1), \ldots, (E_k, R_k)$ independently of Y and reveal to the actual adversary only the prefix preceding Y. This makes Y independent of the sequence, as required by Lemma 2.3. Because all R_i belong to a relation matroid of rank at most n, applying Lemma 2.3 allows us to argue that, with probability at least $1/(2k)$, the challenge block is conditionally likely to expose no relation outside $\mathrm{cl}(R_{Y-1})$. We call this the *good-prefix event*.

On a good prefix, correctness and an averaging argument show that some genuine challenge-identity key simultaneously has high correctness and high relation-containment probability. Replaying its setup and key-generation tapes in the offline coordinate model proves that an admissible candidate exists. The exhaustive search therefore finds one. The real and candidate public-anchor tuples both realize R_{Y-1}, and their composed computations satisfy two-sided relation containment with noticeable probability. Applying Lemma 2.4 (more specifically, Corollary 5) allows transferring candidate-world correctness to the real public-parameter world.

Finally, the adversary tests the candidate on fresh, publicly generated encryptions. If validation fails, it outputs an independent bit; if it succeeds, it applies the candidate to the challenge ciphertext. Concentration bounds ensure that validation accepts a useful candidate with overwhelming probability but prevents a bad candidate from reducing the adversary's success substantially.

Informal Theorem 2.5 (Restricted TS-GGAM Impossibility) *[Informal Version of Theorem 1] Let Π be a restricted TS-GGAM-IBE scheme whose public parameters contain at most n distinct non-origin action-set handles and whose efficiently indexable identity-space family contains at least $(n+1)$ identities. If its pointwise correctness error is at most $1/(640(n+1))$, then a uniform pseudoefficient fully selective adversary achieves advantage $1/(40(n+1)) - e^{-\Omega(\lambda)}$. Consequently, a fully selective-secure restricted scheme with negligible pointwise correctness error can support at most n identities for all sufficiently large λ.*

Attack Success. Let $k = (n+1)$. By the random-prefix lemma (Lemma 2.3), the good-prefix event occurs with probability at least $1/(2k)$. Conditioned on this event, correctness guarantees the existence of an admissible candidate, the offline search finds one, and two-sided transcript transfer (Lemma 2.4) shows that its explicit key decrypts real challenge-identity ciphertexts with probability at least $11/20$. The validation step accepts such a useful key except with exponentially small probability. Outside the good-prefix event, validation ensures that the adversary still wins with probability at least $1/2 - e^{-\Omega(\lambda)}$: a poor candidate is rejected and the adversary guesses uniformly, while accepting a candidate whose success is at least $1/2$ cannot hurt. Consequently,

$$\Pr[\mathcal{A} \text{ wins}] \geq \frac{1}{2} + \frac{1}{2k}\left(\frac{11}{20} - \frac{1}{2}\right) - e^{-\Omega(\lambda)} = \frac{1}{2} + \frac{1}{40k} - e^{-\Omega(\lambda)}.$$

Attack Efficiency. The attack is pseudoefficient because it performs only n learning stages, polynomially many encryption-decryption samples per stage, and λ validation trials. The exhaustive candidate search may

require unbounded local computation, but it is carried out entirely using the offline simulator and makes no query to the real action oracle. Thus, the total number of generic-action queries remains polynomial, as required. See Section 5.5 for the detailed analysis.

2.3 The Tight TS-GGAM-IBE Impossibility

Section 2.2 established the impossibility for schemes with handle-free identity keys. We remove this restriction using a compiler that preserves the public parameters, ciphertexts, identity space, and security, while introducing only a controlled correctness loss. The resulting bound depends only on the n_{pp} non-origin public handles and is tight: at most n_{pp} identities are possible, and an n-identity construction using at most n such handles achieves the bound.

Eliminating Action-Set Handles from Secret Keys. We now present a compiler that transforms an arbitrary TS-GGAM-IBE scheme into the restricted form required by the attack of Section 2.2. The difficulty is that action-set handles appearing in a secret key act as additional private anchors and, unlike explicit data, cannot be transferred between the real and offline action worlds. As in the GGM approach from [PRV12, SGS21], the compiler removes these hidden objects by learning how they are used in honest encryption-decryption computations and subsequently representing them symbolically. The implementation, however, is specific to the GGAM: it uses the bounded-rank gain graph to learn the relevant relative-action relations and a hybrid real-symbolic execution to reproduce decryption without retaining the original secret handles. We present an overview of our compiler below. See Section 6.1 for additional details.

A general TS-GGAM identity key may contain action-set handles $\widehat{Z}_1, \ldots, \widehat{Z}_s$. We treat the distinct secret-only handles as additional persistent anchors, together with the origin and the public anchors:

$$\mathbf{A}_{\mathsf{id}} = (\widetilde{x}, P_1, \ldots, P_d, Z_1, \ldots, Z_s).$$

If $d \le n_{\mathsf{pp}}$ and $s \le n_{\mathsf{sk}}$, then the corresponding relation matroid has rank at most $(n_{\mathsf{pp}}+n_{\mathsf{sk}})$. The compiled key-generation algorithm first produces an ordinary secret key. For each message bit it then runs M independent, instrumented encryption-decryption computations and collects the positive relative-action relations exposed by them. It replaces the action-set handles in the key by formal indices and retains a canonical gain-labeled spanning forest of the learned relations. The compiled key contains only the following: (i) the explicit portion of the original key; (ii) an explicit shape describing repeated, public, and secret-only handle occurrences; and (iii) the gain-labeled relation forest. The compiled decryption interprets the original decryption code using a hybrid real-symbolic handle interface. A formal secret term (Z_j, g) records the result of translating secret anchor Z_j by $g \in G$. If the learned forest connects its component to a public anchor, the term can be materialized by acting on the corresponding real public handle. A component disconnected from all public anchors remains purely symbolic. Equality of two symbolic terms is decided from their normalized coordinates in the forest; equality involving only real handles is tested normally.

The symbolic execution never introduces a false positive: every symbolic equality is justified by a learned relation realized by the genuine anchor tuple. Conversely, if genuine and symbolic decryption differ, the genuine execution must expose a positive relation outside the closure of the learned forest. The repeated-sampling bound for generic-action transcripts (as summarized in Lemma 2.2) allows us to argue that this escape event occurs with probability at most $(n_{\mathsf{pp}} + n_{\mathsf{sk}})/(M + 1)$.

Informal Theorem 2.6 (Secret-key Handle Elimination) *[Informal Version of Theorem 6] Let Π be a bit-message TS-GGAM-IBE scheme with at most n_{pp} non-origin public action-set handles and at most n_{sk} handles in an identity secret key. For every polynomially bounded parameter $M \ge 1$, there is a compiled scheme $\widehat{\Pi}$ satisfying all of the following: (i) setup, public parameters, encryption, ciphertexts, and the identity space are unchanged; (ii) identity secret keys of $\widehat{\Pi}$ contain no action-set handles; (iii) the correctness error increases by at most $(n_{\mathsf{pp}} + n_{\mathsf{sk}})/(M + 1)$; and (iv) every selective attack against $\widehat{\Pi}$ yields an attack against Π with the same advantage.*

We stress that the above security reduction is exact, rather than merely computational or statistical. Given an adversary against the compiled scheme, the reduction participates in the selective-security experiment for the original scheme and forwards the public parameters and challenge ciphertext unchanged. Whenever the

adversary requests an identity key, the reduction obtains the corresponding original key, runs the compiler's prescribed training procedure using that key, and returns the resulting handle-free compiled key. This produces exactly the same key distribution and oracle view as a genuine execution of the compiled scheme; consequently, the adversary's view and winning event are identically distributed in the two experiments. There is therefore no security loss: the reduction against the original scheme has precisely the same advantage as the adversary against the compiled scheme. Since the training uses only polynomially many additional generic-action queries, pseudoefficiency is preserved as well.

From Averaged Correctness to the General Impossibility. The restricted TS-GGAM-IBE attack theorem (Theorem 2.5) is stated setup-wise, whereas standard IBE correctness averages over setup. We bridge the two formulations through good setup worlds. Fix the $(n_{\mathsf{pp}} + 1)$ identities used by the attack. For each identity and message bit, the expectation over setup of the conditional decryption error is bounded by the standard correctness error. Markov's inequality, followed by a union bound over the selected identities and both messages, shows that all required conditional correctness bounds hold simultaneously except with small probability (see Lemma 8 for a formal exposition).

We apply the handle-elimination compiler described above with a sufficiently large polynomial M. On every good setup world, the compiled scheme satisfies the correctness threshold of the restricted attack. On bad setup worlds, candidate validation guarantees success at least $1/2$ up to a negligible term, so these worlds cannot cancel the advantage obtained on good ones. Exact security preservation then transfers the attack from the compiled scheme to the original scheme.

Informal Theorem 2.7 (General TS-GGAM Impossibility) *[Informal Version of Theorem 7 and Corollary 6] Let Π be a TS-GGAM-IBE scheme for an efficiently indexable identity family ID_λ whose public parameters contain at most $n = n_{\mathsf{pp}}(\lambda)$ distinct non-origin action-set handles. If Π satisfies fully selective security and negligible standard correctness error, then $|\mathsf{ID}_\lambda| \le n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ.*

An important feature of the resulting bound is that it depends only on the number n_{pp} of public action-set handles, and not on the number n_{sk} of handles contained in an original identity secret key. The latter handles appear as additional anchors only during the compiler's training phase, where the relevant relation matroid has rank at most $(n_{\mathsf{pp}} + n_{\mathsf{sk}})$. Accordingly, n_{sk} influences the number of training samples required to make the compiler's additional correctness error smaller than the inverse-polynomial correctness threshold required by the attack. Once these secret handles have been replaced by their symbolic description, however, the compiled scheme is restricted: every subsequent attack computation is anchored only at the public origin and the public-parameter handles. The random-prefix attack therefore operates in a relation matroid of rank at most n_{pp}, yielding the identity bound $|\mathsf{ID}_\lambda| \le n_{\mathsf{pp}}(\lambda)$. In other words, adding arbitrarily many polynomially bounded handles to an identity key may increase the cost of compilation, but it cannot increase the number of identities supported by a secure TS-GGAM-IBE scheme.

A Matching Construction. To show that the preceding identity bound is inherent rather than an artifact of the proof, we give a simple TS-GGAM-IBE construction that meets it exactly. For every n, the construction supports n identities using at most n non-origin public action-set handles, with negligible correctness error and full selective security in the TS-GGAM. Thus, one public handle per supported identity is both necessary and sufficient. Let $\mathsf{ID} = [n]$. The construction is simply n independent copies of an elementary GGAM public-key encryption, with one public-key/secret-key pair assigned to each identity. Correctness thus follows immediately: abelianness gives perfect correctness for $b = 0$, while a random encryption of 1 is mis-classified with probability $1/|G|$.

The main work lies in analyzing the generic security of the construction. Note that an adversary may interleave action queries across all public keys and the challenge components and detect accidental equalities among the resulting handles. The argument below couples the two challenge experiments until the first such non-forced equality and bounds the probability of this event by the corresponding collision probability. Concretely, we observe that the two challenge distributions contain either the planted parallelogram $\big(a\star\widetilde{x}, r\star\widetilde{x}, (a\circ r)\star\widetilde{x}\big)$ or the tuple $\big(a\star\widetilde{x}, r\star\widetilde{x}, s\star\widetilde{x}\big)$. A generic-action computation distinguishes them only if it encounters a non-forced equality involving the challenge-dependent anchors. If its equality-saturated execution encounters at most T handle occurrences, a union bound gives distance at most $2\binom{T}{2}/|G|$ (see Lemma 11 in Section 6.3

for the formal argument). Consequently, the scheme supports n identities with at most n non-origin public handles and is secure whenever $1/|G_\lambda|$ is negligible.

Informal Theorem 2.8 (Tight Public-Handle Bound) *[Informal Version of Theorem 8 and Corollary 7] For every polynomially bounded $n = n(\lambda)$, there exists a fully selective-secure restricted TS-GGAM-IBE scheme supporting n identities with at most n non-origin public handles, provided $1/|G_\lambda|$ is negligible. Conversely, any fully selective-secure TS-GGAM-IBE scheme with negligible correctness error and at most n distinct non-origin action-set handles supports at most n efficiently indexable identities.*

2.4 Extending the Impossibility from TS-GGAM to RSR-GGAM

The TS-GGAM result in Section 2.3 applies to algorithms that treat action-set handles as opaque objects. We now move toward the more permissive RSR-GGAM, where action-set elements are represented by ordinary random strings whose bits may be inspected, hashed, and used to influence subsequent computation. This additional freedom prevents the TS-GGAM proof from carrying over directly: semantic agreement up to handle renaming no longer implies identical algorithmic behavior, and labels allocated during one computation may remain hidden in the shared representation and affect later computations.

To address this, we first isolate a natural class of *restricted key-label-oblivious* and *composably label-sanitizable* RSR-GGAM-IBE schemes for which these representation effects can be safely controlled. We then lift the gain-graph-based TS-GGAM attack to this class through a novel, representation-aware simulation framework that controls the random labels shared across real and alternative computations. The resulting impossibility retains the tight TS-GGAM bound (at most n_{pp} identities for n_{pp} designated public labels).

Finally, to make the sanitizability requirement concrete, we give a natural syntactic sufficient condition: labels that are erased at a computation boundary must not influence the retained state through their literal representations, in which case a boundary-renaming argument establishes composable label sanitizability.

The Restricted and Sanitizable RSR Syntax. To recover the alternative-world attack in the RSR-GGAM, we must formalize when a secret key and the surrounding retained state are free of inaccessible information about previously generated random labels. Merely requiring that a secret key contain no designated action-set label is insufficient; since labels are ordinary strings, an algorithm could copy or hash a private label into an ordinary field and later use the resulting value to influence its behavior.

To address this, we impose two targeted conditions. Restricted *key-label obliviousness* controls the labels hidden by the setup and key-generation prefix, while *composable label sanitizability* ensures that private labels discarded between later encryption-decryption computations can be safely forgotten. Together, these conditions preserve exactly the label symmetry needed by the alternative-world simulation while still permitting representation-sensitive computation on labels that remain visible. Concretely, we call an RSR-GGAM-IBE *restricted key-label-oblivious* when both of the following hold: (i) identity secret keys contain no designated action-set labels; and (ii) after setup and any polynomially bounded sequence of key-generation invocations, the labels of semantic elements not retained in the structured public parameters or explicit keys remain conditionally uniform after fixing the semantic transcript and all retained state.

Recall additionally that the TS-GGAM attack needs to compose many encryption-decryption computations. Hence, to adapt the attack to the restricted key-label-oblivious RSR-GGAM, we require composable label sanitizability. Consider an interface point between two composed computations at which the state needed by subsequent computations is retained, while all transient labels from the completed computation are intended to be forgotten; we call such a point a *sanitization boundary*. At such a boundary, write $\mathrm{dom}(\mathsf{LT}) = D_{\mathsf{live}} \dot\cup D_{\mathsf{priv}}$, where D_{live} consists of elements whose designated labels survive the boundary and D_{priv} consists of all other elements whose labels have been allocated.

To state the required symmetry, we fix all information that is allowed to remain visible across the boundary: the ordinary retained state K; the label-free semantic history τ_{sem}; the table $\mathsf{LT}_{\mathsf{live}}$ assigning labels to semantic elements that remain live; the set Rej of strings already rejected as invalid; and the set D_{priv} of semantic elements whose labels are to be forgotten. Crucially, we do *not* fix the literal strings assigned to the elements of D_{priv}. Even after all the retained information above has been fixed, composable sanitizability requires the private table $\mathsf{LT}_{\mathsf{priv}}$ to remain uniformly distributed over $\mathsf{Inj}\big(D_{\mathsf{priv}}, \Sigma_\lambda \setminus \big(\mathrm{im}(\mathsf{LT}_{\mathsf{live}}) \cup \mathsf{Rej}\big)\big)$. The same condition must continue to hold after any subset of earlier boundaries has already been sanitized. See Section 7.1 for a more formal treatment.

Permitted Representation Sensitivity. We note that the above requirements are weaker than global type safety or complete label opacity (as was assumed in the TS-GGAM). Labels that remain live across a boundary (including public and ciphertext labels) may be inspected, hashed, or used to influence control flow, since their representations are retained in $\mathsf{LT}_{\mathsf{live}}$. Transient labels may likewise be used during their lifetime; the requirement is only that, once such a label is forgotten, its literal value leave no residual information in the retained semantic or ordinary state. Equivalently, forgotten labels must remain freely permutable without affecting any later computation. This rules out, for example, retaining the hash of an erased private label in an explicit key field, while still allowing substantial representation-sensitive behavior on labels that remain visible.

For example, an encryption algorithm may hash a retained public label and use the result to choose explicit randomness or control flow. Such behavior is not type safe, since it depends on the literal label representation, but it is compatible with our RSR conditions because the public label remains live across the boundary. What is excluded is only representation-dependent information about a label that is subsequently erased.

A Syntactic Criterion. Before proceeding with the attack description, we point out that composable label sanitizability is a semantic condition and may therefore appear difficult to verify directly. To address this, in Section 7.8, we present a natural syntactic sufficient condition that captures a standard discipline for handling temporary random labels: a label intended to disappear at a boundary is treated as an opaque token during its lifetime and leaves no representation-dependent information in the surviving state.

More precisely, in the *erasure-safe boundary normal form*, a boundary-private label may be copied, compared with another valid label, submitted from label-typed storage to the action oracle, or recorded in a renaming-invariant semantic transcript, but its literal bits may not affect retained ordinary state, explicit group operands, the rejection set, or the choice of labels that remain live. A boundary-renaming argument then shows that the erased label table remains conditionally uniform. This criterion discharges the composable-sanitizability hypothesis only. The requirement that setup and key generation be restricted key-label-oblivious is imposed separately in the RSR impossibility theorem.

Informal Lemma 2.9 (Syntactic Sanitizability) *[Informal Summary of Lemma 23, Corollary 15, and Theorem 12] Under erasure-safe boundary normal form, consistently renaming private labels preserves the execution probability and retained state. Since forgotten labels remain conditionally uniform and safely erasable, syntactic label sanitizability implies composable label sanitizability.*

In particular, canonical random-label implementations of type-safe schemes satisfy this criterion because they treat labels only as opaque tokens: labels may be copied, compared, or submitted to the action oracle, but their literal bits never affect retained ordinary state. Our matching construction has the same property, since its algorithms manipulate action-set labels only through these permitted opaque operations. These examples demonstrate that the criterion captures a natural class of implementations rather than an artificial restriction tailored solely to the proof. See Proposition 1 and Corollary 16 (Section 7.8) for a more detailed exposition.

Lazy Random Representations. Having identified the label symmetry required by the attack, our first technical step is to expose the randomness of the representation only as it becomes relevant. In the standard RSR-GGAM, a random injection is chosen for the entire action set at the beginning of the experiment, so even semantic elements that are never encountered already have hidden labels; this makes it difficult to synchronize alternative worlds or forget labels left behind by earlier computations. We instead allocate a fresh random label only when a semantic element is first exposed, while remembering strings that have previously been rejected as invalid. Except when an algorithm happens to guess the fixed label of an unseen element, this lazy process is indistinguishable from sampling the entire representation in advance. Under the label-slack assumption, such a guess occurs only with negligible probability, allowing the subsequent proof to couple labels synchronously and sanitize private representation state.

Concretely, the lazy-label oracle maintains a partial injection $\mathsf{LT} : X \rightharpoonup \Sigma_\lambda$ and a rejection set $\mathsf{Rej} \subseteq \Sigma_\lambda \setminus \mathrm{im}(\mathsf{LT})$. When a semantic element is first exposed, the oracle assigns it a uniform unused string outside the table image and rejection set. If an algorithm submits an unallocated string, that string is rejected and permanently added to Rej. Remembering rejected strings is necessary: a string cannot be invalid at one point and later become the label of a newly exposed element. Let $N = |X|$, $M_\lambda = |\Sigma_\lambda|$, and suppose an execution

begins with at most p exposed semantic elements and makes at most q action queries. Lemma 12 (Section 7.1) couples the fixed-representation and lazy-label experiments with error

$$\epsilon_{\mathsf{lbl}}(p,q) = \min\left\{1, \frac{qN}{M_\lambda - p - q}\right\}.$$

Note that failure occurs only when the algorithm guesses the fixed label of a semantic element not yet allocated in the lazy experiment. Under label slack, this error is negligible.

Label-Synchronous Transfer. The two-sided transfer lemma from the TS-GGAM shows that the real and candidate computations have the same semantic equality and action-input/output structure. In the RSR-GGAM, however, semantic agreement up to renaming is not enough: an algorithm may inspect the literal bits of a label, so assigning different strings to corresponding elements could change its local computation or control flow. The lazy representation allows us to overcome this problem by assigning the same fresh string to corresponding semantic elements whenever they are first encountered in the two worlds. By also synchronizing invalid-label queries and rejected strings, we obtain identical literal transcripts rather than merely isomorphic semantic transcripts. This label-synchronous transfer is the RSR analogue of the TS-GGAM transfer lemma and is what ultimately allows candidate-world correctness to be carried back to the real execution.

We call two lazy tables *synchronous* if corresponding semantic elements are assigned literally identical strings and the two rejection sets are identical. Consider two executions whose public-anchor tuples realize the same learned relation graph and for which every positive relation arising on either side is implied by that graph, which is precisely the two-sided transfer condition described in the TS-GGAM-IBE attack above. If the executions begin with synchronously labeled public anchors, then whenever corresponding action queries expose new semantic elements, the coupling assigns those elements the same fresh label. If a response coincides with a previously encountered element in one execution, equality saturation and two-sided closure force the corresponding coincidence in the other. Invalid-label queries are likewise rejected simultaneously because the rejection sets remain synchronized. The following (informal) lemma captures this; see Section 7.2 for additional details.

Informal Lemma 2.10 (Label-Synchronous Transfer) *[Informal Summary of Lemmas 13-14 and Corollary 9] If two lazy-label generic-action executions satisfy the two-sided semantic transfer condition and begin with synchronous public tables and rejection sets, then they can be coupled to have identical complete bit-level transcripts. This includes all label-dependent computation, invalid-label queries, control-flow decisions, and ordinary outputs.*

Forgetting Private Labels and Symmetry-Preserving Candidate Search. Even when a private label is absent from every retained output, it remains recorded in a shared lazy-label table and may affect subsequent allocations or oracle queries. We therefore compare the real continuation, which retains $(\mathsf{LT}_{\mathsf{live}}, \mathsf{LT}_{\mathsf{priv}}, \mathsf{Rej})$, with a sanitized continuation initialized from $(\mathsf{LT}_{\mathsf{live}}, \emptyset, \emptyset)$. Composable label sanitizability makes the erased private table a uniform hidden injection after the retained state has been fixed. The two continuations can consequently differ only if a future query guesses a forgotten private label or a fresh sanitized allocation selects a string occupied by the forgotten private table or rejection set. If there are at most p live elements, a private elements, r rejected strings, and q subsequent queries, Lemma 15 bounds this probability by

$$\epsilon_{\mathsf{forget}}(p,a,r,q) = \min\left\{1, \frac{q(a+r)}{M_\lambda - p - a - r - q}\right\}.$$

Applying the replacement successively at every designated boundary gives the composable-resetting loss $\sum_{j\in J} \epsilon_{\mathsf{forget}}(p_j, a_j, r_j, Q_j)$, as formalized by Lemma 16. Under label-slack and polynomial bounds, this loss is negligible. We may therefore analyze successive encryption–decryption samples as beginning from the same retained public-label table with fresh private-label randomness.

The offline candidate search must preserve the same conditional symmetry. Selecting a candidate according to its complete private label table could bias the forgotten labels and invalidate the resetting argument. We instead project each setup–key-generation prefix to the retained summary

$$\mathsf{Sum}(\mathsf{Pref}) = \big(K, \tau_{\mathsf{sem}}, \mathbf{P}, D_{\mathsf{priv}}, \mathsf{LT}_{\mathsf{live}}, \mathsf{Rej}\big),$$

which records all legitimately retained ordinary and semantic information but omits $\mathsf{LT}_{\mathsf{priv}}$. Candidates are enumerated and selected using only this summary, and their continuations begin from the sanitized state $\mathsf{San}(S) = (\mathsf{LT}_{\mathsf{live}}, \emptyset, \emptyset)$. Thus the search chooses an equivalence class of private-label assignments rather than privileging one literal assignment.

Because live labels remain representation-sensitive, an admissible candidate must reproduce the real public parameters as literal structured strings, not merely up to renaming. Its semantic anchor tuple must realize the learned relation set, and both the real and candidate sanitized continuations must expose only relations implied by that set. Label-synchronous transfer then couples their complete bit-level transcripts and ordinary outputs. The following statement combines the resulting existence, symmetry, and transfer guarantees.

Informal Lemma 2.11 (Sanitized Candidate Search) *[Informal Version of Lemmas 17-18] There is a deterministic offline procedure, making no query to the real action oracle, that finds an admissible sanitized candidate whenever one exists. Because its choice depends only on the retained summary, it preserves the conditional uniformity of the omitted private-label table. Moreover, if the real and candidate summaries have identical literal public parameters, their anchor tuples realize the same learned relation set, and both sanitized continuations expose only relations implied by that set, then their complete bit-level transcripts, and hence their ordinary outputs, can be coupled identically.*

The RSR-GGAM-IBE Attack. At a conceptual level, the attack follows the same random-prefix strategy as in the restricted TS-GGAM setting: choose the challenge identity among $(n_{\mathsf{pp}}+1)$ identities, learn relations from the preceding identities, find and validate an offline candidate key, and apply it to the challenge ciphertext. The main additional difficulty is that all RSR computations share a fixed random representation. Labels allocated during setup or one sampled computation may therefore remain hidden in the representation table and correlate later computations. The sanitization machinery developed above lets us pass, at a small statistical cost, to a resetting experiment in which every sample starts from the same public-label state with fresh private-label randomness. We make this transition in three steps. First, relation learning and candidate validation do not depend on the challenge ciphertext. Challenge encryption can therefore be postponed until these phases are complete, and final candidate-key decryption can be treated as part of the same composed computation as challenge encryption; this is formalized by Lemma 19 (Section 7.5).

Second, setup and extraction-key generation are executed against one fixed representation and then sanitized. Every subsequent encryption-decryption sample begins from the retained public-label table with an empty rejection set and an otherwise fresh lazy-label continuation, and its transient labels are discarded when the sample ends. As discussed under *Lazy Random Representations*, Lemma 12 (Section 7.1) allows the fixed representation to be replaced by a lazy one, while the private-label-forgetting and composable-resetting bounds of Lemmas 15 and 16 (Section 7.3) allow successive computations to begin from sanitized states. Combining these replacements bounds the distance between the resetting experiment and the original RSR experiment by δ_{comp}, which consists of the relevant fixed-to-lazy coupling losses and the cumulative forgetting loss.

Third, we must ensure that IBE correctness survives the same change of representation state. Lemma 20 (Section 7.5) applies the preceding fixed-to-lazy and forgetting arguments specifically to an honest encryption-decryption computation and shows that its error in the sanitized experiment is at most $(\epsilon_{\mathsf{corr}} + \delta_{\mathsf{cor}})$, where ϵ_{corr} is the averaged correctness error of the original IBE scheme, and δ_{cor} accounts for the representation-coupling and private-label-forgetting losses. Markov's inequality and a union bound then imply that, except with small probability, the fixed setup world has good sanitized correctness for both messages and all $(n_{\mathsf{pp}}+1)$ selected identities. See Lemma 21 for a formal exposition.

Conditioned on such a setup world, every relation-learning sample starts from the same sanitized public state and uses fresh continuation randomness. For the analysis, we associate a hidden sanitized genuine-key summary with every identity and define the complete recursive relation sequence independently of the random challenge position. These summaries do not change the adversary's view (it receives only the explicit keys for the identities it is permitted to extract). However, they ensure that every hypothetical sampling block is defined from the same reset state. The resulting relation sequence still lies in a gain-graph matroid of rank at most n_{pp}. The random-prefix argument, existence and selection of an admissible candidate, public validation, and final transcript transfer therefore proceed as in the TS-GGAM attack, with label-synchronous coupling replacing handle renaming. Under label-slack and polynomial query bounds, δ_{cor} and δ_{comp} are negligible, yielding the (informal) impossibility theorem below.

Informal Theorem 2.12 (RSR-GGAM-IBE Impossibility) *[Informal Version of Theorem 9 and Corollary 12] Assume that the random-string representation satisfies the standard label-slack condition. Let Π be a restricted key-label-oblivious and composably label-sanitizable RSR-GGAM-IBE scheme over an efficiently indexable identity-space family $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda\in\mathbb{N}}$. Suppose that its public parameters contain at most $n_{\mathsf{pp}}(\lambda)$ distinct designated non-origin labels, its action-query complexity is polynomially bounded, and its standard correctness error is negligible. If Π is fully selectively secure, then $|\mathsf{ID}_\lambda| \le n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ.*

Tightness in the RSR-GGAM. Finally, we show that the public-label upper bound established in Theorem 2.12 is tight even against the stronger, representation-sensitive adversaries permitted by the RSR-GGAM. In particular, replacing the handles in the TS-GGAM matching construction by random labels gives an n-identity scheme with at most n designated non-origin public labels. Moreover, the scheme is key-label-oblivious and satisfies the syntactic sanitizability criterion above. See Section 7.6 for the detailed analysis.

Informal Theorem 2.13 (Tight Public-Label Bound) *[Informal Version of Theorem 10] Suppose that $1/|G_\lambda|$ is negligible and that the random-string representation satisfies the standard label-slack condition. For every polynomially bounded $n = n(\lambda)$, there exists a fully selectively secure RSR-GGAM-IBE scheme supporting n identities with at most n distinct designated non-origin public labels. The scheme has negligible correctness error, is restricted key-label-oblivious and composably label-sanitizable, and remains secure against pseudoefficient representation-sensitive RSR-GGAM adversaries.*

Together with the preceding impossibility theorem, this shows that one designated non-origin public label per identity is both necessary and sufficient within the sanitizable RSR-GGAM class.

2.5 Extending the Impossibility to Twist-Augmented Group Actions

The preceding results concern the ordinary GGAM interface, which can translate a hidden action-set element by an explicitly known group element but cannot reverse its unknown displacement from the public origin. Several cryptographically relevant group actions additionally admit an efficient involution, called a *twist*, satisfying $\mathsf{tw}(a \star \widetilde{x}) = a^{-1} \star \widetilde{x}$. Because this operation cannot generally be implemented using the ordinary action oracle, our preceding impossibility theorem does not formally cover the twist-augmented model, and twist queries can reveal relations that are not captured by ordinary anchored gain graphs. We show that the provenance framework nevertheless extends after equipping each anchor with an orientation: this gives an unconditional upper bound of $2n$ identities from n non-origin public labels, and recovers the tight n-identity bound, together with the matching construction, whenever squaring in the acting group is bijective.

We show that the anchored provenance framework extends to this stronger interface. Let $P = (P_0, \dots, P_d)$ denote the tuple of public action-set anchors, with $P_0 = \widetilde{x}$ the public origin. By regularity of the action, each anchor $P_i \in \mathcal{X}$ has a unique coordinate $p_i \in G$ such that $P_i = p_i \star \widetilde{x}$ where p_0 is the identity of G. These coordinates are used in the analysis and need not be known to the generic algorithm. We represent every derived action-set element by an oriented term (i, s, g), where i identifies its originating anchor, $s \in \{+1, -1\}$ specifies whether that anchor's coordinate is used directly or inverted, and $g \in G$ is an explicitly tracked translation. Its semantics are $[\![(i, s, g)]\!]_P = (g \circ p_i^s) \star \widetilde{x}$. For the abelian acting group G, an action by $a \in G$ and the twist $\mathsf{tw}(a \star \widetilde{x}) = a^{-1} \star \widetilde{x}$ update terms as stated below:

$$(i, s, g) \xmapsto{a\star(\cdot)} (i, s, a \circ g), \qquad (i, s, g) \xmapsto{\mathsf{tw}} (i, -s, g^{-1}).$$

Thus, a twist reverses the orientation of a provenance chain while preserving its originating anchor; it still cannot combine two independently supplied anchors.

A direct analysis treats the two orientations of each non-origin anchor as distinct virtual anchors. The resulting relation graph has at most $(2n + 1)$ vertices and rank at most $2n$, yielding an unconditional upper bound of $2n$ supported identities. When the squaring map $\mathsf{sq}\colon G \longrightarrow G$ given by $\mathsf{sq}(g) = g^2$ is bijective (equivalently, when G has no nontrivial 2-torsion) the bound improves to n. Indeed, equality between (i, s, g) and (j, t, h) reveals the signed relation $p_i^s \circ p_j^{-t} = g^{-1} \circ h$. These relations form a rooted signed-frame matroid. Each unrooted balanced component retains one free anchor coordinate, whereas an unbalanced cycle gives an equation $p_i^2 = c$, which uniquely fixes p_i under bijective squaring; a component containing the origin

is already fixed. Summing over the components gives rank at most n. This is formalized by Lemma 22, which extends the provenance and transfer guarantees of Lemma 2.1 to an RSR-GGAM interface that additionally supports twists (see Section 7.7 for more details).

Informal Theorem 2.14 (Impossibility with Twists) *[Informal Version of Theorem 11 and Corollary 13] Assume the standard label-slack condition. Let Π be a fully selectively secure, restricted key-label-oblivious, and composably label-sanitizable twist-augmented RSR-GGAM-IBE scheme with negligible standard correctness error and polynomial action-and-twist-query complexity. Suppose its identity space is efficiently indexable and its public parameters contain at most $n = n_{\mathsf{pp}}(\lambda)$ distinct designated non-origin labels. Then the following hold:*

1. *In general, $|\mathsf{ID}_\lambda| \le 2n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ; and*
2. *If the squaring map $\mathsf{sq}\colon G_\lambda \to G_\lambda$, $\mathsf{sq}(g) = g^2$, is bijective for all sufficiently large λ, then $|\mathsf{ID}_\lambda| \le n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ.*

Finally, we also show that the matching RSR-GGAM-IBE construction described above remains secure under twists when squaring is bijective. After conditioning on the challenge-independent background, every non-forced challenge-dependent equality has the form $z^\alpha = c$ for $\alpha \in \{\pm 1, \pm 2\}$. The corresponding power map is bijective, so each equality occurs with probability at most $1/|G|$. Corollary 14 (Section 7.7) therefore gives the same asymptotic security bound as in the untwisted model. In particular, for odd-order actions, n public labels are both necessary and sufficient to support n identities.

3 Preliminaries

Notation. We write $\lambda \in \mathbb{N}$ for the security parameter, $[n] = \{1, \dots, n\}$, and $[a, b] = \{a, \dots, b\}$. For a finite set S, $x \leftarrow S$ denotes uniform sampling; more generally, $x \leftarrow D$ denotes sampling from a distribution D. We write $\mathsf{SUPPORT}(D)$ for the support of D, $\Delta(D_0, D_1)$ for statistical distance, and $\mathrm{negl}(\lambda)$ for an unspecified negligible function. For a function f, we write $\mathrm{im}(f)$, $\mathrm{dom}(f)$, and $f|_S$ for its image, domain, and restriction, respectively; $\dot\cup$ denotes disjoint union.

Throughout, $\mathsf{GA} = (G, X, \star, \widetilde{x})$ denotes a regular action of a finite abelian group G on X, with distinguished origin $\widetilde{x}$. We write the group operation multiplicatively as $\circ$, with identity e and inverse g^{-1}. Thus every $x \in X$ has a unique representation $x = g \star \widetilde{x}$. Security experiments, advantages, error probabilities, and query bounds are implicitly parameterized by λ when clear from context, and all algorithms are probabilistic unless stated otherwise.

3.1 Identity-Based Encryption

We first recall the syntax and security of identity-based encryption (IBE) in the standard model. Let $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda \in \mathbb{N}}$ be a non-empty family of identity spaces.

Definition 1 (Identity-Based Encryption). *An identity-based encryption scheme over ID is a tuple $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of probabilistic polynomial-time algorithms with the following syntax:*

$$\begin{aligned}
(\mathsf{msk}, \mathsf{pp}) &\leftarrow \mathsf{Setup}(1^\lambda), \\
\mathsf{sk}_{\mathsf{id}} &\leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}), && \mathsf{id} \in \mathsf{ID}_\lambda, \\
C &\leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m), && \mathsf{id} \in \mathsf{ID}_\lambda, \quad m \in \{0,1\}, \\
m' &\leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C), && m' \in \{0, 1, \perp\}.
\end{aligned}$$

Here msk and pp denote the master secret key and public parameters, respectively, and $\mathsf{sk}_{\mathsf{id}}$ denotes the secret key associated with identity id.

Definition 2 (Correctness of IBE). *An IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ has decryption error $\epsilon = \epsilon(\lambda)$ if, for every $\lambda \in \mathbb{N}$, every $\mathsf{id} \in \mathsf{ID}_\lambda$, and every $m \in \{0,1\}$,*

$$\Pr[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C) = m] \ge 1 - \epsilon(\lambda),$$

where $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$, *and* $C \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m)$. *The probability is over the internal randomness of all four algorithms. The scheme is perfectly correct if* $\epsilon(\lambda) = 0$, *and correct if* ϵ *is negligible.*

Definition 3 (Fully Selective Security Experiment for IBE). *Let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an adversary. The fully selective security experiment* $\mathsf{Expt}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda)$ *is defined as follows.*

1. *Run the first stage of the adversary:* $(\mathsf{id}^\star, \mathsf{id}_1, \ldots, \mathsf{id}_q, \mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda)$, *subject to the requirements that* $\mathsf{id}^\star, \mathsf{id}_1, \ldots, \mathsf{id}_q \in \mathsf{ID}_\lambda$ *and* $\mathsf{id}^\star \notin \{\mathsf{id}_1, \ldots, \mathsf{id}_q\}$. *If these conditions fail, output an independent uniformly random bit and halt.*
2. *Generate* $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}(1^\lambda)$.
3. *For every* $i \in [q]$, *generate* $\mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_i)$.
4. *Sample* $b \leftarrow \{0,1\}$ *and generate the challenge ciphertext* $C^\star \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}^\star, b)$.
5. *Run the second stage:* $b' \leftarrow \mathcal{A}_2(\mathsf{state}, \mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_q, C^\star)$.
6. *Output* 1 *if* $b' = b$, *and output* 0 *otherwise.*

The number $q = q(\lambda)$ *of extraction identities must be polynomially bounded.*

Definition 4 (Fully Selective Security of IBE). *The fully selective advantage of* $\mathcal{A}$ *against* Π *is*

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right|.$$

An IBE scheme Π *is fully selective-secure if* $\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}$ *is negligible for every probabilistic polynomial-time adversary* $\mathcal{A}$.

3.2 Regular Group Actions

For each security parameter λ, we consider a regular group-action instance $\mathsf{GA}_\lambda = (G_\lambda, X_\lambda, \star_\lambda, \widetilde{x}_\lambda)$, together with public parameters describing G_λ and its operations. Elements of G_λ have explicit and unique representations, and group operations, inversion, equality testing, membership testing, and sampling are available to all algorithms. We omit the subscript λ when clear from context.

3.3 The Random Set Representation Generic Group Action Model

We recall the random-representation generic group-action model of [DHK+23], which we call the *random set representation GGAM* (RSR-GGAM). We state the model for an explicit finite group G and totalize the action oracle over an ambient string space; invalid strings are rejected.

Random Set Representations. Let $\mathsf{GA} = (G, X, \star, \widetilde{x})$ be a finite regular group action. For a label length $\ell = \ell(\lambda)$ with $|X| \le 2^\ell$, sample a uniformly random injection $\sigma_X : X \hookrightarrow \{0,1\}^\ell$. We write $[x]_{\sigma_X} = \sigma_X(x)$ and $\mathcal{L}_{\sigma_X} = \sigma_X(X)$.

Label Slack. We assume sufficient slack to make guessing an unseen valid label negligible. Formally, for every pair of non-negative polynomially bounded functions $p = p(\lambda)$ and $q = q(\lambda)$, for all sufficiently large λ,

$$2^{\ell(\lambda)} > p(\lambda) + q(\lambda) \qquad \text{and} \qquad \frac{q(\lambda)|X_\lambda|}{2^{\ell(\lambda)} - p(\lambda) - q(\lambda)} = \mathrm{negl}(\lambda).$$

Here p bounds the number of initially exposed semantic elements and q the number of action-oracle queries. In particular, it suffices that $\mathrm{poly}(\lambda)|X_\lambda|/2^{\ell(\lambda)}$ be negligible.

Action Oracle and Generic Algorithms. The action oracle

$$\mathcal{O}_\star^{\sigma_X} : G \times \{0,1\}^\ell \longrightarrow \{0,1\}^\ell \cup \{\bot\}$$

is defined by

$$\mathcal{O}_\star^{\sigma_X}(g, L) = \begin{cases} \sigma_X\big(g \star \sigma_X^{-1}(L)\big), & L \in \mathcal{L}_{\sigma_X}, \\ \bot, & L \notin \mathcal{L}_{\sigma_X}. \end{cases}$$

The group G remains explicit, while elements of X are accessed through their random labels; in particular, the public origin is represented by $[\widetilde{x}]_{\sigma_X}$. Since labels are ordinary strings, an RSR-GGAM algorithm may inspect, copy, compare, hash, or otherwise compute on their literal representations, and may submit arbitrary strings to the action oracle. An algorithm is q-query generic if it makes at most q action-oracle queries. It is *pseudoefficient* if this number is polynomially bounded, while its local computation may be unbounded, and *efficient* if its entire running time is polynomially bounded.

Finite Randomness and Instrumented Transcripts. For every security parameter and syntactic input profile, each probabilistic algorithm has a publicly known, effectively computable finite random-tape bound and halts on every tape. Thus its support is effectively enumerable, including for pseudoefficient algorithms. An instrumented execution records every action-oracle query and response and every literal label supplied to or returned by the oracle. When a valid label is first encountered, it may additionally be compared with all previously encountered valid labels, with the resulting equality pattern recorded in the transcript. These local comparisons do not affect the algorithm's control flow or action-query complexity. Invalid oracle inputs are recorded as ordinary strings but are not assigned semantic anchored terms. Instrumentation is available only for computations locally executed by the adversary or used as proof devices; it does not expose a challenger's internal execution.

Designated-Output Validation. Whenever an RSR-GGAM algorithm outputs a structured object containing distinct designated label strings $L_1, \ldots, L_t$, the surrounding experiment validates them by querying $\mathcal{O}_\star^{\sigma_X}(e, L_i)$ for each $i \in [t]$, and accepts the output only if every response equals L_i. These validation queries are included in the action-query complexity. This does not alter well-formed outputs, but ensures that every accepted designated label has been exposed to the action oracle; failures arising from previously unseen valid labels in the lazy-label experiment are charged to the coupling loss analyzed later. Unless explicitly extended, the RSR-GGAM provides only the action interface $\mathcal{O}_\star^{\sigma_X}$ and no additional operation or independent opaque-sampling interface on X.

3.4 The Type-Safe Generic Group Action Model

We also use a type-safe generic group action model (TS-GGAM), in which elements of the acted-on set are represented by opaque typed handles rather than ordinary strings. Our formulation specializes existing type-safe generic models [Mau05, Zha22, BGZ23, OZ24] to finite regular group actions with explicitly represented acting group G.

Handle Semantics. Fix $\mathsf{GA} = (G, X, \star, \widetilde{x})$. Let $\mathcal{H}_X$ be an abstract handle universe and $\sigma_X : X \hookrightarrow \mathcal{H}_X$ an injective semantic assignment, whose image comprises the semantically valid handles. We write $\widehat{x} = \sigma_X(x)$ and $\widehat{x}_0 = \sigma_X(\widetilde{x})$. The assignment σ_X defines handle semantics but is hidden from generic algorithms. The action and equality interfaces are

$$\mathcal{O}_\star^{\sigma_X} : G \times \mathrm{im}(\sigma_X) \to \mathrm{im}(\sigma_X), \qquad \mathcal{O}_\star^{\sigma_X}(g, \widehat{x}) = \sigma_X(g \star x),$$

and

$$\mathsf{Eq}_X(\widehat{x}, \widehat{y}) = 1 \iff \widehat{x} = \widehat{y} \iff x = y.$$

Handles outside $\mathrm{im}(\sigma_X)$ are unreachable in a well-typed execution.

Type-Safe Generic Algorithms. A type-safe algorithm operates on two disjoint data types: explicit data, including elements of G, and opaque objects of type $\mathsf{Handle}[X]$. It may perform arbitrary computation on explicit data, but on X-handles may only (i) store, copy, and forward an existing handle; (ii) compare two existing handles through Eq_X; (iii) invoke $\mathcal{O}_\star^{\sigma_X}(g, \widehat{x})$ for explicit $g \in G$ and an existing handle $\widehat{x}$; and (iv) return a handle in an output position of type $\mathsf{Handle}[X]$. In particular, it cannot inspect or arithmetize a handle, convert it to explicit data, or construct a handle that it has not received or obtained from the action oracle. These are syntactic type restrictions. Unless explicitly provided by the surrounding experiment, there is no independent opaque-sampling interface for X. An algorithm can sample $g \leftarrow G$ and obtain $\mathcal{O}_\star^{\sigma_X}(g, \widehat{x}_0)$, but then knows its displacement g from the origin. More generally, the experiment may provide polynomially many initial handles; these form the *anchors* from which all subsequently encountered handles descend.

Renaming Invariance. Opacity is equivalently captured by invariance under consistent handle renaming. For any bijection $\pi : \mathcal{H}_X \to \mathcal{H}'_X$, let $\sigma'_X = \pi \circ \sigma_X$. Applying π consistently to all input handles, oracle responses, and typed outputs yields an execution under σ'_X with the same probability, explicit outputs, and control-flow decisions. Thus a type-safe algorithm can distinguish handles only through their equality pattern and the action interface. We use this renaming invariance as a semantic convention for the opaque-handle interface.

Equality-Explicit and Equality-Saturated Normal Form. We instrument handle equality through explicit Eq_X calls, which is equivalent to primitive equality of opaque handles and therefore does not change the model. We further use a cached *equality-saturated* execution: whenever an input handle first becomes available or an action query returns a new handle occurrence, it is compared with every previously encountered input or action-response occurrence. The outcomes are recorded but do not affect the original algorithm's control flow or outputs, and each pair is compared at most once. Consequently, an execution beginning with p input handles and making at most q action queries encounters at most $(p + q)$ such occurrences and requires at most $\binom{p+q}{2}$ additional equality comparisons. Equality saturation therefore preserves action-query complexity and incurs only polynomial overhead whenever $p, q = \mathrm{poly}(\lambda)$.

Query Complexity and Pseudoefficiency. We count calls to $\mathcal{O}_\star^{\sigma_X}$ as action queries and treat equality tests as free, as equality of random labels is locally available in the RSR-GGAM. Charging equality as well, as in some prior formulations [BGZ23], changes polynomial query bounds by at most a polynomial factor by the preceding observation. A type-safe algorithm is *pseudoefficient* if it makes polynomially many action queries, even if its local computation is unbounded, and *efficient* if its entire running time is polynomial.

Finite Randomness and Totality. For every security parameter and well-typed syntactic input profile, each probabilistic generic algorithm has a publicly known, effectively computable finite random-tape bound and halts on every random tape. By padding, we may regard it as reading exactly this many bits; hence its support on every fixed well-typed input is finite and effectively enumerable. This convention does not restrict the local computation of pseudoefficient algorithms.

Relation to Prior Formulations. Our TS-GGAM is a specialization of known type-safe generic models [Mau05, Zha22, BGZ23, OZ24], rather than a new generic model. As in the Maurer-style group-action model of [BGZ23], it exposes action and equality interfaces; we specialize to finite regular actions with explicit G, permit polynomially many initial handles supplied by the surrounding experiment, omit independent X-sampling and auxiliary set operations, and count only action queries.

4 Identity-Based Encryption in Generic Group-Action Models

We now instantiate the IBE syntax of Section 3.1 in the RSR-GGAM and TS-GGAM of Sections 3.3 and 3.4, respectively.

4.1 IBE in the Type-Safe GGAM

We instantiate the IBE syntax of Section 3.1 in the TS-GGAM of Section 3.4. All algorithms access the same group-action instance, semantic handle assignment, and action oracle, and the message space is $\{0, 1\}$.

Canonical Handle-Shape Convention. A typed output may contain repeated occurrences of a handle or handles already available to the surrounding experiment. We canonically write such an object as $U = (U_{\mathrm{exp}}, \mathsf{Shape}_U; \widehat{H}_1, \ldots, \widehat{H}_t)$, where the $\widehat{H}_i$ are pairwise distinct X-handles and the explicit Shape_U records which handle occurs in each handle-valued position. For public parameters, the origin $\widehat{x}_0$ is an ambient handle with reserved index 0 and is therefore omitted from the list of non-origin handles. We write $n = n(\mathsf{pp})$ for the number of distinct non-origin handles in pp, and let $n_{\mathsf{pp}}(\lambda)$ and $n_{\mathsf{sk}}(\lambda)$ be publicly known, efficiently computable polynomial bounds on the numbers of distinct non-origin handles in pp and distinct handles in any identity secret key, respectively. We usually absorb Shape_U into U_{exp} and write $U = (U_{\mathrm{exp}}, \widehat{H}_1, \ldots, \widehat{H}_t)$.

Efficiently Indexable Identity Spaces. An identity-space family $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ is *efficiently indexable* if there is a public polynomial-time algorithm $\mathsf{IndexID}$ such that, on input $(1^\lambda, 1^t)$ for polynomially bounded t, it outputs either $\perp$ or an ordered tuple of t pairwise distinct identities in ID_λ, and outputs such a tuple whenever $t \le |\mathsf{ID}_\lambda|$. We assume $\mathsf{ID}_\lambda \neq \emptyset$ for every λ. In particular, $\mathsf{IndexID}(1^\lambda, 1^1) \neq \perp$; our impossibility theorem otherwise invokes the indexer only with $t = n_{\mathsf{pp}}(\lambda) + 1$.

Syntax of TS-GGAM-IBE. Let $\mathsf{GA}_\lambda = (G_\lambda, \mathcal{X}_\lambda, \star_\lambda, \widetilde{x}_\lambda)$ be a family of regular group actions, with public group parameters gpar_λ and public origin handle $\widehat{x}_0 = \sigma(\widetilde{x}_\lambda)$. Let $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of identity spaces.

Definition 5 (TS-GGAM-IBE). *A* TS-GGAM-IBE *scheme, over the identity space* ID *is a tuple* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *of probabilistic type-safe generic-action algorithms with the following syntax.*

$$\begin{aligned}
(\mathsf{msk}, \mathsf{pp}) &\leftarrow \mathsf{Setup}^{\mathcal{O}_\star^{\sigma X}}(1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_0),\\
\mathsf{sk}_{\mathsf{id}} &\leftarrow \mathsf{KeyGen}^{\mathcal{O}_\star^{\sigma X}}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}),\\
C &\leftarrow \mathsf{Enc}^{\mathcal{O}_\star^{\sigma X}}(\mathsf{pp}, \mathsf{id}, m),\\
m' &\leftarrow \mathsf{Dec}^{\mathcal{O}_\star^{\sigma X}}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C),
\end{aligned}$$

where $\mathsf{id} \in \mathsf{ID}_\lambda$, $m \in \{0,1\}$, *and* $m' \in \{0,1,\perp\}$. *In canonical form,*

$$\mathsf{msk} = (\mathsf{msk}_{\mathrm{exp}}, \widehat{M}_1, \ldots, \widehat{M}_{n_{\mathsf{msk}}}), \qquad \mathsf{pp} = (\mathsf{pp}_{\mathrm{exp}}, \widehat{P}_1, \ldots, \widehat{P}_n), \quad n \le n_{\mathsf{pp}}(\lambda),$$

$$\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}_{\mathsf{id},\mathrm{exp}}, \widehat{Z}_1, \ldots, \widehat{Z}_s), \quad s \le n_{\mathsf{sk}}(\lambda), \qquad C = (C_{\mathrm{exp}}, \widehat{T}_1, \ldots, \widehat{T}_{n_{\mathsf{ct}}}).$$

The displayed handles in each object are distinct according to the canonical shape convention above; in particular, the $\widehat{P}_i$ are precisely the distinct non-origin public-parameter handles. All four algorithms use the same group-action instance, semantic assignment, and action oracle, and make polynomially many action queries.

Definition 6 (Correctness). *A TS-GGAM-IBE scheme has decryption error* $\epsilon = \epsilon(\lambda)$ *if, for every* $\lambda \in \mathbb{N}$, *every* $\mathsf{id} \in \mathsf{ID}_\lambda$, *and every* $m \in \{0,1\}$,

$$\Pr\Big[\mathsf{Dec}^{\mathcal{O}_\star^{\sigma}}\big(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C\big) = m\Big] \ge 1 - \epsilon(\lambda)$$

where $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}^{\mathcal{O}_\star^{\sigma}}(1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_0)$, $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}^{\mathcal{O}_\star^{\sigma}}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$, *and* $C \leftarrow \mathsf{Enc}^{\mathcal{O}_\star^{\sigma}}(\mathsf{pp}, \mathsf{id}, m)$, *and where the probability is taken over the internal randomness of* Setup, KeyGen, Enc, *and* Dec.

We say that Π is *perfectly correct* if $\epsilon(\lambda) = 0$, and that it is *correct* if $\epsilon(\lambda)$ is negligible.

Remark 1 (Pointwise Correctness). For some intermediate results, it will be convenient to consider the stronger pointwise correctness requirement under which the preceding inequality holds for every $(\mathsf{msk}, \mathsf{pp})$ in the support of $\mathsf{Setup}^{\mathcal{O}_\star^{\sigma}}$. Our final result will use the standard averaged notion in Definition 6.

Fully Selective Security of TS-GGAM-IBE. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a type-safe generic-action adversary. The state passed from $\mathcal{A}_1$ to $\mathcal{A}_2$ is typed and may contain explicit values and set-element handles.

Definition 7 (Fully Selective Security Experiment for TS-GGAM-IBE). *The fully selective security experiment* $\mathsf{Expt}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda)$ *is defined as follows.*

1. *Instantiate* $\mathsf{GA}_\lambda = (G_\lambda, \mathcal{X}_\lambda, \star_\lambda, \widetilde{x}_\lambda)$*, fix an injective semantic handle assignment* $\sigma : \mathcal{X}_\lambda \to \mathcal{H}_\lambda$*, and set* $\widehat{x}_0 = \sigma(\widetilde{x}_\lambda)$*.*
2. *Run the first stage of the adversary:* $(\mathsf{id}^\star, \mathsf{id}_1, \ldots, \mathsf{id}_q, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathcal{O}^\sigma_\star}(1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_0)$*, where* $\mathsf{id}^\star, \mathsf{id}_1, \ldots, \mathsf{id}_q \in \mathsf{ID}_\lambda$ *and* $\mathsf{id}^\star \notin \{\mathsf{id}_1, \ldots, \mathsf{id}_q\}$*. If these conditions are not satisfied, output an independent uniformly random bit and halt.*
3. *Generate* $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}^{\mathcal{O}^\sigma_\star}(1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_0)$*.*
4. *For every* $i \in [q]$*, generate* $\mathsf{sk}_i \leftarrow \mathsf{KeyGen}^{\mathcal{O}^\sigma_\star}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_i)$*.*
5. *Sample a uniformly random challenge bit* $b \leftarrow \{0,1\}$ *and generate* $C^\star \leftarrow \mathsf{Enc}^{\mathcal{O}^\sigma_\star}(\mathsf{pp}, \mathsf{id}^\star, b)$*.*
6. *Run the second stage of the adversary:* $b' \leftarrow \mathcal{A}_2^{\mathcal{O}^\sigma_\star}(\mathsf{state}, \mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_q, C^\star)$*.*
7. *Output* 1 *if* $b' = b$*, and output* 0 *otherwise.*

The number of extraction identities $q = q(\lambda)$ is required to be polynomially bounded. Since the adversary is type-safe, it may pass previously obtained handles between its two stages, but it cannot inspect their representations or cast ordinary strings as handles.

Definition 8 (Fully Selective Security for TS-GGAM-IBE). *The fully selective advantage of* $\mathcal{A}$ *against* Π *in the fully selective security experiment above is*

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right|.$$

A TS-GGAM-IBE scheme Π *is* fully selective-secure *if, for every pseudoefficient type-safe generic-action adversary* $\mathcal{A}$*, the function* $\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible in* λ*.*

Remark 2. The above security requirement is independent of the particular naming of the type-safe handles. Indeed, the renaming invariance of the scheme and the adversary implies that replacing σ by $\pi \circ \sigma$, for any bijection π of the handle space, does not change the experiment's output distribution.

Restricted TS-GGAM-IBE. As an intermediate step of our proof, we present an impossibility result that only considers TS-GGAM-IBE schemes whose identity secret keys do not contain actionable set-element handles.

Definition 9 (Restricted TS-GGAM-IBE). *A TS-GGAM-IBE scheme as in Definition 5 is* restricted *if every identity secret key is entirely explicit, i.e., for every* $\mathsf{id} \in \mathsf{ID}_\lambda$*,* $\mathsf{sk}_{\mathsf{id}} = \mathsf{sk}_{\mathsf{id},\mathsf{exp}}$ *and* $n_{\mathsf{sk}}(\lambda) = 0$*. In particular, no component of* $\mathsf{sk}_{\mathsf{id},\mathsf{exp}}$ *can subsequently be supplied to* $\mathcal{O}^\sigma_\star$ *as a set-element handle.*

The restriction applies only to identity secret keys; the master secret key, public parameters, and ciphertexts may still contain X-handles. We later remove this restriction, with a controlled correctness loss, without changing the identity space or the number of public-parameter handles.

4.2 IBE in the RSR-GGAM

We analogously instantiate IBE in the RSR-GGAM of Section 3.3. All algorithms access the same group-action instance, random representation σ_X, and action oracle. Unlike in the TS-GGAM, labels are ordinary strings: algorithms may inspect and compute on their bits and may submit arbitrary strings of the appropriate length to the action oracle.

Canonical label-shape convention. We canonically write an output containing designated set-element labels as $U = (U_{\mathrm{exp}}, \mathsf{Shape}_U; L_1, \ldots, L_t)$, where $L_1, \ldots, L_t \in \mathrm{im}(\sigma_X)$ are pairwise distinct and the explicit Shape_U records their occurrences. For public parameters, the origin label $L_0 = [\widetilde{x}]_{\sigma_X}$ has reserved index 0 and is omitted from the list of non-origin labels. We usually absorb Shape_U into U_{exp} and write $U = (U_{\mathrm{exp}}, L_1, \ldots, L_t)$. This convention designates the output positions intended to represent elements of X; it does *not* restrict U_{exp} from containing other strings, including values computed from labels, nor does it restrict which strings an algorithm may submit to the action oracle. We reuse the efficiently indexable identity spaces of Section 4.1, and let $n_{\mathsf{pp}}(\lambda)$ and $n_{\mathsf{sk}}(\lambda)$ be polynomial bounds on the numbers of distinct designated non-origin labels in pp and distinct designated labels in any identity secret key, respectively.

Syntax of RSR-GGAM-IBE. Let $\mathsf{GA}_\lambda = (G_\lambda, \mathcal{X}_\lambda, \star_\lambda, \widetilde{x}_\lambda)$ be a family of regular group actions, with public group parameters gpar_λ. Let $\sigma_X \colon \mathcal{X}_\lambda \hookrightarrow \{0,1\}^{\ell(\lambda)}$ be a uniformly random injective representation, and let $L_0 = [\widetilde{x}_\lambda]_{\sigma_X} = \sigma_X(\widetilde{x}_\lambda)$ be the public origin label.

Definition 10 (RSR-GGAM-IBE). *An* RSR-GGAM-IBE scheme *over* ID *is a tuple* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *of probabilistic RSR-GGAM algorithms with syntax*

$$\begin{aligned}(\mathsf{msk}, \mathsf{pp}) &\leftarrow \mathsf{Setup}^{\mathcal{O}_\star^{\sigma_X}}(1^\lambda, \mathsf{gpar}_\lambda, L_0),\\ \mathsf{sk}_{\mathsf{id}} &\leftarrow \mathsf{KeyGen}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}),\\ C &\leftarrow \mathsf{Enc}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{id}, m),\\ m' &\leftarrow \mathsf{Dec}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C),\end{aligned}$$

where $\mathsf{id} \in \mathsf{ID}_\lambda$, $m \in \{0,1\}$, *and* $m' \in \{0,1,\perp\}$. *In canonical form,*

$$\mathsf{msk} = (\mathsf{msk}_{\mathrm{exp}}, M_1, \ldots, M_{n_{\mathsf{msk}}}), \qquad \mathsf{pp} = (\mathsf{pp}_{\mathrm{exp}}, P_1, \ldots, P_n), \quad n \le n_{\mathsf{pp}}(\lambda),$$

$$\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}_{\mathsf{id},\mathrm{exp}}, Z_1, \ldots, Z_s), \quad s \le n_{\mathsf{sk}}(\lambda), \qquad C = (C_{\mathrm{exp}}, T_1, \ldots, T_{n_{\mathrm{ct}}}).$$

The displayed entries are distinct designated valid labels according to the convention above. Each algorithm makes polynomially many action queries but may otherwise compute arbitrarily on label strings and submit arbitrary strings in $\{0,1\}^{\ell(\lambda)}$ to $\mathcal{O}_\star^{\sigma_X}$.

Definition 11 (Correctness). *An RSR-GGAM-IBE scheme has decryption error* $\epsilon = \epsilon(\lambda)$ *if, for every* $\lambda \in \mathbb{N}$, *every* $\mathsf{id} \in \mathsf{ID}_\lambda$, *and every* $m \in \{0,1\}$,

$$\Pr\left[\mathsf{Dec}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C) = m\right] \ge 1 - \epsilon(\lambda),$$

where $\sigma_X \leftarrow \mathrm{Inj}\big(\mathcal{X}_\lambda, \{0,1\}^{\ell(\lambda)}\big)$, $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}^{\mathcal{O}_\star^{\sigma_X}}(1^\lambda, \mathsf{gpar}_\lambda, [\widetilde{x}_\lambda]_{\sigma_X})$, $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$, *and* $C \leftarrow \mathsf{Enc}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{id}, m)$. *The probability is over the choice of the random representation and the internal randomness of all four algorithms.*

We say that Π is perfectly correct if $\epsilon(\lambda) = 0$, and that it is correct if ϵ is negligible.

Remark 3 (Pointwise correctness). For intermediate results, one may consider the stronger requirement that the correctness inequality hold after fixing the representation σ_X and every $(\mathsf{msk}, \mathsf{pp})$ in the support of setup. The final impossibility statement will use the standard averaged correctness notion of Definition 11.

Fully Selective Security of RSR-GGAM-IBE. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an RSR-GGAM adversary. Its state consists entirely of ordinary data and may, in particular, contain valid labels, invalid strings, or arbitrary strings computed from previously observed labels.

Definition 12 (Fully Selective Security Experiment for RSR-GGAM-IBE). *The fully selective security experiment* $\mathsf{Expt}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda)$ *is defined as follows.*

1. *Instantiate* $\mathsf{GA}_\lambda = (G_\lambda, \mathcal{X}_\lambda, \star_\lambda, \widetilde{x}_\lambda)$*, sample a uniformly random injection* $\sigma_X \leftarrow \mathrm{Inj}\big(\mathcal{X}_\lambda, \{0,1\}^{\ell(\lambda)}\big)$*, and set* $L_0 = [\widetilde{x}_\lambda]_{\sigma_X}$*.*
2. *Run the first stage of the adversary:* $(\mathsf{id}^\star, \mathsf{id}_1, \ldots, \mathsf{id}_q, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathcal{O}_\star^{\sigma_X}}(1^\lambda, \mathsf{gpar}_\lambda, L_0)$*, subject to the requirement that* $\mathsf{id}^\star, \mathsf{id}_1, \ldots, \mathsf{id}_q \in \mathsf{ID}_\lambda$ *and* $\mathsf{id}^\star \notin \{\mathsf{id}_1, \ldots, \mathsf{id}_q\}$*. If these conditions fail, output an independent uniformly random bit and halt.*
3. *Generate* $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}^{\mathcal{O}_\star^{\sigma_X}}(1^\lambda, \mathsf{gpar}_\lambda, L_0)$*.*
4. *For every* $i \in [q]$*, generate* $\mathsf{sk}_i \leftarrow \mathsf{KeyGen}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_i)$*.*
5. *Sample* $b \leftarrow \{0,1\}$ *and generate* $C^\star \leftarrow \mathsf{Enc}^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{pp}, \mathsf{id}^\star, b)$*.*
6. *Run the second stage of the adversary:* $b' \leftarrow \mathcal{A}_2^{\mathcal{O}_\star^{\sigma_X}}(\mathsf{state}, \mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_q, C^\star)$*.*
7. *Output* 1 *if* $b' = b$*, and output* 0 *otherwise.*

All algorithms and both stages of the adversary access the same random representation and action oracle. The number $q = q(\lambda)$ of extraction identities is required to be polynomially bounded.

Definition 13 (Fully Selective Security for RSR-GGAM-IBE). *The fully selective advantage of* $\mathcal{A}$ *against* Π *is*

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right|.$$

An RSR-GGAM-IBE scheme Π *is* fully selective-secure *if* $\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}$ *is negligible for every pseudoefficient RSR-GGAM adversary* $\mathcal{A}$*.*

Remark 4. Unlike in the TS-GGAM, neither the scheme nor the adversary is required to be renaming invariant: its behavior may depend arbitrarily on the literal labels selected by σ_X.

Restricted Key-Label-Oblivious RSR-GGAM-IBE. Our intermediate RSR impossibility result requires not only that identity secret keys contain no designated labels, but also that labels allocated privately during setup and key generation leave no information in the retained state. Consider, in the lazy-label formulation of Section 7.1, a prefix consisting of setup followed by polynomially many key-generation invocations. At its end, let D_{pub} contain the origin and all semantic elements represented by designated labels retained in pp and the generated secret keys, and let D_{priv} contain every other semantic element whose label was allocated during the prefix. The master secret key and unused local states are discarded; the retained ordinary state consists of the explicit components and canonical shapes of pp and the generated secret keys.

Private-Label Symmetry. Consider a completed lazy-label prefix. Let D_{pub} and D_{priv} denote, respectively, the semantic elements whose labels are retained and the semantic elements whose labels were allocated during the prefix but are not retained. Write $\mathsf{LT}_{\mathsf{pub}} = \mathsf{LT}|_{D_{\mathsf{pub}}}$ and $\mathsf{LT}_{\mathsf{priv}} = \mathsf{LT}|_{D_{\mathsf{priv}}}$. Let K denote the retained ordinary state and τ_{sem} the semantic prefix transcript obtained by replacing labels by their semantic elements while retaining ordinary values, control flow, and equality outcomes.

Definition 14 (Private-Label Symmetry). *A lazy-label prefix satisfies* private-label symmetry *if, for every fixed value* $\big(K, \tau_{\mathsf{sem}}, \mathsf{LT}_{\mathsf{pub}}, \mathsf{Rej}, D_{\mathsf{priv}}\big)$ *having positive probability, the conditional distribution of* $\mathsf{LT}_{\mathsf{priv}}$ *is uniform over all injections* $D_{\mathsf{priv}} \hookrightarrow \Sigma_\lambda \setminus (\mathrm{im}(\mathsf{LT}_{\mathsf{pub}}) \cup \mathsf{Rej})$*.*

Definition 15 (Restricted Key-Label-Oblivious RSR-GGAM-IBE). *An RSR-GGAM-IBE scheme* Π *is* restricted key-label-oblivious *if the following conditions hold.*

1. *Every identity secret key contains no designated set-element labels. Equivalently,* $\mathsf{sk}_{\mathsf{id}} = \mathsf{sk}_{\mathsf{id,exp}}$ *and* $n_{\mathsf{sk}}(\lambda) = 0$ *for every* λ *and every* $\mathsf{id} \in \mathsf{ID}_\lambda$*.*
2. *For every polynomially bounded sequence of key-generation identities, the resulting setup-key-generation prefix satisfies private-label symmetry in the sense of Definition 14, with* D_{pub} *and* D_{priv} *defined above.*

The first condition alone would not suffice: because secret keys are ordinary bit strings, their explicit components could retain a private label or label-derived information. Private-label symmetry rules out such residual information by requiring forgotten private labels to remain uniformly distributed, conditioned on the retained state. This is a substantive restriction of RSR-GGAM-IBE; it does not restrict computation on public or ciphertext labels.

Composable Label Sanitizability. The preceding condition controls private labels created during setup and key generation. To separate the polynomially many subsequent encryption–decryption computations used by our attack, we require the same symmetry at composition boundaries. A *label-erasing controller* schedules polynomially many invocations of $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}$, and Dec and polynomially many action queries, while possibly performing unbounded local computation. At an invocation boundary it may retain live interface objects and their designated labels, declared explicit outputs, canonical semantic relations extracted from instrumented transcripts, and ordinary control state independent of transient literal labels. Equivalently, after fixing the declared explicit outputs and live interface objects, its scheduling, control flow, and retained state are invariant under every permutation of transient labels fixing the live labels. Thus transient labels may be forwarded, equality-tested, and used to extract semantic relations, but their literal representations or values derived therefrom cannot otherwise be retained by the controller. This restriction does not constrain label-dependent computation internal to a scheme algorithm; any resulting declared explicit output is instead included in the retained state below. At an invocation boundary of the corresponding lazy-label execution, let K be the complete retained ordinary state and τ_{sem} the semantic prefix transcript, and partition the allocated semantic elements as $\mathrm{dom}(\mathsf{LT}) = D_{\mathsf{live}} \dot\cup D_{\mathsf{priv}}$, where D_{live} consists of elements represented by designated labels in live interface objects. Write $\mathsf{LT}_{\mathsf{live}} = \mathsf{LT}|_{D_{\mathsf{live}}}$ and $\mathsf{LT}_{\mathsf{priv}} = \mathsf{LT}|_{D_{\mathsf{priv}}}$.

Definition 16 (Composable Label Sanitizability). *An RSR-GGAM-IBE scheme is* composably label-sanitizable *if the following holds for every polynomially bounded label-erasing composition of its algorithms and every invocation boundary in that composition: for every fixed value* $\big(K, \tau_{\mathsf{sem}}, \mathsf{LT}_{\mathsf{live}}, \mathsf{Rej}, D_{\mathsf{priv}}\big)$ *having positive probability, the conditional distribution of* $\mathsf{LT}_{\mathsf{priv}}$ *is uniform over all injections* $D_{\mathsf{priv}} \hookrightarrow \Sigma_\lambda \setminus \big(\mathrm{im}(\mathsf{LT}_{\mathsf{live}}) \cup \mathsf{Rej}\big)$. *The requirement also applies to executions obtained by inserting sanitization steps at an arbitrary subset of earlier invocation boundaries.*

The final clause provides the composability required by our later hybrid argument. The condition does not impose type safety: scheme algorithms may inspect, branch on, or otherwise compute with literal public and ciphertext labels. It requires only that labels discarded at a sanitization boundary remain uniformly permutable after conditioning on the complete retained semantic and ordinary state. Our RSR-GGAM impossibility theorem applies to restricted key-label-oblivious RSR-GGAM-IBE schemes that are additionally composably label-sanitizable.

5 Impossibility of Restricted TS-GGAM-IBE

This section develops the anchored-transcript framework underlying our impossibility results and applies it first to restricted TS-GGAM-IBE. We show that, subject to the correctness and security losses quantified below, such a scheme cannot support more identities than the number of non-origin X-handles in its public parameters.

5.1 Anchored Transcripts and Gain Graphs

The action oracle can translate a known X-element by an explicit group element, but cannot combine independently supplied X-elements. Hence every derived handle remains a known translate of an input *anchor*, while positive equalities between differently anchored handles reveal relative-action relations among the anchors. We encode these relations by gain graphs, whose rank bounds the number of independent relations exposed by a generic computation. Throughout, let $(G, X, \star)$ be a regular left group action and $\sigma : X \to \mathcal{H}$ a type-safe injective encoding.

Gain Graphs. We first introduce the graph-theoretic language used to represent relations among set-element handles.

Definition 17 (G-Gain Graph). *A* G-gain graph *is a tuple* $\Gamma = (V, E, \gamma)$ *where* (V, E) *is a directed graph and* $\gamma : E \longrightarrow G$ *assigns a group element, called a* gain*, to every oriented edge. For every oriented edge* $(u, v) \in E$*, we require that* $(v, u) \in E$ *and* $\gamma(v, u) = \gamma(u, v)^{-1}$*. The intended meaning of an edge* (u, v) *with gain* g *is the relative-action relation* $x_v = g \star x_u$ *between the set elements associated with vertices* u *and* v*.*

For a directed path $P = (v_0, v_1, \ldots, v_\ell)$ in Γ, define its gain as

$$\gamma(P) = \gamma(v_{\ell-1}, v_\ell) \circ \cdots \circ \gamma(v_1, v_2) \circ \gamma(v_0, v_1).$$

Thus, if all edges along P are satisfied by a collection of set elements $\{x_v\}_{v \in V}$, then $x_{v_\ell} = \gamma(P) \star x_{v_0}$.

Definition 18 (Realization and Balancedness). *Let $\Gamma = (V, E, \gamma)$ be a G-gain graph as in Definition 17.*

- *We say that a tuple $\boldsymbol{x} = (x_v)_{v \in V} \in \mathcal{X}^V$* realizes *$\Gamma$ if, for every $(u, v) \in E$, $x_v = \gamma(u, v) \star x_u$.*
- *We say that the gain graph Γ is* balanced *if every closed walk P in Γ has gain $\gamma(P) = e$.*

Every realizable gain graph is balanced: a closed walk based at v gives $x_v = \gamma(P) \star x_v$, and freeness forces $\gamma(P) = e$. Consequently, in a balanced graph all paths between two connected vertices have the same gain.

Definition 19 (Gain-Graph Closure). *Let $\Gamma = (V, E, \gamma)$ be a balanced G-gain graph. We define the* closure *of Γ as follows:*

$$\mathrm{cl}(\Gamma) = \left\{ (u, v, g) \in V \times V \times G \;\middle|\; \begin{array}{l} \text{there exists a path } P \text{ from } u \text{ to } v \\ \text{such that } \gamma(P) = g \end{array} \right\}.$$

Thus, $(u, v, g) \in \mathrm{cl}(\Gamma)$ represents the relation $x_v = g \star x_u$ implied by the edges of Γ.

We allow paths of length zero and define their gain to be e. Consequently, $(u, u, e) \in \mathrm{cl}(\Gamma)$ for every $u \in V$.

Definition 20 (Gain-Graph Rank). *Let $\Gamma = (V, E, \gamma)$ be a balanced G-gain graph, and let $c(\Gamma)$ denote the number of connected components in its underlying undirected graph. We define* $\mathrm{rank}(\Gamma) = |V| - c(\Gamma)$. *Equivalently,* $\mathrm{rank}(\Gamma)$ *is the number of edges in any spanning forest of the underlying graph.*

Adding a balanced edge increases the rank exactly when it joins two previously disconnected components. If its endpoints are already connected, its gain is forced by balancedness and the edge already lies in the closure. In particular, a gain graph on $(d+1)$ vertices has rank at most d.

Anchored Terms. Consider an invocation of a type-safe generic-action algorithm $\mathcal{B}$ whose distinct input handles are $\widehat{x}_0, \widehat{x}_1, \ldots, \widehat{x}_d$, where $\widehat{x}_i = \sigma(x_i)$ for $x_i \in \mathcal{X}$. Repeated occurrences of the same input handle are identified and represented by a single anchor. We call $x_0, x_1, \ldots, x_d$ the *anchors* of the computation.

Definition 21 (Anchored Term). *An* anchored term *is defined as a pair of the form $(i, g) \in [0, d] \times G$. Relative to an anchor tuple $\boldsymbol{x} = (x_0, \ldots, x_d)$, the semantics of (i, g) is $[\![(i, g)]\!]_{\boldsymbol{x}} = g \star x_i$.*

Each input occurrence $\widehat{x}_i$ receives (i, e). Copies preserve their term, and an action query by $h \in G$ on a handle carrying (i, g) returns a handle carrying $(i, h \circ g)$. Hence every input or action-response occurrence has a syntactically determined anchored term. Distinct terms may nevertheless denote the same semantic handle; precisely these coincidences reveal relations between anchors. Crucially, no action query produces a term depending on more than one anchor.

Transcript Gain Graph. We next formalize the claim that the algebraic information contained in a type-safe generic-action transcript is captured by a bounded-rank gain graph. Fix the random tape of a potentially randomized type-safe generic-action algorithm $\mathcal{B}$, so that $\mathcal{B}$ becomes deterministic. Let $\mathsf{Tr}_{\mathcal{B}}(\boldsymbol{x})$ denote its complete execution transcript on explicit input w and input handles $\sigma(x_0), \ldots, \sigma(x_d)$, as in Section 3.4. Let $\mathcal{Q}_{\mathsf{Tr}} \subseteq [0, d] \times G$ be the set of anchored terms occurring in the transcript Tr.

Definition 22 (Transcript Gain Graph). *Let* Tr *be a cached equality-saturated transcript with anchored-term set $Q_{\mathrm{Tr}} \subseteq [0, d] \times G$. The* transcript gain graph $\Gamma_{\mathrm{Tr}} = (V_{\mathrm{Tr}}, E_{\mathrm{Tr}}, \gamma_{\mathrm{Tr}})$ *is defined as follows: (i) the graph has vertex set $V_{\mathrm{Tr}} = [0, d]$; (ii) for every ordered pair (i, j) for which there exist terms $(i, g), (j, h) \in Q_{\mathrm{Tr}}$ whose corresponding handle occurrences are equal, the graph contains a single edge (i, j), whose gain is $\gamma_{\mathrm{Tr}}(i, j) = h^{-1} \circ g$; and (iii) The graph also contains the reverse edge (j, i) with gain $\gamma_{\mathrm{Tr}}(j, i) = \gamma_{\mathrm{Tr}}(i, j)^{-1}$.*

Observe that the gain assigned to (i, j) is well-defined. Indeed, suppose two equal term pairs yield gains $a, b \in G$ from i to j. The anchor tuple then satisfies $x_j = a \star x_i$ and $x_j = b \star x_i$. It follows that $(b^{-1} \circ a) \star x_i = x_i$, and freeness of the action gives $a = b$. Thus repeated observations of the same relation produce the same directed edge and are included only once.

Lemma 1 (Anchored-Transcript Lemma). *Let $\mathcal{B}$ be a deterministic type-safe generic-action algorithm with explicit input w and distinct input handles $\sigma(x_0), \ldots, \sigma(x_d)$. Let $\mathsf{Tr} = \mathsf{Tr}_{\mathcal{B}}(\boldsymbol{x})$ be its execution transcript and let Γ_{Tr} be its transcript gain graph. Then:*

1. **Anchored Provenance.** *Every input or action-response handle occurrence appearing in* Tr *has a designated anchored term $(i, g) \in Q_{\mathrm{Tr}}$ satisfying $\widehat{z} = \sigma(g \star x_i)$. The term is determined syntactically by the originating input occurrence and the sequence of action queries along its provenance. Distinct occurrences representing the same semantic handle may have different designated terms.*
2. **Realization and Balancedness.** *The anchor tuple $\boldsymbol{x} = (x_0, \ldots, x_d)$ realizes Γ_{Tr}. Consequently, Γ_{Tr} is balanced.*
3. **Characterization of Positive Equalities.** *For every $(i, g), (j, h) \in \mathcal{Q}_{\mathsf{Tr}}$, $\sigma(g \star x_i) = \sigma(h \star x_j)$ if and only if $(i, j, h^{-1} \circ g) \in \mathrm{cl}(\Gamma_{\mathsf{Tr}})$.*
4. **Rank Bound.** *The number of independent relative-action relations exposed by the transcript is at most d, i.e., $\mathrm{rank}(\Gamma_{\mathsf{Tr}}) \le d$. More precisely, a newly observed positive equality is not implied by the preceding positive equalities only if it connects two previously disconnected anchor components.*
5. **Transcript Transfer.** *Let $\boldsymbol{y} = (y_0, \ldots, y_d) \in \mathcal{X}^{d+1}$ be another anchor tuple satisfying, for every pair $(i, g), (j, h) \in \mathcal{Q}_{\mathsf{Tr}}$,*

$$g \star x_i = h \star x_j \quad \Longleftrightarrow \quad g \star y_i = h \star y_j. \tag{1}$$

Then the executions of $\mathcal{B}$ on $\boldsymbol{x}$ and $\boldsymbol{y}$ are identical up to a consistent renaming of typed handles. In particular, the two executions issue the same sequence of explicit action queries, follow the same control-flow branches, produce identical explicit outputs, and assign the same anchored terms to corresponding output handles.

Proof. We prove the five statements in order.

Anchored Provenance. Assign (i, e) to the ith input occurrence and preserve terms under copying. If an occurrence carrying (i, g) is acted on by h, then $\mathcal{O}^{\sigma}_{\star}(h, \sigma(g \star x_i)) = \sigma((h \circ g) \star x_i)$, so assign $(i, h \circ g)$ to the response. Induction over action queries proves the claim; if distinct provenances reach the same semantic handle, both terms are retained and equality saturation records their coincidence.

Realization and Balancedness. Equality of occurrences carrying (i, g) and (j, h) implies, by injectivity, $g \star x_i = h \star x_j \Longrightarrow x_j = (h^{-1} \circ g) \star x_i$, which is exactly the corresponding edge relation. Thus $\mathbf{x}$ realizes Γ_{Tr}, and realizability implies balancedness.

Characterization of Positive Equalities. The forward implication follows directly from the definition of Γ_{Tr}. Conversely, if $(i, j, h^{-1} \circ g) \in \mathrm{cl}(\Gamma_{\mathrm{Tr}})$, realization gives $x_j = (h^{-1} \circ g) \star x_i$, hence $h \star x_j = g \star x_i$ and injectivity of σ yields the claimed handle equality.

Rank Bound. Γ_{Tr} has $(d+1)$ vertices, so $\mathrm{rank}(\Gamma_{\mathrm{Tr}}) = (d+1) - c(\Gamma_{\mathrm{Tr}}) \le d$. Moreover, if a new positive-equality edge joins already connected vertices, balancedness forces its gain to equal that of the existing path; the relation therefore already lies in the preceding closure.

Transcript Transfer. Couple the two executions inductively, pairing input handles with the same anchor index. Maintain identical explicit states, an equality-preserving renaming of encountered handles, and identical anchored terms on corresponding occurrences. Type safety implies that the next operation depends only on the explicit state and handle-equality pattern. Condition (1) preserves every equality outcome. For an action query by $a \in G$ on a term (i, g), both responses receive $(i, a \circ g)$, and the same condition ensures that the response coincides with a previously encountered handle in one execution exactly when its counterpart does so in the other. Hence the renaming extends consistently and the invariant holds throughout the execution, proving the claim.

This completes the proof of Lemma 1.

Corollary 1 (Bounded Relation Rank). *Consider a type-safe generic-action computation whose distinct input handles consist of the public origin and n additional set-element handles. The computation can expose at most n independent positive relative-action relations among these handles:* $\mathrm{rank}(\Gamma_{\mathsf{Tr}}) \le n$.

The above bound holds regardless of the algorithm's local computational power, the adaptivity of its oracle queries, or the algebraic structure of G.

5.2 Sampling Relative-Action Relations

The anchored-transcript lemma bounds the rank of the positive relations exposed by a generic computation. We now derive two sampling consequences: repeated independent executions rarely reveal a relation outside the accumulated closure, while a uniformly chosen position in an arbitrary relation sequence has noticeable probability of being conditionally closure-contained.

Relation Matroid of an Anchor Tuple. Fix an anchor tuple $\mathbf{x} = (x_0, \dots, x_d) \in X^{d+1}$. By regularity, for every $i \neq j$ there is a unique $\delta_{i,j} \in G$ satisfying $x_j = \delta_{i,j} \star x_i$. For $0 \le i < j \le d$, define the canonical relation $e_{i,j} = (i, j, \delta_{i,j})$, and let $E_{\mathbf{x}} = \{e_{i,j} : 0 \le i < j \le d\}$.

Definition 23 (Relation Matroid). *The* relation matroid *of $\boldsymbol{x}$, denoted $\mathcal{M}_{\boldsymbol{x}}$, is the graphic matroid on ground set $\mathcal{E}_{\boldsymbol{x}}$. Thus, a relation set $R \subseteq \mathcal{E}_{\boldsymbol{x}}$ is independent if and only if the corresponding undirected edges form a forest.*

For $R \subseteq E_{\mathbf{x}}$, write $\mathsf{rank}_{\mathbf{x}}(R)$ and $\mathrm{cl}_{\mathbf{x}}(R)$ for its matroid rank and closure. Explicitly, $\mathsf{rank}_{\mathbf{x}}(R) = (d+1) - c(R)$, and $e_{i,j} \in \mathrm{cl}_{\mathbf{x}}(R)$ iff i and j are connected in $([0, d], R)$. Since all canonical relations are simultaneously realized by $\mathbf{x}$, this closure agrees with gain-graph closure; in particular, $\mathsf{rank}(M_{\mathbf{x}}) = d$.

Remark 5. The matroid is only an analysis object: an algorithm need not know $\delta_{i,j}$ for an unobserved relation. Once a positive equality between terms (i, g) and (j, h) is observed, however, the corresponding displacement $h^{-1} \circ g$ is explicit.

Definition 24 (Relation Set of a Transcript). *Let Tr be a generic-action transcript on the anchor tuple $\boldsymbol{x}$, and let Γ_{Tr} be its transcript gain graph. We define $\mathsf{Rel}(\mathsf{Tr}) \subseteq \mathcal{E}_{\boldsymbol{x}}$ to be the set of canonical relations $e_{i,j}$ for which Γ_{Tr} contains a positive relation between the distinct anchors i and j.*

In the above definition, observations are included once, using the canonical orientation $i < j$. By Lemma 1, $\mathrm{rk}_{\boldsymbol{x}}\big(\mathsf{Rel}(\mathsf{Tr})\big) \le d$.

A General Repeated-Sampling Lemma. We first prove the sampling statement for an arbitrary finite matroid.

Lemma 2 (Repeated-Sampling Closure Lemma). *Let $\mathcal{M} = (\mathcal{E}, \mathcal{I})$ be a finite matroid of rank at most r, and let $R_1, \dots, R_M$ be independent and identically distributed random variables taking values in $2^{\mathcal{E}}$. Then*

$$\Pr\left[R_M \not\subseteq \mathrm{cl}_{\mathcal{M}}(R_1 \cup \cdots \cup R_{M-1})\right] \le \frac{r}{M}.$$

Proof. For $i \in \{0, \dots, M\}$, define $U_i = R_1 \cup \cdots \cup R_i$, where $U_0 = \emptyset$. Let $B_i = \{R_i \not\subseteq \mathrm{cl}_{\mathcal{M}}(U_{i-1})\}$. By the defining properties of matroid closure,

$$B_i \quad \Longleftrightarrow \quad \mathrm{rk}_{\mathcal{M}}(U_i) > \mathrm{rk}_{\mathcal{M}}(U_{i-1}).$$

Thus, whenever B_i occurs, the rank of the accumulated relation set increases by at least one. Since the rank of $\mathcal{M}$ is at most r, for every realization of $R_1, \dots, R_M$, we have $\sum_{i=1}^{M} \mathbf{1}[B_i] \le r$. Taking expectations gives

$$\sum_{i=1}^{M} \Pr[B_i] \le r. \tag{2}$$

We next prove that the sequence $p_i = \Pr[B_i]$ is non-increasing. Let $\mathcal{D}$ denote the common distribution of the R_i, and define, for every $S \subseteq E$, $f(S) = \Pr_{R \leftarrow \mathcal{D}}\big[R \not\subseteq \mathrm{cl}_{\mathcal{M}}(S)\big]$. The function f is antitone with respect to set inclusion. Indeed, if $S \subseteq T$, then monotonicity of matroid closure gives $\mathrm{cl}_{\mathcal{M}}(S) \subseteq \mathrm{cl}_{\mathcal{M}}(T)$, and therefore

$f(T) \le f(S)$. Since R_i is independent of $U_{i-1} = R_1 \cup \cdots \cup R_{i-1}$, we have $p_i = \mathbb{E}\big[f(U_{i-1})\big]$. Moreover, under the natural coupling, $U_{i-1} \subseteq U_i$. Consequently, for every $i \in [M-1]$,

$$p_{i+1} = \mathbb{E}\big[f(U_i)\big] \le \mathbb{E}\big[f(U_{i-1})\big] = p_i.$$

Thus, $p_1 \ge p_2 \ge \cdots \ge p_M$. Consequently, $M \cdot p_M \le \sum_{i=1}^{M} p_i \le r$, where the final inequality follows from Equation (2). Therefore, $p_M \le r/M$, which is the claimed bound. This concludes the proof of Lemma 2.

Applying Lemma 2 to the relation matroid of an anchor tuple gives the following corollary.

Corollary 2 (Repeated Sampling of Generic-Action Transcripts). *Fix an anchor tuple $\boldsymbol{x} = (x_0, \ldots, x_d)$. Let $\mathsf{Tr}_1, \ldots, \mathsf{Tr}_M$ be independent and identically distributed generic-action transcripts on $\boldsymbol{x}$, and let $R_i = \mathsf{Rel}(\mathsf{Tr}_i)$. Then*

$$\Pr\left[R_M \not\subseteq \mathrm{cl}_{\boldsymbol{x}}(R_1 \cup \cdots \cup R_{M-1})\right] \le \frac{d}{M}.$$

The above corollary does not require the individual relation sets R_i to be independent internally. Each transcript may contain an arbitrarily correlated and adaptively generated collection of relations. The only requirement is that the transcripts themselves be independent samples from the same distribution.

Conditional Form. The same corollary holds conditionally on arbitrary fixed side information. More precisely, let W be a random variable such that, for every w in its support, conditioning on $W = w$ fixes the anchor tuple $\mathbf{x}$ and leaves $Tr_1, \ldots, \mathrm{Tr}_M$ independent and identically distributed. Then, for every such w,

$$\Pr\left[R_M \not\subseteq \mathrm{cl}_{\mathbf{x}}\big(R_1 \cup \cdots \cup R_{M-1}\big) \,\big|\, W = w\right] \le \frac{d}{M}.$$

This follows by applying Corollary 2 to the conditional distribution given $W = w$.

A Random-prefix Closure Lemma. The next lemma does not assume that the sampled relation sets are independent or identically distributed.

Lemma 3 (Random-Prefix Closure Lemma). *Let $\mathcal{M} = (\mathcal{E}, \mathcal{I})$ be a finite matroid of rank at most r. Let $R_1, \ldots, R_k$ be arbitrarily jointly distributed random variables taking values in $2^{\mathcal{E}}$, where $k > r$. Let $Y \leftarrow [k]$ be uniform and independent of $(R_1, \ldots, R_k)$. For every $i \in \{1, \ldots, k\}$ and every prefix $(U_1, \ldots, U_{i-1})$ in the support of $(R_1, \ldots, R_{i-1})$, define*

$$p_i(U_1, \ldots, U_{i-1}) = \Pr\Big[R_i \subseteq \mathrm{cl}_{\mathcal{M}}(U_1 \cup \cdots \cup U_{i-1}) \;\Big|\; R_1 = U_1, \ldots, R_{i-1} = U_{i-1}\Big].$$

For $i = 1$, the right-hand side is interpreted relative to $\mathrm{cl}_{\mathcal{M}}(\emptyset)$. Then

$$\Pr\left[p_Y(R_1, \ldots, R_{Y-1}) \ge \frac{k-r}{2k}\right] \ge \frac{k-r}{2k}.$$

Proof. For a fixed realization $(U_1, \ldots, U_k)$ of $(R_1, \ldots, R_k)$, define $V_i = U_1 \cup \cdots \cup U_i$ and $V_0 = \emptyset$. For at most r indices $i \in \{1, \ldots, k\}$, it can hold that $U_i \not\subseteq \mathrm{cl}_{\mathcal{M}}(V_{i-1})$. Indeed, every such index strictly increases the matroid rank, while the total rank is at most r. Therefore, for at least $k - r$ indices, $U_i \subseteq \mathrm{cl}_{\mathcal{M}}(V_{i-1})$. Since Y is uniform and independent of the relation sets, it follows that

$$\Pr\left[R_Y \subseteq \mathrm{cl}_{\mathcal{M}}(R_1 \cup \cdots \cup R_{Y-1}) \,|\, R_1, \ldots, R_k\right] \ge \frac{k-r}{k}.$$

Averaging over $(R_1, \ldots, R_k)$ yields

$$\Pr\left[R_Y \subseteq \mathrm{cl}_{\mathcal{M}}(R_1 \cup \cdots \cup R_{Y-1})\right] \ge \frac{k-r}{k}. \tag{3}$$

Let $P = p_Y(R_1, \ldots, R_{Y-1})$. Since Y is uniform and independent of $(R_1, \ldots, R_k)$, the law of total expectation gives

$$\begin{aligned}\mathbb{E}[P] &= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}\big[p_i(R_1, \ldots, R_{i-1})\big] \\ &= \frac{1}{k}\sum_{i=1}^{k}\Pr\left[R_i \subseteq \mathrm{cl}_{\mathcal{M}}(R_1 \cup \cdots \cup R_{i-1})\right] \\ &= \Pr\left[R_Y \subseteq \mathrm{cl}_{\mathcal{M}}(R_1 \cup \cdots \cup R_{Y-1})\right] \\ &\geq \frac{k-r}{k},\end{aligned}$$

where the final inequality follows from Equation (3). Next, let $\beta = \Pr\left[P \geq \frac{k-r}{2k}\right]$. Since $0 \leq P \leq 1$, we have

$$\mathbb{E}[P] \leq \beta \cdot 1 + (1-\beta)\cdot\frac{k-r}{2k} = \frac{k-r}{2k} + \beta\left(1 - \frac{k-r}{2k}\right).$$

Combining this with $\mathbb{E}[P] \geq (k-r)/k$ gives $\beta \geq (k-r)/2k$. This completes the proof of Lemma 3.

Application to Gain-Graph Relations. Applying Lemma 3 to the relation matroid of an anchor tuple gives the following corollary.

Corollary 3 (Random-Prefix Gain-Graph Closure). *Fix an anchor tuple $\boldsymbol{x} = (x_0, \ldots, x_d)$, and let $R_1, \ldots, R_k \subseteq \mathcal{E}_{\boldsymbol{x}}$ be arbitrarily jointly distributed random relation sets, where $k > d$. Let $Y \leftarrow [k]$ be uniform and independent of the relation sets. Then, with probability at least $(k-d)/2k$ over $(Y, R_1, \ldots, R_{Y-1})$,*

$$\Pr\Big[R_Y \subseteq \mathrm{cl}_{\boldsymbol{x}}(R_1 \cup \cdots \cup R_{Y-1}) \;\Big|\; Y, R_1, \ldots, R_{Y-1}\Big] \geq \frac{k-d}{2k}.$$

Conditioning on External Information. Lemma 3 and its corollary remain valid after conditioning on arbitrary side information W, provided that, conditional on every w in its support, Y remains uniform on $[k]$ and independent of $(R_1, \ldots, R_k)$. No independence among the R_i is required. In later applications, the full sequence may be defined hypothetically even though the adversary observes only the prefix preceding Y; it is essential only that this joint sequence be defined independently of Y. Looking ahead, Corollary 2 will be used for repeated encryption-decryption executions of a fixed identity, while Corollary 3 will be used when selecting the challenge identity among $n_{\mathsf{pp}} + 1$ identities. Negative equality outcomes need no separate sampling argument: Section 5.3 shows that two-sided containment of positive relations suffices to preserve the complete equality pattern.

5.3 Two-Sided Transfer of Generic-Action Transcripts

Lemma 1 transfers a transcript when the complete equality pattern is preserved. For our attack it is more convenient to track only positive relations. The following two-sided criterion suffices: if every positive relation arising in either of two executions is implied by one common gain graph realized in both worlds, then neither execution can acquire an equality absent from the other.

Positive Relation Triples. Let $\mathbf{x} = (x_0, \ldots, x_d)$ be an anchor tuple and let Tr be a generic-action transcript on $\mathbf{x}$.

Definition 25 (Positive Relation Set). *The* positive relation set *of* Tr *is*

$$\mathsf{PosRel}(\mathsf{Tr}) = \left\{(i, j, h^{-1} \circ g) \;\middle|\; (i,g),(j,h) \in \mathcal{Q}_{\mathsf{Tr}}, \sigma(g \star x_i) = \sigma(h \star x_j)\right\}.$$

Thus $(i, j, a) \in \mathrm{PosRel}(\mathrm{Tr})$ represents the relation $x_j = a \star x_i$.

Equality saturation and Lemma 1 imply $\mathrm{PosRel}(\mathrm{Tr}) \subseteq \mathrm{cl}(\Gamma_{\mathrm{Tr}})$. The inclusion may be strict, since closure may contain relations implied by paths for which no corresponding pair of anchored terms occurs. We continue to use Rel(Tr) when only the underlying canonical relation-matroid edges are relevant; PosRel(Tr) additionally records orientation and gain.

Lemma 4 (Two-sided Transcript Transfer). *Let $\mathcal{B}$ be a deterministic type-safe generic-action algorithm with explicit input w. Consider two executions of $\mathcal{B}$ with respective anchor tuples $\boldsymbol{x} = (x_0, \ldots, x_d)$ and $\boldsymbol{y} = (y_0, \ldots, y_d)$. Let $\mathrm{Tr}_{\mathbf{x}} = \mathrm{Tr}_{\mathcal{B}}(\mathbf{x})$ and $\mathrm{Tr}_{\mathbf{y}} = \mathrm{Tr}_{\mathcal{B}}(\mathbf{y})$ denote the cached equality-saturated transcripts of the two executions, as defined in Section 3.4. Suppose there exists a balanced gain graph $\Gamma = (\{0, \ldots, d\}, E, \gamma)$ such that: (i) both $\boldsymbol{x}$ and $\boldsymbol{y}$ realize Γ; and (ii) every positive relation arising in either execution is implied by Γ:*

$$\mathsf{PosRel}(\mathsf{Tr}_{\boldsymbol{x}}) \cup \mathsf{PosRel}(\mathsf{Tr}_{\boldsymbol{y}}) \subseteq \mathrm{cl}(\Gamma). \tag{4}$$

Then the two executions are identical up to a consistent renaming of typed handles.

In particular, we prove that the two executions issue the same action queries, follow the same control flow, and produce identical explicit outputs.

Proof. Couple the executions inductively, pairing input occurrences with the same anchor index. We maintain identical explicit states, corresponding handle occurrences carrying the same anchored terms, and a bijection between encountered handles that preserves and reflects equality. Suppose corresponding occurrences carry (i, g) and (j, h). If they are equal in the $\mathbf{x}$-execution, then

$$(i, j, h^{-1} \circ g) \in \mathrm{PosRel}(\mathrm{Tr}_{\mathbf{x}}) \subseteq \mathrm{cl}(\Gamma).$$

Since $\mathbf{y}$ realizes Γ, the corresponding occurrences are equal in the $\mathbf{y}$-execution. The converse follows symmetrically from $\mathrm{PosRel}(\mathrm{Tr}_{\mathbf{y}}) \subseteq \mathrm{cl}(\Gamma)$. Hence every equality test has the same outcome in both executions. This reasoning is well founded even though Equation (4) refers to the complete transcripts: if the executions had a first point of divergence, every term occurring before that point would occur in both transcripts, and any positive equality causing the divergence in one execution would, by the preceding argument, also hold in the other.

Type safety and renaming invariance therefore force the same next instruction. For a corresponding action query by $a \in G$ on a term (i, g), both responses receive $(i, a \circ g)$. If one response coincides with a previously encountered handle, equality saturation places the corresponding positive relation in that execution's PosRel set, and the two-sided argument forces the same coincidence in the other execution. Thus either both responses are old corresponding handles or both are new, in which case the bijection extends to them. The invariant persists throughout the computation, proving the claim. This completes the proof of Lemma 4.

Remark 6 (Necessity of Two-sided Control). We remark here that the one-sided condition $\mathrm{PosRel}(\mathrm{Tr}_{\mathbf{x}}) \subseteq \mathrm{cl}(\Gamma)$ is insufficient: it prevents an equality present in the $\mathbf{x}$-execution from disappearing in the $\mathbf{y}$-execution, but does not exclude additional equalities in the latter. Such an equality could change the control flow. The two-sided condition rules this out.

A Relation-Set Formulation. Let $\mathbf{x}$ and $\mathbf{y}$ be anchor tuples on $[0, d]$, and let $R \subseteq E_{\mathbf{x}} \cap E_{\mathbf{y}}$ be a canonical relation set realized by both tuples. Write Γ_R for its gain graph. Since both tuples realize R, their gain-labeled closures agree:

$$\mathrm{cl}_{\mathbf{x}}(R) = \mathrm{cl}_{\mathbf{y}}(R) = \mathrm{cl}(\Gamma_R).$$

Corollary 4 (Two-sided Transfer from a Learned Relation Set). *Let $\mathcal{B}$, $\mathbf{x}$, and $\mathbf{y}$ be as in Lemma 4. Let $\mathrm{Tr}_{\mathbf{x}} = \mathrm{Tr}_{\mathcal{B}}(\mathbf{x})$ and $\mathrm{Tr}_{\mathbf{y}} = \mathrm{Tr}_{\mathcal{B}}(\mathbf{y})$ be their cached equality-saturated transcripts. Let $R \subseteq \mathcal{E}_{\mathbf{x}} \cap \mathcal{E}_{\mathbf{y}}$ be a canonical relation set realized by both anchor tuples, and let Γ_R be its common gain graph. Suppose that $\mathrm{Rel}_{\mathbf{x}}(\mathrm{Tr}_{\mathbf{x}}) \subseteq \mathrm{cl}_{\mathbf{x}}(R)$ and $\mathrm{Rel}_{\mathbf{y}}(\mathrm{Tr}_{\mathbf{y}}) \subseteq \mathrm{cl}_{\mathbf{y}}(R)$. Then the two executions are identical up to a consistent renaming of typed handles. In particular, they produce identical explicit outputs.*

Proof. Consider any positive relation $(i, j, a) \in \mathrm{PosRel}(\mathsf{Tr}_{\mathbf{x}})$. If $i = j$, then the corresponding equality has the form $g \star x_i = h \star x_i$, where $a = h^{-1} \circ g$. Freeness of the action implies $g = h$ and hence $a = e$. Therefore, $(i, i, e) \in \mathrm{cl}(\Gamma_R)$ by the length-zero path convention.

Suppose now that $i \neq j$. After orienting its endpoints according to the canonical convention, the corresponding canonical relation belongs to $\mathrm{Rel}_{\mathbf{x}}(\mathsf{Tr}_{\mathbf{x}})$ and hence to $\mathrm{cl}_{\mathbf{x}}(R)$. Since $\mathrm{cl}_{\mathbf{x}}(R) = \mathrm{cl}(\Gamma_R)$ as gain-labeled relation sets, it follows that $(i, j, a) \in \mathrm{cl}(\Gamma_R)$. Consequently, $\mathrm{PosRel}(\mathsf{Tr}_{\mathbf{x}}) \subseteq \mathrm{cl}(\Gamma_R)$. The same argument gives $\mathrm{PosRel}(\mathsf{Tr}_{\mathbf{y}}) \subseteq \mathrm{cl}(\Gamma_R)$. The conclusion now follows from Lemma 4. This completes the proof of Corollary 4.

Application to Encryption-Decryption Computations. We now discuss the application of the two-sided transfer lemma (Lemma 4) to encryption-decryption computations in the restricted-key attack on TS-GGAM-IBE. Concretely, let $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a TS-GGAM-IBE scheme over the identity space ID as in Definition 5. Let $n = n(\mathsf{pp}) \leq n_{\mathsf{pp}}(\lambda)$. Let $\mathsf{pp} = \big(\mathsf{pp}_{\mathrm{exp}}, \widehat{P}_1, \ldots, \widehat{P}_n\big)$ be the public parameters of this TS-GGAM-IBE scheme, where $\mathsf{pp}_{\mathrm{exp}}$ is explicit and the remaining components are X-handles. In the restricted setting, the identity secret key contains no X-handles and can therefore be written as an explicit value $\mathsf{sk}_{\mathrm{exp}}$. Fixing an identity id, a message m, and the random tapes r_{Enc} and r_{Dec} of encryption and decryption turns the experiment that encrypts m under id and then decrypts the resulting ciphertext into a single deterministic type-safe generic-action computation. We denote this composed computation by

$$\mathcal{B}_{\mathsf{id},m,r_{\mathsf{enc}},r_{\mathsf{dec}}}(\mathsf{pp}_{\mathsf{exp}}, \mathsf{sk}_{\mathsf{exp}}, \widehat{x}_0, \widehat{P}_1, \ldots, \widehat{P}_n) = \mathsf{Dec}\big(\mathsf{pp}, \mathsf{sk}_{\mathsf{exp}}, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m; r_{\mathsf{enc}}); r_{\mathsf{dec}}\big),$$

where the randomness of encryption and decryption is fixed as part of the explicit description of the computation. We regard this composition as a single type-safe generic-action algorithm and use its cached equality-saturated transcript.

Since the identity secret key is explicit, the only persistent set-element anchors of this composed computation are the public origin and the n public-parameter handles. Every ciphertext handle generated by Enc has an anchored term relative to one of these public anchors. Consequently, the positive relation set of the complete encryption-decryption computation belongs to a relation matroid of rank at most $n \leq n_{\mathsf{pp}}(\lambda)$ (this follows from Corollary 1).

Transfer of Restricted-Key Decryption. Let $\mathsf{pp} = \big(\mathsf{pp}_{\mathrm{exp}}, \widehat{P}_1, \ldots, \widehat{P}_n\big)$ and $\mathsf{pp}' = \big(\mathsf{pp}_{\mathrm{exp}}, \widehat{P}'_1, \ldots, \widehat{P}'_n\big)$ be two IBE public parameters that have the same explicit component. Let $\mathbf{P} = \big(\widetilde{x}, P_1, \ldots, P_n\big)$ and $\mathbf{P}' = \big(\widetilde{x}, P'_1, \ldots, P'_n\big)$ be the corresponding public-anchor tuples. Let $R \subseteq \mathcal{E}_{\mathbf{P}} \cap \mathcal{E}_{\mathbf{P}'}$ be a canonical relation set realized by both tuples, and let Γ_R be its common gain graph. Fix an identity id, a message m, an explicit secret key $\mathsf{sk}_{\mathrm{exp}}$, and randomness $r_{\mathsf{Enc}}, r_{\mathsf{Dec}}$. Let $\mathrm{Tr}_{\mathbf{P}}$ and $\mathrm{Tr}_{\mathbf{P}'}$ be the cached equality-saturated transcripts of the respective composed encryption-decryption computations.

Corollary 5 (Transfer of Restricted-Key Decryption). *Suppose the following holds:* $\mathrm{PosRel}(\mathrm{Tr}_{\mathbf{P}}) \cup \mathrm{PosRel}(\mathrm{Tr}_{\mathbf{P}'}) \subseteq \mathrm{cl}(\Gamma_R)$. *Then the two composed computations produce the same explicit decryption output. In particular, the same conclusion holds if* $\mathrm{Rel}_{\mathbf{P}}(\mathrm{Tr}_{\mathbf{P}}) \subseteq \mathrm{cl}_{\mathbf{P}}(R)$ *and* $\mathrm{Rel}_{\mathbf{P}'}(\mathrm{Tr}_{\mathbf{P}'}) \subseteq \mathrm{cl}_{\mathbf{P}'}(R)$.

Proof. For the first formulation, apply Lemma 4 to the deterministic composed algorithm with anchor tuples $\mathbf{P}, \mathbf{P}'$. The second formulation follows from Corollary 4. This completes the proof of Corollary 5.

5.4 Offline Simulation and Alternative Setup Worlds

Our attack against restricted TS-GGAM-IBE searches offline for an alternative setup world consistent with the explicit public parameters and learned relative-action relations. Since a pseudoefficient adversary may perform unbounded local computation, it can enumerate finite random-tape supports and simulate fresh type-safe group-action worlds without querying the real action oracle. In this section, we formalize this observation and define the alternative worlds.

The Coordinate-based Offline Oracle. Fix a regular action $(G, X, \star, \widetilde{x})$. Every $x \in X$ has a unique origin-relative coordinate $a \in G$ such that $x = a \star \widetilde{x}$, and acting by $g \in G$ maps this coordinate to $g \circ a$. Let $\widetilde{\mathcal{H}}$ be an infinite formal handle space, disjoint from the real TS-GGAM handle space, with a fixed effective enumeration $\widetilde{\mathcal{H}} = \{\widetilde{h}_0, \widetilde{h}_1, \ldots\}$ where $\widetilde{h}_0 = \widehat{\widetilde{x}}_0$. We write $\mathrm{dom}(\mathsf{HT})$ for the set of coordinates currently represented and $\mathrm{im}(\mathsf{HT}) = \{\, \mathsf{HT}[a] : a \in \mathrm{dom}(\mathsf{HT}) \,\} \subseteq \widetilde{\mathcal{H}}$ for the set of formal handles issued so far.

Definition 26 (Coordinate-based Offline Action Oracle). *The offline oracle maintains a finite partial injection* $\mathsf{HT} : G \rightharpoonup \widetilde{\mathcal{H}}$*, where* $\mathsf{HT}[e] = \widetilde{\widehat{x}}_0$*. On a well-typed query* $(g, \mathsf{HT}[a])$*, it computes* $b = g \circ a$*, allocates* $\mathsf{HT}[b]$ *as the first unused handle in the fixed enumeration if* $b \notin \mathrm{dom}(\mathsf{HT})$*, and returns*

$$\widetilde{\mathcal{O}}_\star^{\mathsf{HT}}(g, \mathsf{HT}[a]) = \mathsf{HT}[g \circ a].$$

Inputs outside $G \times \mathrm{im}(\mathsf{HT})$ *are rejected. Equality is syntactic; hence, for encountered coordinates,* $\mathsf{HT}[a] = \mathsf{HT}[b]$ *if and only if* $a = b$*.*

The table HT and the coordinates in its domain are visible only to the outer offline simulator. An algorithm executed inside the simulated world receives only the formal handles, and may interact with them solely through the prescribed action-oracle and handle-equality interfaces.

Initialization on Simulated Input Handles. For an execution whose input elements have coordinates $a_0, \ldots, a_d$, the simulator first extends HT to $\{e, a_0, \ldots, a_d\}$ using the same deterministic allocation rule and supplies $\mathsf{HT}[a_i]$ as the corresponding input handle. Subsequent entries are allocated lazily.

Lemma 5 (Exact Offline Simulation). *Let* $\mathcal{B}$ *be a type-safe generic-action algorithm with fixed random tape and explicit input, and suppose its set-element inputs are* $x_i = a_i \star \widetilde{x}$ *for* $i \in [0, d]$*. Then its execution with the real type-safe oracle on* $\sigma(x_0), \ldots, \sigma(x_d)$ *is identical, up to a consistent handle renaming, to its execution with* $\widetilde{\mathcal{O}}_\star^{\mathsf{HT}}$ *on* $\mathsf{HT}[a_0], \ldots, \mathsf{HT}[a_d]$*. In particular, the executions have identical explicit outputs and equality patterns.*

Proof. Fix the random tape and explicit input of $\mathcal{B}$. During the execution, define a partial map from real action-set handles to offline handles by $\phi\big(\sigma(a \star \widetilde{x})\big) = \mathsf{HT}[a]$ for every coordinate a encountered so far. Regularity of the action and injectivity of σ and HT imply that ϕ is well defined and injective.

We prove by induction over the instructions executed by $\mathcal{B}$ that its real and offline machine states agree after applying ϕ to every action-set register. Explicit computations are identical in the two executions. For an action-oracle instruction, the real execution returns $\sigma\big(g \star (a \star \widetilde{x})\big) = \sigma\big((g \circ a) \star \widetilde{x}\big)$, whereas the offline execution returns $\mathsf{HT}[g \circ a]$; these values correspond under the extended map ϕ. Moreover, two real handles are equal exactly when their coordinates are equal, which is exactly when the corresponding offline handles are syntactically equal. Thus every explicit handle-equality test has the same outcome in the two executions. The two executions therefore have identical control flow and explicit outputs, and their action-set handles differ only by the injective renaming ϕ. Now, the renaming invariance of $\mathcal{B}$ completes the proof of Lemma 5.

Remark 7. Evaluations of $\widetilde{\mathcal{O}}_\star$ are local computations and make no calls to the real oracle $\mathcal{O}_\star^\sigma$; hence they do not contribute to the adversary's generic query complexity.

Enumerating Algorithmic Supports. By the finite-randomness and totality convention of Section 3.4, the complete random-tape supports of Setup, KeyGen, Enc, and Dec are finite and effectively enumerable. Together with Lemma 5, unbounded local computation can therefore enumerate setup and key-generation outputs, encryption-decryption transcripts, and exact probabilities of decidable events on such transcripts. Each candidate setup tape defines a fresh offline world with its own table $\mathsf{HT}^\star$; all subsequent simulations associated with that candidate reuse this table, which is extended lazily, whereas a new setup candidate starts with a fresh table.

Alternative Public Parameters. Let $n = n(\mathsf{pp}) \le n_{\mathsf{pp}}(\lambda)$. Let

$$\mathsf{pp} = \big(\mathsf{pp}_{\mathsf{exp}}, \widehat{P}_1, \ldots, \widehat{P}_n\big)$$

be the real public parameters of a TS-GGAM-IBE. Let $\mathbf{P} = (\widetilde{x}, P_1, \ldots, P_n)$ be the real public-anchor tuple, and let $E_{\mathbf{P}}$ be its canonical relation ground set from Definition 23. Throughout Sections 5.4 and 5.5, a *learned relation set* is $R \subseteq E_{\mathbf{P}}$. We write Γ_R for the corresponding balanced gain graph and $\mathrm{cl}_{\mathbf{P}}(R)$ for its closure in the relation matroid of $\mathbf{P}$. When another anchor tuple $\mathbf{P}^\star$ also realizes R, the gain-labeled closures agree, and we write

$$\mathrm{cl}(R) = \mathrm{cl}_{\mathbf{P}}(R) = \mathrm{cl}_{\mathbf{P}^\star}(R).$$

Consider a candidate setup execution generated using the offline oracle. Write its public parameters as $\mathsf{pp}^\star = \left(\mathsf{pp}^\star_{\mathsf{exp}}, \widehat{P}^\star_1, \ldots, \widehat{P}^\star_{n^\star}\right)$ [3]. Let $a^\star_0 = e$, and let $a^\star_i \in G$ be the origin-relative coordinate of $\widehat{P}^\star_i$, so that $P^\star_i = a^\star_i \star \widetilde{x}$.

Definition 27 (Consistency with Learned Relations). *The candidate public parameters* $\mathsf{pp}^\star$ *are* consistent with R *if: (i) their explicit component agrees with the real one:* $\mathsf{pp}^\star_{\mathsf{exp}} = \mathsf{pp}_{\mathsf{exp}}$*; and (ii) their anchor tuple realizes every relation in* R*, i.e., for every* $(i, j, \delta) \in R$*,* $a^\star_j = \delta \circ a^\star_i$ *(or equivalently,* $P^\star_j = \delta \star P^\star_i$*).*

Symbolic Descriptions of Offline Worlds. A simulated handle in $\mathcal{H}_\sim$ is not an element of the real TS-GGAM handle space and must never be submitted to the real action oracle. We therefore represent a candidate setup world by an explicit symbolic record $\mathsf{W}^\star = \left(\rho^\star_{\mathsf{Setup}}, \mathsf{HT}^\star, \widetilde{\mathsf{msk}}^\star, \widetilde{\mathsf{pp}}^\star\right)$. Here $\rho^\star_{\mathsf{Setup}}$ is the setup random tape, $\mathsf{HT}^\star$ is the coordinate-annotated offline table, and $\widetilde{\mathsf{msk}}^\star, \widetilde{\mathsf{pp}}^\star$ are the outputs of the corresponding offline execution of Setup. The record also retains the coordinates associated with every simulated action-set handle appearing in these outputs. When $\mathsf{W}^\star$ is fixed, we write $\mathsf{msk}^\star$ and $\mathsf{pp}^\star$ for its simulated setup outputs and write $\widetilde{\mathcal{O}}^{\mathsf{W}^\star}_\star$ for the coordinate-based oracle using its table. This notation is internal to the offline world: neither $\mathsf{pp}^\star$ nor $\mathsf{msk}^\star$ is asserted to be a well-typed object in the real TS-GGAM execution.

Candidate Identity Keys. Fix an offline world $\mathsf{W}^\star$ and an identity id. Running KeyGen inside this world on an enumerated random tape produces a candidate secret key $\mathsf{sk}^\star_{\mathsf{id}}$. By the restricted TS-GGAM-IBE syntax, secret keys contain no action-set handles. Hence $\mathsf{sk}^\star_{\mathsf{id}}$ is an explicit object that can later be supplied to an algorithm in the real execution. An *offline candidate record* for id is a pair $\mathsf{Cand}^\star_{\mathsf{id}} = \left(\mathsf{W}^\star, \mathsf{sk}^\star_{\mathsf{id}}\right)$.

Defining Candidate-World Probabilities. Let $P^\star$ be the candidate public-anchor tuple encoded by $W^\star$. Fix an identity id and an explicit candidate key $\mathsf{sk}^\star_{\mathsf{id}}$. For $b \in \{0,1\}$ and fixed encryption and decryption random tapes ρ_{Enc} and ρ_{Dec}, let $\mathrm{Tr}^\star_{\mathsf{id},b}(\rho_{\mathsf{Enc}}, \rho_{\mathsf{Dec}})$ denote the cached equality-saturated transcript of the composed candidate-world computation

$$\mathsf{Dec}^{\widetilde{\mathcal{O}}^{\star,W^\star}}\left(\mathsf{pp}^\star, \mathsf{sk}^\star_{\mathsf{id}}, \mathsf{Enc}^{\widetilde{\mathcal{O}}^{\star,W^\star}}(\mathsf{pp}^\star, \mathsf{id}, b; \rho_{\mathsf{Enc}}); \rho_{\mathsf{Dec}}\right).$$

For $b \in \{0,1\}$, define the candidate-world correctness probability by

$$p^\star_{\mathsf{cor}}(b) = \Pr_{\rho_{\mathsf{Enc}}, \rho_{\mathsf{Dec}}}\left[\mathsf{Dec}^{\widetilde{\mathcal{O}}^{\star,W^\star}}\left(\mathsf{pp}^\star, \mathsf{sk}^\star_{\mathsf{id}}, \mathsf{Enc}^{\widetilde{\mathcal{O}}^{\star,W^\star}}(\mathsf{pp}^\star, \mathsf{id}, b; \rho_{\mathsf{Enc}}); \rho_{\mathsf{Dec}}\right) = b\right].$$

For a learned relation set R, define the candidate-world relation-containment probability

$$p^\star_{\mathsf{rel}}(b; R) = \Pr_{\rho_{\mathsf{Enc}}, \rho_{\mathsf{Dec}}}\left[\mathrm{Rel}_{P^\star}\left(\mathrm{Tr}^\star_{\mathsf{id},b}(\rho_{\mathsf{Enc}}, \rho_{\mathsf{Dec}})\right) \subseteq \mathrm{cl}_{P^\star}(R)\right].$$

Definition 28 (Admissible Offline Candidate). *Let* $\tau_{\mathrm{cor}}, \tau_{\mathrm{rel}} \in \mathbb{Q} \cap [0,1]$*. An offline candidate record* $\mathsf{Cand}^\star_{\mathsf{id}} = (\mathsf{W}^\star, \mathsf{sk}^\star_{\mathsf{id}})$ *is* $(\tau_{\mathrm{cor}}, \tau_{\mathrm{rel}})$*-admissible relative to* $\mathcal{R}$ *if the following conditions hold:*

1. $\mathsf{W}^\star$ *is induced by a random tape in the support of* $\mathsf{Setup}(1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_{0,\sim})$.
2. $\mathsf{sk}^\star_{\mathsf{id}}$ *is induced by a random tape in the support of* $\mathsf{KeyGen}^{\widetilde{\mathcal{O}}^{\mathsf{W}^\star}_\star}(\mathsf{pp}^\star, \mathsf{msk}^\star, \mathsf{id})$.
3. $\mathsf{pp}^\star$ *is consistent with* $\mathcal{R}$ *in the sense of Definition 27.*
4. *For every* $b \in \{0,1\}$*,* $p^\star_{\mathrm{cor}}(b) \geq \tau_{\mathrm{cor}}$.
5. *For every* $b \in \{0,1\}$*,* $p^\star_{\mathrm{rel}}(b; R) \geq \tau_{\mathrm{rel}}$.

Lemma 6 (Alternative-Candidate Search). *For rational thresholds* $\tau_{\mathrm{cor}}, \tau_{\mathrm{rel}} \in \mathbb{Q} \cap [0,1]$*, there exists an algorithm*

$$\mathsf{FindCandidate}\left(1^\lambda, \mathsf{gpar}_\lambda, \mathsf{pp}_{\mathrm{exp}}, \mathsf{id}, \mathcal{R}, \tau_{\mathrm{cor}}, \tau_{\mathrm{rel}}\right),$$

that makes no calls to the real group-action oracle and, using only unbounded local computation and coordinate-based offline simulation, outputs an admissible record $\mathsf{Cand}^\star_{\mathsf{id}} = (\mathsf{W}^\star, \mathsf{sk}^\star_{\mathsf{id}})$ *whenever one exists; otherwise, it outputs* $\perp$.

[3] Note that consistency with the real public parameters implies $n^\star = n$, since equality of the explicit canonical shapes fixes the handle-occurrence structure.

Proof. Fix canonical orderings of the finite random-tape spaces of all four scheme algorithms and use the deterministic handle-allocation rule of Definition 26. Enumerate every setup tape, starting a fresh offline table for each. Discard a world unless its explicit public parameters equal $\mathsf{pp}_{\mathrm{exp}}$ and it is consistent with R. For each surviving world, enumerate all KeyGen tapes for id. Each resulting explicit key defines a candidate record. For every record, enumerate all encryption and decryption tapes for both $b \in \{0,1\}$, reusing the candidate world's table, and thereby compute $p^{\star}_{\mathsf{cor}}(b)$ and $p^{\star}_{\mathsf{rel}}(b;R)$ exactly. Return the first record satisfying Definition 28. All supports are finite, the probabilities are rational and exactly comparable with the thresholds, and every oracle evaluation is coordinate-based local computation. Finally, we invoke exhaustiveness to complete the proof of Lemma 6.

Remark 8 (Offline Records versus Real Typed Objects). The formal handles inside $W^{\star}$ are never submitted to the real action oracle. They serve only to describe and continue the offline simulation. In contrast, $\mathsf{sk}^{\star}_{\mathsf{id}}$ is entirely explicit and can be extracted from $\mathsf{Cand}^{\star}_{\mathsf{id}}$ and used in the real execution. This is precisely where the restricted secret-key syntax is needed.

Lemma 6 reduces the remaining task to showing that an admissible candidate exists with noticeable probability for the selected challenge identity. Section 5.5 does so using correctness and random-prefix closure, and then transfers the candidate-world behavior to the real public-parameter world using Corollary 5.

5.5 Impossibility of Restricted TS-GGAM-IBE

We now prove a tight impossibility result for restricted TS-GGAM-IBE schemes. Our proof follows a random-prefix strategy. If the public parameters contain n_{pp} non-origin set-element handles, then their relation matroid, including the public origin, has rank at most n_{pp}. Among any $(n_{\mathsf{pp}}+1)$ identities, a uniformly selected identity is therefore conditionally likely to expose no new relation beyond those exposed by the preceding identities. The adversary in our attack uses the preceding identities to construct an alternative key for the selected identity and then applies the two-sided transfer lemma to show that this key is useful under the real public parameters.

Theorem 1 (Impossibility of Restricted TS-GGAM-IBE). *Let $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a restricted TS-GGAM-IBE scheme over an efficiently indexable identity-space family $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda\in\mathbb{N}}$. Suppose that its public parameters contain at most $n = n_{\mathsf{pp}}(\lambda)$ distinct non-origin $\mathcal{X}$-handles, where n_{pp} is publicly known, efficiently computable, and polynomially bounded, and let $\epsilon(\lambda)$ denote its pointwise correctness error. Let*

$$\Lambda^{\mathsf{res}}_{\Pi} = \left\{ \lambda \in \mathbb{N} : |\mathsf{ID}_\lambda| \geq n_{\mathsf{pp}}(\lambda) + 1 \text{ and } \epsilon(\lambda) \leq \frac{1}{640\big(n_{\mathsf{pp}}(\lambda)+1\big)} \right\}.$$

Then there exists a single pseudoefficient type-safe generic-action adversary $\mathcal{A}$ such that, for every $\lambda \in \Lambda^{\mathsf{res}}_{\Pi}$,

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) \geq \frac{1}{40\big(n_{\mathsf{pp}}(\lambda)+1\big)} - e^{-\Omega(\lambda)}.$$

Consequently, if $\Lambda^{\mathsf{res}}_{\Pi}$ is infinite, then Π is not fully selective-secure. In particular, a fully selective-secure restricted scheme with negligible pointwise correctness error satisfies $|\mathsf{ID}_\lambda| \leq n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ.

The Adversary against Restricted TS-GGAM-IBE. Let $k = (n+1)$, $L = 8(n+1)$, $T = \lambda$, $\tau_{\mathsf{cor}} = 19/20$, and $\tau_{\mathsf{rel}} = 4/5$. We define the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that interacts with the challenger in the fully selective restricted TS-GGAM-IBE security experiment as follows.

1. $\mathcal{A}_1$ computes $\mathbf{S} \leftarrow \mathsf{IndexID}(1^\lambda, 1^{n+1})$. If $\mathbf{S} = \bot$, it enters a fallback mode. It obtains $(\mathsf{id}_0) \leftarrow \mathsf{IndexID}(1^\lambda, 1^{k'})$ for $k' = 1$, declares id_0 as its challenge identity with no extraction identities, and outputs $\mathsf{state} = (\mathsf{fallback})$. On the other hand, if $\mathbf{S} = (\mathsf{id}_1, \ldots, \mathsf{id}_{n+1})$, it samples $Y \leftarrow [n+1]$, declares id_Y as its challenge identity and $\mathsf{id}_1, \ldots, \mathsf{id}_{Y-1}$ as its extraction identities, and outputs

$$\mathsf{state} = \big(\mathsf{attack}, 1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_0, Y, \mathsf{id}_1, \ldots, \mathsf{id}_{n+1}\big).$$

2. The algorithm $\mathcal{A}_2$ proceeds as follows:
   (a) $\mathcal{A}_2$ receives $\big(\mathsf{state}, \mathsf{pp}, \mathsf{sk}_{\mathsf{id}_1}, \ldots, \mathsf{sk}_{\mathsf{id}_{Y-1}}, C^\star\big)$. If $\mathsf{state} = (\mathsf{fallback})$, then $\mathcal{A}_2$ outputs an independent uniformly random bit and halts.

   (b) Otherwise, $\mathcal{A}_2$ infers state as an attack-mode state and parses

$$\mathsf{state} = \big(\mathsf{attack}, 1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_0, Y, \mathsf{id}_1, \ldots, \mathsf{id}_{n+1}\big),$$

$$\mathsf{pp} = (\mathsf{pp}_{\mathrm{exp}}, \widehat{P}_1, \ldots, \widehat{P}_d), \qquad d \le n,$$

   where $\widehat{P}_1, \ldots, \widehat{P}_d$ are the distinct non-origin $\mathcal{X}$-handles occurring in pp. It sets

$$\widehat{\mathbf{P}} := (\widehat{x}_0, \widehat{P}_1, \ldots, \widehat{P}_d)$$

   and treats this tuple as the indexed list of public anchors. From every instrumented transcript it extracts gain-labeled relations using only anchor indices, provenance annotations, and positive handle-equality outcomes.

   (c) Let $R_0 = \varnothing$. $\mathcal{A}_2$ constructs $R_1 \ldots, R_{Y-1}$ recursively as follows: for every $i < Y$, it performs the following steps:

      i. $\mathcal{A}_2$ runs $\mathsf{Cand}_i^\star \leftarrow \mathsf{FindCandidate}\big(1^\lambda, \mathsf{gpar}_\lambda, \mathsf{pp}_{\mathrm{exp}}, \mathsf{id}_i, R_{i-1}, \tau_{\mathsf{cor}}, \tau_{\mathsf{rel}}\big)$. If $\mathsf{Cand}_i^\star \neq \perp$, write $\mathsf{Cand}_i^\star = (\mathsf{W}_i^\star, \mathsf{sk}_i^\star)$, where $\mathsf{sk}_i^\star$ is the explicit candidate key contained in the record. Otherwise, $\mathsf{sk}_i^\star$ is undefined.

      ii. Next, for each $v \in [L]$ and $b \in \{0,1\}$, $\mathcal{A}_2$ uses fresh independent randomness to execute the genuine-key computation $\mathsf{Dec}\big(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_i, b)\big)$ under the real action oracle. Let $\mathsf{Tr}_{i,b,v}^{\mathrm{gen}}$ be the corresponding cached equality-saturated transcript. $\mathcal{A}_2$ sets

$$F_{i,v}^{\mathrm{gen}} = \bigcup_{b \in \{0,1\}} \mathrm{Rel}_{\mathbf{P}}\big(\mathsf{Tr}_{i,b,v}^{\mathrm{gen}}\big).$$

      iii. If $\mathsf{sk}_i^\star$ is defined, $\mathcal{A}_2$ also uses fresh independent randomness to execute $\mathsf{Dec}\big(\mathsf{pp}, \mathsf{sk}_i^\star, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_i, b)\big)$ for each $b \in \{0,1\}$. Let $\mathsf{Tr}_{i,b,v}^{\mathrm{cand}}$ denote the corresponding transcript. $\mathcal{A}_2$ sets

$$F_{i,v}^{\mathrm{cand}} = \bigcup_{b \in \{0,1\}} \mathrm{Rel}_{\mathbf{P}}\big(\mathsf{Tr}_{i,b,v}^{\mathrm{cand}}\big).$$

         On the other hand, if $\mathsf{sk}_i^\star$ is undefined, it sets $F_{i,v}^{\mathrm{cand}} = \varnothing$.

      iv. Finally, $\mathcal{A}_2$ sets

$$F_{i,v} = F_{i,v}^{\mathrm{gen}} \cup F_{i,v}^{\mathrm{cand}}, \qquad E_i = \bigcup_{v=1}^{L} F_{i,v}, \qquad R_i = R_{i-1} \cup E_i.$$

         Note that every set $F_{i,v}$, E_i, and R_i is a subset of the canonical relation ground set $E_{\mathbf{P}}$. Hence the entire sequence belongs to the relation matroid of $\mathbf{P}$, whose rank is at most $d \le n$.

   (d) $\mathcal{A}_2$ then runs $\mathsf{Cand}_Y^\star \leftarrow \mathsf{FindCandidate}\big(1^\lambda, \mathsf{gpar}_\lambda, \mathsf{pp}_{\mathrm{exp}}, \mathsf{id}_Y, R_{Y-1}, \tau_{\mathsf{cor}}, \tau_{\mathsf{rel}}\big)$. If $\mathsf{Cand}_Y^\star = \perp$, $\mathcal{A}_2$ outputs a uniformly random bit. Otherwise, write $\mathsf{Cand}_Y^\star = (\mathsf{W}_Y^\star, \mathsf{sk}_Y^\star)$, and proceed to the next validation phase using only the explicit key $\mathsf{sk}_Y^\star$.

   (e) $\mathcal{A}_2$ validates $\mathsf{sk}_Y^\star$ as follows. For each $t \in [T]$, it independently samples $b_t \leftarrow \{0,1\}$, generates $C_t \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_Y, b_t)$, and computes

$$b_t' \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_Y^\star, C_t).$$

   Let $N_{\mathrm{val}} = |\{t \in [T] : b_t' = b_t\}|$. If $N_{\mathrm{val}} < (21/40)T$, $\mathcal{A}_2$ outputs a uniformly random bit. Otherwise, it computes

$$d^\star \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_Y^\star, C^\star).$$

   If $d^\star \in \{0,1\}$, $\mathcal{A}_2$ outputs $d^\star$. Otherwise, if $d^\star = \perp$, it outputs a uniformly random bit.

Remark 9. Condition on a setup output $(\mathsf{msk}, \mathsf{pp})$ and let $\mathbf{P} = (\widetilde{x}, P_1, \ldots, P_d)$ for $d \le n$ be its semantic anchor tuple. Although $\mathcal{A}_2$ does not know these semantic elements or their coordinates, every relation inserted into R_i is computable from its instrumented transcript.

For the remainder of the analysis, fix an arbitrary $\lambda \in \Lambda_{\Pi}^{\mathsf{res}}$. At this parameter, $\mathsf{IndexID}(1^{\lambda}, 1^{n+1})$ returns an ordered tuple of $(n+1)$ pairwise distinct identities, and $\epsilon(\lambda) \le 1/(640(n+1))$. Hence the adversary follows its attack branch.

The Good-Prefix Event. For every $i \in [n+1]$ and every prefix in the support of $(R_1, \ldots, R_{i-1})$, let

$$q_i(R_1, \ldots, R_{i-1}) = \Pr[E_i \subseteq \mathrm{cl}_{\mathbf{P}}(R_{i-1}) \,|\, R_1, \ldots, R_{i-1}] \,.$$

Let $\mathsf{GoodPrefix}$ be an event defined as

$$\mathsf{GoodPrefix} = \left\{ q_Y(R_1, \ldots, R_{Y-1}) \ge \frac{1}{2(n+1)} \right\}.$$

Claim 2 *For every fixed setup output* $(\mathsf{msk}, \mathsf{pp})$*,*

$$\Pr[\mathsf{GoodPrefix} \mid \mathsf{msk}, \mathsf{pp}] \ge \frac{1}{2(n+1)}.$$

Proof. Condition on $(\mathsf{msk}, \mathsf{pp})$ and define hypothetically the complete sequence $(E_i, R_i)_{i \in [n+1]}$ by running the prefix-generation procedure for every identity, independently of Y. Hence Y remains uniform and independent of $(R_1, \ldots, R_{n+1})$. Let $k = (n+1)$. The relation matroid of $\mathbf{P}$ has rank $d \le n = k-1$, so applying Corollary 3 gives

$$\Pr\left[\Pr[R_Y \subseteq \mathrm{cl}_{\mathbf{P}}(R_1 \cup \cdots \cup R_{Y-1}) \,|\, Y, R_1, \ldots, R_{Y-1}] \ge \frac{k-d}{2k} \,\middle|\, \mathsf{msk}, \mathsf{pp} \right] \ge \frac{k-d}{2k}.$$

Since $d \le n = k-1$, we have

$$\frac{k-d}{2k} \ge \frac{1}{2k} = \frac{1}{2(n+1)}.$$

Since the R_i are cumulative, $R_1 \cup \cdots \cup R_{Y-1} = R_{Y-1}$ and $R_Y = R_{Y-1} \cup E_Y$. Thus the displayed conditional probability is exactly q_Y, which completes the proof of Claim 2.

Existence of an Admissible Candidate. We next show that every good prefix admits a candidate for identity id_Y.

Claim 3 *Fix a setup output* $(\mathsf{msk}, \mathsf{pp})$ *and a realization* $Y, R_1, \ldots, R_{Y-1}$ *for which* $\mathsf{GoodPrefix}$ *occurs. Then there exists an offline candidate record for identity* id_Y *that is* $(\tau_{\mathsf{cor}}, \tau_{\mathsf{rel}})$*-admissible relative to* R_{Y-1}*. Consequently,* $\mathsf{Cand}_Y^{\star} \neq \bot$*.*

Proof. Set $q_0 = 1/(2(n+1))$. The definition of $\mathsf{GoodPrefix}$ gives

$$\Pr[E_Y \subseteq \mathrm{cl}_{\mathbf{P}}(R_{Y-1}) \,|\, Y, R_1, \ldots, R_{Y-1}] \ge q_0. \tag{5}$$

Let $Z \leftarrow \mathsf{KeyGen}^{\mathcal{O}_{\star}^{\sigma}}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_Y)$ be the genuine random key for identity id_Y. For every key value z in the support of Z, let

$$\rho(z) = \Pr[E_Y \subseteq \mathrm{cl}_{\mathbf{P}}(R_{Y-1}) \,|\, Y, R_1, \ldots, R_{Y-1}, Z = z] \,, \qquad S_{\mathrm{rel}} = \left\{ z : \rho(z) \ge \frac{q_0}{2} \right\}.$$

Equation (5) implies

$$q_0 \le \Pr[Z \in S_{\mathrm{rel}}] \cdot 1 + \big(1 - \Pr[Z \in S_{\mathrm{rel}}]\big) \frac{q_0}{2},$$

and hence

$$\Pr[Z \in S_{\mathrm{rel}}] \ge \frac{q_0}{2 - q_0} \ge \frac{q_0}{2} = \frac{1}{4(n+1)}. \tag{6}$$

For $b \in \{0,1\}$ and every fixed key value z, let

$$\mathrm{err}_b(z) = \Pr\big[\mathsf{Dec}\big(\mathsf{pp}, z, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_Y, b)\big) \neq b\big] .$$

Pointwise correctness gives $\mathbb{E}_Z[\mathrm{err}_b(Z)] \leq \epsilon(\lambda)$ for each b. Next, Markov's inequality and a union bound therefore give

$$\Pr\left[\exists b \in \{0,1\} : \mathrm{err}_b(Z) > \frac{1}{20}\right] \leq 40\epsilon(\lambda) \leq \frac{1}{16(n+1)}. \tag{7}$$

Combining Equations (6) and (7), there exists a value $z \in S_{\mathrm{rel}}$ such that

$$\mathrm{err}_b(z) \leq \frac{1}{20} \qquad \text{for every } b \in \{0,1\}. \tag{8}$$

Conditioned on the fixed prefix and on $Z = z$, $F_{Y,1}, \ldots, F_{Y,L}$ are independent and identically distributed. Since $E_Y = \bigcup_{v=1}^{L} F_{Y,v}$, we obtain

$$\Pr[F_{Y,1} \subseteq \mathrm{cl}_{\mathbf{P}}(R_{Y-1}) \,|\, Y, R_1, \ldots, R_{Y-1}, Z = z] = \rho(z)^{1/L} \geq \left(\frac{1}{4(n+1)}\right)^{1/(8(n+1))} \geq \frac{4}{5}. \tag{9}$$

Because $F_{Y,1}$ contains the canonical relations arising from the genuine-key computation for each message, Equation (9) implies that, for every $b \in \{0,1\}$,

$$\Pr\Big[\mathrm{Rel}_{\mathbf{P}}\big(\mathsf{Tr}^{\mathsf{gen}}_{Y,b}(\mathsf{pp}, z)\big) \subseteq \mathrm{cl}_{\mathbf{P}}(R_{Y-1})\Big] \geq \frac{4}{5}. \tag{10}$$

Choose a setup random tape that produces the fixed real setup output $(\mathsf{msk}, \mathsf{pp})$ and a key-generation random tape that produces z. Replay these tapes using the coordinate-based offline oracle. By Lemma 5, the resulting symbolic world $\mathsf{W}^{\mathrm{real}}$ is isomorphic to the real setup world and produces the same explicit key z. Consider the offline candidate record $\mathsf{Cand}^{\mathrm{real}}_Y = (\mathsf{W}^{\mathrm{real}}, z)$. Its simulated public parameters have explicit component $\mathsf{pp}_{\mathrm{exp}}$ and realize every relation in R_{Y-1}. Equation (8) gives candidate-world correctness at least 19/20 for both messages, while Equation (10) and the exact offline simulation give candidate-world relation containment at least 4/5 for both messages. Thus, $\mathsf{Cand}^{\mathrm{real}}_Y$ is admissible.

Completeness of Lemma 6 now implies that $\mathsf{FindCandidate}$ returns an admissible record. Notice that this argument is not circular: if the original search returned $\perp$, then E_Y contained only the genuine-key relations, and the preceding argument constructs an admissible record, contradicting the completeness of Lemma 6. This completes the proof of Claim 3.

Usefulness of the Alternative Key. Fix a good prefix and write the returned candidate record as $\mathsf{Cand}^\star_Y = (\mathsf{W}^\star_Y, \mathsf{sk}^\star_Y)$. Let $\mathsf{pp}^\star_Y$ denote the simulated public parameters encoded by $\mathsf{W}^\star_Y$, and let $\mathbf{P}^\star_Y$ be their public-anchor tuple.

Claim 4 *Fix a setup output* $(\mathsf{msk}, \mathsf{pp})$ *and a realization* $(Y, R_1, \ldots, R_{Y-1})$ *for which* $\mathsf{GoodPrefix}$ *occurs. Let* $(W^\star_Y, \mathsf{sk}^\star_Y)$ *be the candidate returned by* $\mathsf{FindCandidate}$ *on this fixed prefix. Then, for every* $b \in \{0,1\}$,

$$\Pr_{\rho_{\mathsf{Enc}}, \rho_{\mathsf{Dec}}}\Big[\mathrm{Rel}_P\big(\mathrm{Tr}^{\mathsf{cand}}_{Y,b}(\mathsf{pp}, \mathsf{sk}^\star_Y; \rho_{\mathsf{Enc}}, \rho_{\mathsf{Dec}})\big) \subseteq \mathrm{cl}_P(R_{Y-1})\Big] \geq \frac{4}{5}.$$

Proof. The fixed prefix determines $\mathsf{sk}^\star_Y$ because $\mathsf{FindCandidate}$ is deterministic. Let p_b denote the probability in the claim. If $E_Y \subseteq \mathrm{cl}_{\mathbf{P}}(R_{Y-1})$, then all L independently sampled candidate-key transcripts for message b satisfy containment. Hence

$$\frac{1}{2(n+1)} \leq q_Y(R_1, \ldots, R_{Y-1}) \leq p_b^L.$$

Plugging in $L = 8(n+1)$, we obtain

$$p_b \geq \left(\frac{1}{2(n+1)}\right)^{1/(8(n+1))} \geq \frac{4}{5}.$$

This completes the proof of Claim 4.

Claim 5 *Under the fixed setup output and good prefix of Claim 4, for every $b \in \{0,1\}$,*

$$\Pr\left[\mathsf{Dec}\big(\mathsf{pp}, \mathsf{sk}^\star_Y, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_Y, b)\big) = b\right] \geq \frac{11}{20},$$

where the probability is over fresh encryption and decryption randomness.

Proof. Couple the real- and candidate-public-parameter executions using the same encryption and decryption tapes. Admissibility gives, in the candidate world, relation containment with probability at least $4/5$ and correct decryption with probability at least $19/20$. By Claim 4, real-world relation containment also has probability at least $4/5$. Thus both containment events occur with probability at least

$$\frac{4}{5} + \frac{4}{5} - 1 = \frac{3}{5}.$$

Both anchor tuples realize R_{Y-1} and have the same explicit public component, so Corollary 5 makes the two decryption outputs identical whenever both containment events occur. Consequently,

$$\Pr[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}^\star_Y, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_Y, b)) \neq b] \leq \left(1 - \frac{3}{5}\right) + \left(1 - \frac{19}{20}\right) = \frac{9}{20}.$$

This completes the proof of Claim 5.

Validation and Final Advantage. For a fixed explicit candidate key sk, define its average real-world success probability by

$$p(\mathsf{sk}) = \Pr_{b, r_{\mathsf{Enc}}, r_{\mathsf{Dec}}}\left[\mathsf{Dec}\big(\mathsf{pp}, \mathsf{sk}, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_Y, b; r_{\mathsf{Enc}}); r_{\mathsf{Dec}}\big) = b\right],$$

where b is uniform. By Claim 5, on every fixed setup output and every fixed good prefix, $p(\mathsf{sk}^\star_Y) \geq 11/20$. Since validation uses $T = \lambda$ fresh independent trials, a Chernoff bound gives

$$\Pr\left[N_{\mathrm{val}} < \frac{21}{40}T \,\middle|\, \mathsf{GoodPrefix}\right] \leq e^{-\Omega(T)} = e^{-\Omega(\lambda)}.$$

It follows that, conditioned on $\mathsf{GoodPrefix}$, the adversary wins with probability at least $11/20 - e^{-\Omega(\lambda)}$. It remains to analyze a prefix outside $\mathsf{GoodPrefix}$. Fix the setup output and a realization of the prefix for which $\neg\mathsf{GoodPrefix}$ occurs. Since $\mathsf{FindCandidate}$ is deterministic on its explicit input, its output is fixed under this conditioning. If it outputs $\perp$, the adversary wins with probability exactly $1/2$. If it outputs $(W^\star_Y, \mathsf{sk}^\star_Y) \neq \perp$ with $p(\mathsf{sk}^\star_Y) < 1/2$, then a Chernoff bound gives

$$\Pr\left[N_{\mathrm{val}} \geq \frac{21}{40}T\right] \leq e^{-\Omega(T)}.$$

So the adversary wins with probability at least $\frac{1}{2} - e^{-\Omega(T)}$. Finally, suppose that $p(\mathsf{sk}^\star_Y) \geq 1/2$, and let $\alpha = \Pr\left[N_{\mathrm{val}} \geq (21/40)T\right]$ be the probability that the candidate key passes validation. Independence of validation from the challenge implies that the adversary's conditional winning probability is at least

$$(1-\alpha)\frac{1}{2} + \alpha p(\mathsf{sk}^\star_Y) \geq \frac{1}{2}.$$

Therefore, in every case, including when $\mathsf{FindCandidate}$ outputs $\perp$, we have

$$\Pr\big[\mathcal{A} \text{ wins} \;\big|\; \neg\mathsf{GoodPrefix}\big] \geq \frac{1}{2} - e^{-\Omega(T)} = \frac{1}{2} - e^{-\Omega(\lambda)}.$$

By Claim 2, for every fixed setup output,

$$\begin{aligned}
\Pr[\mathcal{A} \text{ wins} \mid \mathsf{msk}, \mathsf{pp}] &\geq \Pr[\mathsf{GoodPrefix} \mid \mathsf{msk}, \mathsf{pp}]\left(\frac{11}{20} - e^{-\Omega(\lambda)}\right) \\
&\quad + \big(1 - \Pr[\mathsf{GoodPrefix} \mid \mathsf{msk}, \mathsf{pp}]\big)\left(\frac{1}{2} - e^{-\Omega(\lambda)}\right) \\
&\geq \frac{1}{2} + \frac{1}{20} \cdot \frac{1}{2(n+1)} - e^{-\Omega(\lambda)} \\
&= \frac{1}{2} + \frac{1}{40(n+1)} - e^{-\Omega(\lambda)}.
\end{aligned}$$

The bound holds uniformly for every setup output in the support. Averaging over the setup randomness therefore gives

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) \geq \frac{1}{40(n+1)} - e^{-\Omega(\lambda)}.$$

This establishes the claimed advantage bound for the fixed parameter $\lambda \in \Lambda^{\mathsf{res}}_{\Pi}$.

Query Complexity. There are at most n prefix-learning stages, each using $O(L) = O(n)$ real encryption-decryption invocations, followed by $T = \lambda$ validation trials. All calls to $\mathsf{FindCandidate}$ are entirely offline. Since the scheme algorithms make polynomially many action queries and $n = n_{\mathsf{pp}}(\lambda)$ is polynomially bounded, $\mathcal{A}$ is pseudoefficient. Since the construction of $\mathcal{A}$ is uniform in λ, this single adversary achieves the stated bound for every $\lambda \in \Lambda^{\mathsf{res}}_{\Pi}$. If $\Lambda^{\mathsf{res}}_{\Pi}$ is infinite, its advantage is inverse-polynomial infinitely often and therefore non-negligible. Finally, if ϵ is negligible and $|\mathsf{ID}_\lambda| \geq n_{\mathsf{pp}}(\lambda) + 1$ infinitely often, then polynomial boundedness of n_{pp} implies

$$\epsilon(\lambda) \leq \frac{1}{640(n_{\mathsf{pp}}(\lambda)+1)}$$

for all sufficiently large such λ, making $\Lambda^{\mathsf{res}}_{\Pi}$ infinite and contradicting full selective security. This completes the proof of Theorem 1.

6 Tight Impossibility of TS-GGAM-IBE

We now remove the restricted-key assumption of Section 5 using a compiler that replaces secret-key X-handles by explicit relative-action information. The compiler preserves public parameters and security, with a controllable correctness loss. Combining it with the restricted attack yields the same n_{pp} identity bound for general TS-GGAM-IBE, and a matching construction shows this bound to be tight over finite abelian regular actions. Throughout, identity keys contain at most $n_{\mathsf{sk}}(\lambda)$ X-handles.

6.1 Eliminating Action-Set Handles from Secret Keys

We present a compiler to transform any TS-GGAM-IBE into a scheme in which the action-set handles contained in a secret key are replaced by an explicit description of the relative-action relations needed during decryption. The compiler works in the equality-explicit normal form of Section 3.4 and may instrument Enc and Dec with proof-level anchored terms. These annotations are not visible to the instrumented algorithms.

Instrumented Decryption Transcripts. Let $n \leq n_{\mathsf{pp}}$, and fix public parameters $\mathsf{pp} = (\mathsf{pp}_{\mathsf{exp}}, \widehat{X}_1, \ldots, \widehat{X}_n)$. By the canonical handle-shape convention, $\widehat{x}_0, \widehat{X}_1, \ldots, \widehat{X}_n$ are pairwise distinct. Write $\widehat{X}_0 = \widehat{x}_0$. Fix an original identity secret key $\mathsf{sk}_{\mathsf{id}}$. Scan its handle-valued occurrences and compare them, using the cached equality interface, with the public handles $\widehat{X}_0, \ldots, \widehat{X}_n$ and with previously encountered secret-key handles. Every occurrence is thereby assigned one of two forms: $\mathsf{pub}(i)$ for some $i \in [0, n]$, or $\mathsf{sec}(j)$ for some $j \in [s]$. Here $\mathsf{pub}(i)$ means that the occurrence is equal to $\widehat{X}_i$, whereas $\widehat{Z}_1, \ldots, \widehat{Z}_s$ are the distinct secret-key handles not equal to any public handle. In particular, $s \leq n_{\mathsf{sk}}$. Let $\mathsf{shape}_{\mathsf{pp}}(\mathsf{sk}_{\mathsf{id}})$ denote the resulting explicit occurrence map, and write

$$\mathsf{sk}_{\mathsf{id}} = \mathsf{shape}_{\mathsf{pp}}\big(\mathsf{sk}_{\mathsf{id,exp}}; \widehat{Z}_1, \ldots, \widehat{Z}_s\big).$$

We associate the distinct persistent handles with the anchor tuple

$$\mathbf{A}_{\mathsf{id}} = (x_0, x_1, \ldots, x_n, z_1, \ldots, z_s),$$

where $x_0 = \widetilde{x}$, x_i is represented by $\widehat{X}_i$, and z_j is represented by $\widehat{Z}_j$. All entries of $\mathbf{A}_{\mathsf{id}}$ are distinct. Its relation matroid $M_{\mathbf{A}_{\mathsf{id}}}$ therefore has rank

$$\mathsf{rank}(M_{\mathbf{A}_{\mathsf{id}}}) \leq |\mathbf{A}_{\mathsf{id}}| - 1 = n + s \leq n_{\mathsf{pp}} + n_{\mathsf{sk}}. \tag{11}$$

Consider an execution $C \leftarrow \mathsf{Enc}^{\mathcal{O}_\star}(\mathsf{pp}, \mathsf{id}, b)$ followed by $b' \leftarrow \mathsf{Dec}^{\mathcal{O}_\star}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C)$. Instrumenting this encryption-decryption execution propagates anchored terms over $\mathbf{A}_{\mathsf{id}}$ and yields the canonical positive relation set $\mathrm{Rel}_{\mathbf{A}_{\mathsf{id}}}(\tau) \subseteq E_{\mathbf{A}_{\mathsf{id}}}$ exactly as in Definition 24.

The Handle-Elimination Compiler. Let $M \geq 1$ be a training parameter. Given a TS-GGAM-IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, we define a compiled TS-GGAM-IBE scheme $\widehat{\Pi} = (\widehat{\mathsf{Setup}}, \widehat{\mathsf{KeyGen}}, \widehat{\mathsf{Enc}}, \widehat{\mathsf{Dec}})$ as follows.

Setup and Encryption. The compiler leaves setup and encryption unchanged:

$$\widehat{\mathsf{Setup}} = \mathsf{Setup}, \qquad \widehat{\mathsf{Enc}} = \mathsf{Enc}.$$

Key Generation. On input $(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$, the compiled key-generation algorithm $\widehat{\mathsf{KeyGen}}$ first computes

$$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}^{\mathcal{O}^\star}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}),$$

and then constructs the shape $\mathsf{Shape}_{\mathsf{pp}}(\mathsf{sk}_{\mathsf{id}})$, the secret-only handles $\widehat{Z}_1, \ldots, \widehat{Z}_s$, and the anchor tuple $\mathbf{A}_{\mathsf{id}}$ as described above. For every $b \in \{0,1\}$ and $t \in [M]$, $\widehat{\mathsf{KeyGen}}$ independently samples and instruments a composed execution

$$\tau_{b,t} \leftarrow \mathsf{Exec}\left[C \leftarrow \mathsf{Enc}^{\mathcal{O}^\star}(\mathsf{pp}, \mathsf{id}, b);\ \mathsf{Dec}^{\mathcal{O}^\star}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, C)\right].$$

The instrumentation assigns anchored terms over A_{id} to all handle occurrences and extracts the canonical relation set $\mathrm{Rel}_{A_{\mathsf{id}}}(\tau_{b,t}) \subseteq E_{A_{\mathsf{id}}}$. Let

$$R_{\mathsf{train}} = \bigcup_{b \in \{0,1\}} \bigcup_{t \in [M]} \mathrm{Rel}_{A_{\mathsf{id}}}(\tau_{b,t}).$$

Fix a public total order on gain-labeled relation triples (u, v, δ), obtained, for example, by ordering the endpoint indices first and then the unique explicit representation of $\delta \in G$. Let $F_{\mathsf{id}} = \mathsf{Forest}(R_{\mathsf{train}})$ be the deterministic greedy spanning forest obtained by processing the relations in this order and retaining a relation exactly when its underlying edge connects two previously disconnected components. Every relation in R_{train} is realized by A_{id}. Consequently, $\mathrm{cl}_{A_{\mathsf{id}}}(F_{\mathsf{id}}) = \mathrm{cl}_{A_{\mathsf{id}}}(R_{\mathsf{train}})$ and $|F_{\mathsf{id}}| \leq n + s \leq n_{\mathsf{pp}} + n_{\mathsf{sk}}$. Finally, $\widehat{\mathsf{KeyGen}}$ outputs

$$\widehat{\mathsf{sk}}_{\mathsf{id}} = \left(\mathsf{sk}_{\mathsf{id},\mathsf{exp}}, \mathsf{shape}_{\mathsf{pp}}(\mathsf{sk}_{\mathsf{id}}), F_{\mathsf{id}}\right).$$

Since both $\mathsf{shape}_{\mathsf{pp}}(\mathsf{sk}_{\mathsf{id}})$ and F_{id} are explicit objects, $\widehat{\mathsf{sk}}_{\mathsf{id}}$ contains no action-set handle, as desired.

Decryption. On input $(\mathsf{pp}, \widehat{\mathsf{sk}}_{\mathsf{id}}, C)$, the compiled decryption algorithm $\widehat{\mathsf{Dec}}$ interprets the code of the original decryption algorithm Dec using a hybrid real-symbolic handle interface. A $\mathsf{pub}(i)$ occurrence is represented by the actual handle $\widehat{X}_i$; a $\mathrm{sec}(j)$ occurrence is represented by a virtual anchor z_j. A symbolic term has the form (z_j, g), with intended semantics $g \star z_j$. Such a term is an explicit internal record maintained by the interpreter; it is not a real action-set handle and is never submitted directly to the real action oracle.

For every connected component C of F_{id}, choose a canonical root r_C, preferring the least-indexed public anchor when one exists, and otherwise the least-indexed secret anchor. Let $\Delta_{F_{\mathsf{id}}}(u, v)$ denote the path gain from u to v in the forest. The normalized representation of (z_j, g) is

$$\mathsf{Norm}_{F_{\mathsf{id}}}(z_j, g) = \left(r_{\mathcal{C}}, g \circ \Delta_{F_{\mathsf{id}}}(r_{\mathcal{C}}, z_j)\right),$$

where $\mathcal{C}$ is the component containing z_j. The interpreter implements the following operations.

1. **Symbolic Action.** If $\mathsf{Norm}_{F_{\mathsf{id}}}(z_j, g) = (r, a)$, then applying $h \in G$ returns the normalized term $(r, h \circ a)$.
2. **Equality of Unmaterialized Symbolic Terms.** Two unmaterialized symbolic terms are declared equal exactly when their normalized representations are identical. Thus, terms normalized as (r, a) and (r', a') are equal exactly when $r = r'$ and $a = a'$.
3. **Materialization.** Suppose the component containing z_j has canonical root $r_{\mathcal{C}} = x_i$, where x_i is a public anchor. If $\mathsf{Norm}_{F_{\mathsf{id}}}(z_j, g) = (x_i, a)$, the symbolic term is materialized as the real handle $\mathcal{O}_\star(a, \widehat{X}_i)$, where $\widehat{X}_0 = \widehat{x}_0$ denotes the public-origin handle. All subsequent operations on this value use the resulting real handle.

4. **Operations on Real Handles.** Actions and equality tests involving only real public, ciphertext, or materialized handles are evaluated using the real action oracle and the ordinary handle-equality interface.
5. **Mixed Equality.** An unmaterialized symbolic term belongs to a forest component containing no public anchor. Such a term is declared unequal to every real public, ciphertext, or materialized handle.

Normalized pairs are internal opaque virtual handles: their explicit representations are maintained by the interpreter and are never exposed to the original Dec. The interpreted algorithm's explicit output is returned unchanged.

Lemma 7 (Correctness of Symbolic Decryption). *Fix* pp, id, *an original secret key* $\mathsf{sk}_{\mathsf{id}}$, *and an indexed anchor tuple* $\mathbf{A}_{\mathsf{id}}$. *Let* $F \subseteq E_{\mathbf{A}_{\mathsf{id}}}$ *be a gain-labeled forest realized by* $\mathbf{A}_{\mathsf{id}}$. *Consider a fresh encryption-decryption execution with genuine instrumented transcript* τ. *If* $\mathrm{Rel}_{\mathbf{A}_{\mathsf{id}}}(\tau) \subseteq \mathrm{cl}_{\mathbf{A}_{\mathsf{id}}}(F)$, *then genuine and symbolic decryption, when supplied with the same ciphertext and run using the same decryption random tape, have identical algorithm-visible equality outcomes, follow the same control flow, and produce the same explicit output.*

Proof. Couple the executions by representing public and ciphertext handles by their actual handles and each secret-only handle $\widehat{Z}_j$ by the normalized virtual anchor z_j. Actions update anchored coordinates identically. It remains to compare equalities. A positive symbolic equality between virtual terms follows from a relation in $\mathrm{cl}_{\mathbf{A}_{\mathsf{id}}}(F)$, which is true because F is realized by $\mathbf{A}_{\mathsf{id}}$; hence it is also positive genuinely. Conversely, a genuine positive equality induces a relation in $\mathrm{Rel}_{\mathbf{A}_{\mathsf{id}}}(\tau) \subseteq \mathrm{cl}_{\mathbf{A}_{\mathsf{id}}}(F)$. If both symbolic operands remain virtual, this closure relation puts them in the same forest component with identical normalized coordinates. If a genuine positive equality relates a secret-derived handle to a public, ciphertext, or materialized handle, the latter has provenance over a public anchor. The corresponding closure relation therefore connects the secret component to a public anchor, so that symbolic value is materialized and the comparison is performed on the actual real handles. Thus all equality outcomes coincide, and hence so do control flow and explicit output. This completes the proof of Lemma 7.

Theorem 6 (Secret-key Handle Elimination). *Let* Π *be a bit-message TS-GGAM-IBE scheme whose public parameters contain at most* n_{pp} *action-set handles and whose identity secret keys contain at most* n_{sk} *action-set handles. For every polynomially bounded training parameter* $M \geq 1$, *the compiler above produces a TS-GGAM-IBE scheme* $\widehat{\Pi} = (\widehat{\mathsf{Setup}}, \widehat{\mathsf{KeyGen}}, \widehat{\mathsf{Enc}}, \widehat{\mathsf{Dec}})$ *with the following properties.*

1. *The public-parameter and ciphertext distributions of* $\widehat{\Pi}$ *are identical to those of* Π.
2. *Every identity secret key of* $\widehat{\Pi}$ *contains no action-set handle.*
3. *Fix a setup output* $(\mathsf{msk}, \mathsf{pp})$, *an identity* id, *an original key* $\mathsf{sk}_{\mathsf{id}}$ *in the support of* $\mathsf{KeyGen}^{\mathcal{O}_\star}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$, *and a message* $b \in \{0,1\}$. *Let* $\widehat{\mathsf{sk}}_{\mathsf{id}}$ *be generated from this fixed original key using the compiler's random training executions. Then*
$$\Pr\Big[\widehat{\mathsf{Dec}}\big(\mathsf{pp}, \widehat{\mathsf{sk}}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, b)\big) \neq b\Big] \leq \epsilon_{\mathsf{id},b} + \frac{n_{\mathsf{pp}} + n_{\mathsf{sk}}}{M+1},$$
where $\epsilon_{\mathsf{id},b} = \Pr\big[\mathsf{Dec}\big(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, b)\big) \neq b\big]$.
4. *For every fully selective adversary* $\mathcal{A}$ *against* $\widehat{\Pi}$, *there exists a fully selective adversary* $\mathcal{B}$ *against* Π *such that* $\mathsf{Adv}^{\mathrm{sel}}_{\widehat{\Pi},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathrm{sel}}_{\Pi,\mathcal{B}}(\lambda)$. *If* $\mathcal{A}$ *is pseudoefficient, then so is* $\mathcal{B}$.

Proof. The first two properties follow immediately from the construction. We prove correctness and security separately.

Correctness. Condition on the fixed setup output, original key, identity, and message. Let $\tau_{b,1}, \ldots, \tau_{b,M}$ be the M training transcripts for b and $\tau_{b,M+1}$ an independent fresh genuine transcript. Their relation sets are i.i.d. subsets of a matroid of rank at most $n_{\mathsf{pp}} + n_{\mathsf{sk}}$ by Equation (11). Corollary 2 therefore gives
$$\Pr\left[\mathrm{Rel}_{\mathbf{A}_{\mathsf{id}}}(\tau_{b,M+1}) \not\subseteq \mathrm{cl}_{\mathbf{A}_{\mathsf{id}}}\left(\bigcup_{t\in[M]} \mathrm{Rel}_{\mathbf{A}_{\mathsf{id}}}(\tau_{b,t})\right)\right] \leq \frac{n_{\mathsf{pp}} + n_{\mathsf{sk}}}{M+1}.$$
Since F_{id} spans all training relations for both messages, the same bound holds with $\mathrm{cl}_{\mathbf{A}_{\mathsf{id}}}(F_{\mathsf{id}})$. On the complementary event, Lemma 7 makes symbolic and genuine decryption agree. A union bound yields Item 3. Averaging over setup and key-generation randomness gives the corresponding standard correctness bound.

Security. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a fully selective-security adversary against $\widehat{\Pi}$. We construct a fully selective-security adversary $\mathcal{B}$ against Π. $\mathcal{B}$ forwards $\mathcal{A}_1$'s challenge and extraction identities. Upon receiving $\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_q, C^\star$, it locally applies the compiler's training procedure to every extracted sk_i, obtaining $\widehat{\mathsf{sk}}_i$, and gives $(\mathsf{pp}, \widehat{\mathsf{sk}}_1, \ldots, \widehat{\mathsf{sk}}_q, C^\star)$ to $\mathcal{A}_2$. The distribution is exact: setup and encryption are unchanged, each original key has the correct distribution, and the added training is precisely that used by $\widehat{\mathsf{KeyGen}}$. Moving training after receipt of $C^\star$ is distributionally invisible because the TS action oracle is stateless, the semantic assignment is fixed globally, and training-only handles are never revealed to $\mathcal{A}$. Hence the advantages are equal. The simulation adds $2M$ encryption-decryption executions per extracted identity, so polynomial M preserves pseudoefficiency. This completes the proof of Theorem 6.

6.2 General Impossibility of TS-GGAM-IBE

We now combine the restricted-key impossibility theorem of Section 5 with the secret-key handle-elimination compiler of Section 6.1. Before doing so, we reconcile the setup-wise correctness condition used in the restricted attack with the standard averaged correctness definition for IBE.

From Averaged Correctness to Good Setup Worlds. Let ω denote a complete post-setup world, including the generic representation and the output $(\mathsf{pp}_\omega, \mathsf{msk}_\omega)$ of the setup algorithm. For an identity id and a message $b \in \{0,1\}$, let the conditional correctness error

$$\mathrm{err}^{\Pi}_{\omega}(\mathsf{id}, b) = \Pr\left[\mathsf{Dec}^{\mathcal{O}_\star}(\mathsf{pp}_\omega, \mathsf{sk}_{\mathsf{id}}, C) \neq b \,\middle|\, \omega\right],$$

where $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}^{\mathcal{O}_\star}(\mathsf{pp}_\omega, \mathsf{msk}_\omega, \mathsf{id})$ and $C \leftarrow \mathsf{Enc}^{\mathcal{O}_\star}(\mathsf{pp}_\omega, \mathsf{id}, b)$. The probability is over the remaining randomness of key generation, encryption, and decryption. Under the standard correctness definition, for every id and b,

$$\mathbb{E}_\omega\left[\mathrm{err}^{\Pi}_{\omega}(\mathsf{id}, b)\right] \leq \varepsilon_{\mathsf{corr}}(\lambda).$$

Definition 29 (Good Setup World). *Let S be a finite set of identities and let $\alpha > 0$. A setup world ω is (S, α)-good for Π if $\mathrm{err}^{\Pi}_{\omega}(\mathsf{id}, b) \leq \alpha$ for every $\mathsf{id} \in S$ and $b \in \{0,1\}$. We denote this event by $\mathsf{Good}_\Pi(S, \alpha)$.*

Lemma 8 (Simultaneously Good Setup Worlds). *Let Π have correctness error at most $\varepsilon_{\mathsf{corr}}(\lambda)$ under the standard averaged correctness definition. Then, for every finite identity set S and every $\alpha > 0$,*

$$\Pr_\omega\left[\neg\mathsf{Good}_\Pi(S, \alpha)\right] \leq \frac{2|S|\varepsilon_{\mathsf{corr}}(\lambda)}{\alpha}.$$

Proof. Fix $\mathsf{id} \in S$ and $b \in \{0,1\}$. By Markov's inequality,

$$\Pr_\omega\left[\mathrm{err}^{\Pi}_{\omega}(\mathsf{id}, b) > \alpha\right] \leq \frac{\mathbb{E}_\omega[\mathrm{err}^{\Pi}_{\omega}(\mathsf{id}, b)]}{\alpha} \leq \frac{\varepsilon_{\mathsf{corr}}(\lambda)}{\alpha}.$$

There are $2|S|$ identity-message pairs. A union bound over these pairs gives

$$\Pr_\omega\left[\exists\, \mathsf{id} \in S,\ b \in \{0,1\} : \mathrm{err}^{\Pi}_{\omega}(\mathsf{id}, b) > \alpha\right] \leq \frac{2|S|\varepsilon_{\mathsf{corr}}(\lambda)}{\alpha}.$$

This is precisely the claimed bound. This completes the proof of Lemma 8.

Lemma 9 (Conditional Correctness after Handle Elimination). *Let $\widehat{\Pi}$ be obtained from Π using the handle-elimination compiler with training parameter M. Suppose that the public parameters and identity secret keys of Π contain at most n_{pp} and n_{sk} action-set handles, respectively. Then, for every setup world ω, identity id, and $b \in \{0,1\}$,*

$$\mathrm{err}^{\widehat{\Pi}}_{\omega}(\mathsf{id}, b) \leq \mathrm{err}^{\Pi}_{\omega}(\mathsf{id}, b) + \frac{n_{\mathsf{pp}} + n_{\mathsf{sk}}}{M+1}.$$

Proof. Theorem 6 gives the claimed inequality for every fixed setup world and fixed original secret key. Averaging that inequality over the randomness of the original key-generation algorithm gives

$$\mathrm{err}_{\omega}^{\widehat{\Pi}}(\mathsf{id}, b) \le \mathrm{err}_{\omega}^{\Pi}(\mathsf{id}, b) + \frac{n_{\mathsf{pp}} + n_{\mathsf{sk}}}{M+1}.$$

This completes the proof of Lemma 9.

Lemma 10 (Setup-wise Restricted Attack). *Let Π' be a restricted TS-GGAM-IBE scheme whose public parameters contain at most n action-set handles. Set $k = (n+1)$ and $\eta = 1/(640k)$. Let $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_k)$ publicly computable ordered tuple of k distinct identities, and let $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_k\}$ be set underlying the ordered tuple $\mathbf{id}$. There exists a pseudoefficient fully selective adversary $\mathcal{A}_{\mathsf{res}}$ with the following properties.*

1. *For every setup world ω satisfying $\mathrm{err}_{\omega}^{\Pi'}(\mathsf{id}, b) \le \eta$ for every $\mathsf{id} \in S$ and $b \in \{0,1\}$,*

$$\Pr[\mathcal{A}_{\mathrm{res}} \text{ wins} \mid \omega] \ge \frac{1}{2} + \frac{1}{40k} - \mathrm{negl}(\lambda).$$

2. *For every setup world ω, without any correctness assumption,*

$$\Pr[\mathcal{A}_{\mathrm{res}} \text{ wins} \mid \omega] \ge \frac{1}{2} - \mathrm{negl}(\lambda).$$

Proof. We first prove Item 1. Run the adversary of Theorem 1 using the public upper bound n, even if the realized public parameters contain only $d \le n$ non-origin handles. Their relation matroid has rank at most d, while $k = (n+1)$; hence

$$\frac{k-d}{2k} \ge \frac{1}{2k}.$$

The proof of Theorem 1 uses only this lower bound and setup-wise correctness at most η. This completes the proof of Item 1.

For Item 2, condition on an arbitrary setup world and prefix. If $\mathsf{FindCandidate}$ returns $\perp$, the adversary guesses uniformly. Otherwise validation is independent of the challenge: a key with average success below $1/2$ passes with probability $e^{-\Omega(\lambda)}$, whereas using a key with success at least $1/2$ cannot reduce the winning probability below $1/2$. Thus the lower bound is $1/2 - e^{-\Omega(\lambda)}$. This completes the proof of Item 2, and hence the proof of Lemma 10.

The General Impossibility Theorem. We now prove the following theorem for the impossibility of TS-GGAM-IBE, thus removing the restriction on the form of identity secret keys from Theorem 1.

Theorem 7 (General TS-GGAM-IBE Impossibility). *Let Π be a bit-message TS-GGAM-IBE scheme satisfying the following conditions.*

1. *Its public parameters contain at most $n_{\mathsf{pp}} = n_{\mathsf{pp}}(\lambda)$ action-set handles.*
2. *Every identity secret key contains at most $n_{\mathsf{sk}} = n_{\mathsf{sk}}(\lambda)$ action-set handles.*
3. *The functions n_{pp} and n_{sk} are publicly known, efficiently computable, and polynomially bounded.*
4. *Its standard averaged correctness error $\epsilon_{\mathsf{corr}}(\lambda)$ is negligible.*
5. *Its identity-space family $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda \in \mathbb{N}}$ is efficiently indexable.*

Define the set of security parameters $\Lambda_\Pi = \{\lambda \in \mathbb{N} : |\mathsf{ID}_\lambda| \ge n_{\mathsf{pp}}(\lambda) + 1\}$. Then there exists a single pseudoefficient fully selective adversary $\mathcal{B}$ such that, for every $\lambda \in \Lambda_\Pi$,

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{B}}(\lambda) \ge \frac{1}{40\big(n_{\mathsf{pp}}(\lambda)+1\big)} - \mathrm{negl}(\lambda).$$

Consequently, if Λ_Π is infinite, then Π is not fully selective-secure.

Proof. We first describe a single adversary that is defined uniformly at every security parameter. Let $n = n_{\mathsf{pp}}(\lambda)$, $s = n_{\mathsf{sk}}(\lambda)$ and $k = (n+1)$. Run $\mathbf{id} \leftarrow \mathsf{IndexID}(1^\lambda, 1^k)$. If $\mathbf{id} = \bot$, the adversary uses the uniform-guessing fallback strategy. Otherwise, let $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_k)$, and let $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_k\}$ be set underlying the ordered tuple $\mathbf{id}$. Let $\eta = 1/(640k)$. We now analyze an arbitrary fixed $\lambda \in \Lambda_\Pi$. Let $\mathsf{Good} = \mathsf{Good}_\Pi(S, \eta/2)$. By Lemma 8,

$$\Pr_\omega[\neg\mathsf{Good}] \le \frac{2k\varepsilon_{\mathsf{corr}}(\lambda)}{\eta/2} = 2560k^2\varepsilon_{\mathsf{corr}}(\lambda) \implies \Pr_\omega[\mathsf{Good}] \ge 1 - 2560k^2\varepsilon_{\mathsf{corr}}(\lambda).$$

Apply the secret-key handle-elimination compiler with training parameter $M = \max\{1, \lceil 1280k(n+s)\rceil\}$. Let $\widehat{\Pi}$ denote the resulting scheme. Since n, s, and k are polynomially bounded, so is M. By Lemma 9, for every $\omega \in \mathsf{Good}$, $\mathsf{id} \in S$, and $b \in \{0,1\}$,

$$\mathrm{err}^{\widehat{\Pi}}_\omega(\mathsf{id}, b) \le \mathrm{err}^{\Pi}_\omega(\mathsf{id}, b) + \frac{n+s}{M+1} \le \frac{\eta}{2} + \frac{1}{1280k} = \eta.$$

The public parameters of $\widehat{\Pi}$ are identical to those of Π and therefore contain at most n action-set handles, while its identity secret keys contain no action-set handles. Apply the adversary $\mathcal{A}_{\mathsf{res}}$ from Lemma 10 to $\widehat{\Pi}$. Lemma 10 gives

$$\begin{aligned}
\Pr[\mathcal{A}_{\mathsf{res}} \text{ wins}] &\ge \Pr[\mathsf{Good}]\left(\frac{1}{2} + \frac{1}{40k}\right) + \Pr[\neg\mathsf{Good}]\frac{1}{2} - \mathsf{negl}(\lambda) \\
&= \frac{1}{2} + \frac{\Pr[\mathsf{Good}]}{40k} - \mathsf{negl}(\lambda) \\
&\ge \frac{1}{2} + \frac{1 - 2560k^2\varepsilon_{\mathsf{corr}}(\lambda)}{40k} - \mathsf{negl}(\lambda).
\end{aligned}$$

Thus, for every $\lambda \in \Lambda_\Pi$,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{sel}}_{\widehat{\Pi},\mathcal{A}_{\mathsf{res}}}(\lambda) &\ge \frac{1 - 2560k^2\epsilon_{\mathsf{corr}}(\lambda)}{40k} - \mathrm{negl}(\lambda) \\
&= \frac{1}{40k} - 64k\epsilon_{\mathsf{corr}}(\lambda) - \mathrm{negl}(\lambda), \\
&= \frac{1}{40k} - \mathrm{negl}(\lambda),
\end{aligned}$$

where the last equality holds because $k = n_{\mathsf{pp}}(\lambda) + 1$ is polynomially bounded and ϵ_{corr} is negligible. Finally, the security-preservation part (Item 4) of Theorem 6 transfers this advantage to a pseudoefficient adversary $\mathcal{B}$ against Π. Consequently, for every $\lambda \in \Lambda_\Pi$,

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{B}}(\lambda) \ge \frac{1}{40\big(n_{\mathsf{pp}}(\lambda)+1\big)} - \mathrm{negl}(\lambda).$$

The adversary $\mathcal{B}$ is one uniform oracle machine: outside Λ_Π it follows the uniform-guessing branch specified above, whereas on every parameter in Λ_Π it executes the restricted attack. If Λ_Π is infinite, then, for all sufficiently large $\lambda \in \Lambda_\Pi$,

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{B}}(\lambda) \ge \frac{1}{80\big(n_{\mathsf{pp}}(\lambda)+1\big)}.$$

This lower bound is inverse-polynomial on infinitely many security parameters, contradicting fully selective security. This completes the proof of Theorem 7.

Corollary 6 (No Large-Identity-Space TS-GGAM-IBE). *Let Π be a fully selective-secure TS-GGAM-IBE scheme with negligible correctness error and polynomially many action-set handles in its public parameters and identity secret keys. Suppose that its identity space is efficiently indexable in the sense of Section 4.1. Then $|\mathsf{ID}_\lambda| \le n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ. In particular, no such scheme with polynomial-size public parameters can support a super-polynomial efficiently indexable identity space.*

Proof. Suppose, toward a contradiction, that $|\mathsf{ID}_\lambda| \geq n_{\mathsf{pp}}(\lambda) + 1$ for infinitely many security parameters λ. Equivalently, the set Λ_{Π} defined in Theorem 7 is infinite, and that theorem gives a single adversary with inverse-polynomial advantage infinitely often, contradicting the assumed fully selective security of Π. It follows that Λ_{Π} is finite, and hence $|\mathsf{ID}_\lambda| \leq n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ. This completes the proof of Corollary 6.

Remark 10 (Quantitative Nature of the Bound). The identity bound depends on the number n_{pp} of public X-handles—not their bit length or the number n_{sk} of secret-key handles. The latter affects only the compiler's training complexity. After elimination of action-set handles from identity secret keys, the restricted attack is governed solely by the public-anchor rank.

6.3 Tightness of the Public-Handle Bound

We finally show that the n_{pp} bound of Section 6.2 is tight over finite abelian regular actions.

Construction of Restricted TS-GGAM-IBE. Let $\mathsf{GA}_\lambda = (G_\lambda, \mathcal{X}_\lambda, \star_\lambda, \widetilde{x}_\lambda)$ be a finite abelian regular group action, and write $N_\lambda = |G_\lambda|$. We omit the subscript λ when it is clear from context. Let $n = n(\lambda)$ be polynomially bounded and let the identity space be $\mathsf{ID}_\lambda = [n]$. We define the scheme $\Pi_{\mathsf{match}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ as follows.

$\mathsf{Setup}^{\mathcal{O}^\sigma_\star}(1^\lambda, \mathsf{gpar}_\lambda, \widehat{x}_0)$**:** Sample independent elements $a_1, \ldots, a_n \leftarrow G$ and compute

$$\widehat{P}_i \leftarrow \mathcal{O}^\sigma_\star(a_i, \widehat{x}_0) = \sigma(a_i \star \widetilde{x})$$

for every $i \in [n]$. Let $\widehat{\mathbf{P}} = (\widehat{P}_1, \ldots, \widehat{P}_n)$ denote the indexed vector of public-handle occurrences. This vector may contain repeated occurrences or occurrences equal to the public origin $\widehat{x}_0$. Let $\widehat{Q}_1, \ldots, \widehat{Q}_t$, where $t \leq n$, be the distinct non-origin handles occurring in $\widehat{\mathbf{P}}$, and let $\mathsf{shape}_{\mathbf{P}}$ be the explicit occurrence map that associates every position $i \in [n]$ with either the origin index 0 or the corresponding handle $\widehat{Q}_j$. Output

$$\mathsf{msk} = (a_1, \ldots, a_n), \qquad \mathsf{pp} = \big(n, \mathsf{shape}_{\mathbf{P}}; \widehat{Q}_1, \ldots, \widehat{Q}_t\big).$$

For readability, we continue to write $\widehat{P}_i$ for the i-th handle occurrence encoded by $\mathsf{shape}_{\mathbf{P}}$; this notation does not assert that the occurrences $\widehat{P}_1, \ldots, \widehat{P}_n$ are pairwise distinct.

$\mathsf{KeyGen}^{\mathcal{O}^\sigma_\star}(\mathsf{pp}, \mathsf{msk}, i)$**:** Output as the secret key corresponding to identity-i (where $i \in [n]$) the explicit group element

$$\mathsf{sk}_i = a_i.$$

$\mathsf{Enc}^{\mathcal{O}^\sigma_\star}(\mathsf{pp}, i, b \in \{0,1\})$**:** Sample $r \leftarrow G$ and compute $\widehat{C}_1 \leftarrow \mathcal{O}^\sigma_\star(r, \widehat{x}_0)$. If $b = 0$, compute $\widehat{C}_2 \leftarrow \mathcal{O}^\sigma_\star(r, \widehat{P}_i)$. If $b = 1$, sample an independent $s \leftarrow G$ and compute $\widehat{C}_2 \leftarrow \mathcal{O}^\sigma_\star(s, \widehat{x}_0)$. Output the ciphertext

$$C = (\widehat{C}_1, \widehat{C}_2).$$

$\mathsf{Dec}^{\mathcal{O}^\sigma_\star}(\mathsf{pp}, \mathsf{sk}_i = a_i, C = (\widehat{C}_1, \widehat{C}_2))$**.** Compute

$$\widehat{D} \leftarrow \mathcal{O}^\sigma_\star(a_i, \widehat{C}_1).$$

Output 0 if $\mathsf{Eq}_{\mathcal{X}}(\widehat{D}, \widehat{C}_2) = 1$, and output 1 otherwise.

Remark 11. The above TS-GGAM-IBE scheme is indeed restricted: observe that every identity secret key consists solely of an explicitly represented element of G, while the public parameters contain at most n distinct non-origin action-set handles.

Lemma 11 (Planted-Parallelogram Indistinguishability). *Let $(G, \mathcal{X}, \star, \widetilde{x})$ be a finite abelian regular group action with $N = |G|$. Sample independent $a, r, s \leftarrow G$, and let $\widehat{P} = \sigma(a \star \widetilde{x})$ and $\widehat{R} = \sigma(r \star \widetilde{x})$. Consider the following two experiments:*

$$\mathsf{Hyb}_0 : \quad \widehat{Z} = \sigma((a \circ r) \star \widetilde{x}), \qquad \mathsf{Hyb}_1 : \quad \widehat{Z} = \sigma(s \star \widetilde{x}).$$

The algorithm may additionally receive arbitrary auxiliary handles $\sigma(g_j \star \widetilde{x})$ and associated explicit side information jointly distributed independently of a, r, s; the g_j's themselves may also be revealed. If the equality-saturated computation encounters at most T input or action-response occurrences, then

$$\Delta(\mathsf{Hyb}_0, \mathsf{Hyb}_1) \leq \frac{2\binom{T}{2}}{N}.$$

Proof. Condition on the algorithm's tape, explicit side information, and all auxiliary coordinates. Treat the origin, auxiliary handles, and their descendants as a common background subsystem and couple all equalities within it identically. Equalities within each of the challenge-dependent anchor classes P, R, Z are likewise syntactically fixed. Define a common formal execution by answering every remaining, non-forced cross-class/background equality negatively. A real execution differs from it only if such a pair collides. For any fixed pair, conditioning on all but one of the independent uniform values a, r, s reduces the equality in either hybrid to fixing a uniform group element (or a quotient/product of independent uniform elements), so its probability is at most $1/N$. There are at most $\binom{T}{2}$ pairs. Thus the bad event has probability at most $\binom{T}{2}/N$ in either hybrid. Outside these events the equality patterns are identical and renaming invariance couples the explicit outputs, yielding the stated factor 2. This completes the proof of Lemma 11.

Theorem 8 (Matching n-Identity Construction of Restricted TS-GGAM-IBE). *Let $\mathsf{GA}_\lambda = (G_\lambda, \mathcal{X}_\lambda, \star_\lambda, \widetilde{x}_\lambda)$ be a family of finite abelian regular group actions, and let $N_\lambda = |G_\lambda|$. For every polynomially bounded $n = n(\lambda)$, the scheme Π_{match} above is a restricted TS-GGAM-IBE scheme over the identity space $\mathsf{ID}_\lambda = [n]$ with the following properties:*

1. *Its public parameters contain at most n distinct non-origin action-set handles.*
2. *Its pointwise correctness error is at most $1/N_\lambda$.*
3. *For every fully selective type-safe adversary $\mathcal{A}$ making at most $q = q(\lambda)$ action-oracle queries,*

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi_{\mathsf{match}}, \mathcal{A}}(\lambda) \leq \frac{2\binom{q+n+3}{2}}{N_\lambda}.$$

Consequently, if N_λ^{-1} is negligible, then Π_{match} has negligible correctness error and is fully selective-secure against pseudoefficient TS-GGAM adversaries.

Proof. The public-handle and secret-key claims follow immediately from the construction. We prove correctness and fully selective-security below.

Correctness. Fix a setup output and an identity $i \in [n]$. If $b = 0$, then abelianness gives

$$\mathcal{O}_\star(a_i, \widehat{C}_1) = \sigma((a_i \circ r) \star \widetilde{x}) = \sigma((r \circ a_i) \star \widetilde{x}) = \mathcal{O}_\star(r, \widehat{P}_i) = \widehat{C}_2.$$

Thus, encryption of 0 is always decrypted correctly. If $b = 1$, decryption outputs 0 exactly when $s = a_i \circ r$. For every fixed a_i and r, this occurs with probability $1/N_\lambda$ over the independent choice of s. The pointwise correctness error is therefore at most $1/N_\lambda$.

Security. We now prove fully selective-security. Fix a fully selective adversary $\mathcal{A}$ and condition on a valid challenge identity $i^\star$ selected by its first stage. Condition further on the non-challenge setup exponents $(a_j)_{j \neq i^\star}$ and on the complete pre-setup execution of the first stage. This conditioning fixes the coordinates of every non-challenge public handle and every handle retained in the adversary's first-stage state. Moreover, these coordinates are jointly independent of the challenge exponent $a_{i^\star}$ and of the encryption randomness r, s. Consequently, all such handles are auxiliary origin-derived handles of the form permitted by Lemma 11.

Concretely, write $a = a_{i^\star}$. The remaining challenge-dependent handles have the semantic coordinates $\widehat{P}_{i^\star} = a \star \widetilde{x}$, $\widehat{C}_1 = r \star \widetilde{x}$, and

$$\widehat{C}_2 = \begin{cases} (r \circ a) \star \widetilde{x}, & b = 0, \\ s \star \widetilde{x}, & b = 1. \end{cases}$$

Because G is abelian, this is precisely the pair of experiments in Lemma 11.

The computation begins with the public origin, at most n public handle occurrences, and the two ciphertext handles. The adversary subsequently receives at most one new action-response occurrence per action query. Hence its complete equality-saturated computation encounters at most $T \le (q + n + 3)$ input or action-response occurrences. Applying Lemma 11 bounds the statistical distance between the adversary's views for $b = 0$ and $b = 1$ by $2\binom{q+n+3}{2}/N$. The adversary's fully selective advantage is no larger than this statistical distance, proving the claimed bound. If q and n are polynomially bounded and N^{-1} is negligible, the bound is negligible. This completes the proof of Theorem 8.

Corollary 7 (Tightness of the Public-Handle Bound). *Over families of finite abelian regular group actions for which $|G_\lambda|^{-1}$ is negligible, for every polynomially bounded $n = n(\lambda)$:*

1. *There exists a fully selective-secure restricted TS-GGAM-IBE scheme supporting n identities and using at most n non-origin public handles; whereas*
2. *No fully selective-secure TS-GGAM-IBE scheme with at most n non-origin public handles and negligible correctness error can support an efficiently indexable identity-space family of size at least $(n + 1)$ for infinitely many security parameters.*

Proof. The first statement follows from Theorem 8. The second follows by specializing Corollary 6 to $n_{\mathsf{pp}}(\lambda) = n(\lambda)$. This completes the proof of Corollary 7.

7 A Tight Impossibility Result for Sanitizable RSR-GGAM-IBE

We now extend the TS-GGAM framework to random-string representations, where algorithms may inspect literal label bits and different invocations may share the same representation state. For restricted key-label-oblivious, composably label-sanitizable RSR-GGAM-IBE, we recover the same tight public-parameter bound as in the type-safe setting.

7.1 Lazy Random-String Representations

We first replace the fixed random injection of the RSR-GGAM by an equivalent lazy allocation process, up to the standard label-slack loss.

The Lazy-Label Oracle. Let $\Sigma_\lambda = \{0,1\}^{\ell(\lambda)}$ and $M_\lambda = |\Sigma_\lambda| = 2^{\ell(\lambda)}$. The lazy-label oracle maintains a partial injective table $\mathsf{LT} \colon \mathcal{X} \rightharpoonup \Sigma_\lambda$ and a rejection set $\mathsf{Rej} \subseteq \Sigma_\lambda \setminus \mathrm{im}(\mathsf{LT})$. The table records the labels that have already been allocated, whereas Rej records strings that have previously been submitted as invalid labels. A string placed in Rej is never subsequently used as the label of a semantic element. Whenever the oracle must expose a semantic element $x \notin \mathrm{dom}(\mathsf{LT})$, it samples $L \leftarrow \Sigma_\lambda \setminus \big(\mathrm{im}(\mathsf{LT}) \cup \mathsf{Rej}\big)$, sets $\mathsf{LT}(x) = L$, and returns L. On an action query (g, L), the oracle proceeds as follows. If $L \notin \mathrm{im}(\mathsf{LT})$, it adds L to Rej and returns $\perp$. Otherwise, it recovers the unique x satisfying $\mathsf{LT}(x) = L$, computes $y = g \star x$, allocates a label for y if necessary, and returns $\mathsf{LT}(y)$. Equality of valid labels is ordinary string equality.

We refer to the resulting model as the *lazy-label RSR-GGAM*, abbreviated LL-GGAM. Notice that rejected strings must be remembered: if a rejected string could later be allocated to a semantic element, the same string would be invalid at one point in the execution and valid at a later point.

Lemma 12 (Lazy-Label Equivalence). *Let $\mathcal{A}$ be an arbitrary, possibly computationally unbounded, algorithm whose execution begins with at most p exposed semantic elements and makes at most q oracle queries. Let $N = |\mathcal{X}|$, and suppose that $M_\lambda > p + q$. Then $\Delta\Big(\mathsf{View}_{\mathcal{A}}^{\mathsf{RSR}}, \mathsf{View}_{\mathcal{A}}^{\mathsf{LL}}\Big) \le \varepsilon_{\mathsf{lbl}}(p, q)$, where*

$$\varepsilon_{\mathsf{lbl}}(p, q) = \min\left\{1,\ \frac{qN}{M_\lambda - p - q}\right\}.$$

Here the first experiment uses a uniformly random injective representation of the entire semantic domain, whereas the second uses the lazy-label oracle described above.

Proof. Couple the experiments by revealing the fixed random injection only when a semantic element is first encountered, using the same label as the lazy experiment. The coupling can fail only if $\mathcal{A}$ submits a currently unallocated string that, in the fixed experiment, is the label of an unseen semantic element. Conditioned on no earlier failure, at least $(M_\lambda - p - q)$ strings remain undetermined, of which at most N label unseen elements. Hence each query causes failure with probability at most $N/(M_\lambda - p - q)$, and a union bound over at most q queries completes the proof of Lemma 12.

Remark 12. Lemma 12 also applies when several generic algorithms share one representation oracle, with p and q counting, respectively, all initially exposed labels and all oracle queries in the composed execution.

Corollary 8 (Provenance of Valid Labels). *Except with probability at most $\varepsilon_{\mathsf{lbl}}(p, q)$, every valid label accepted during an RSR-GGAM execution is either initially exposed or was previously returned by the action oracle.*

7.2 Label-Synchronous Transcript Transfer

Unlike opaque TS handles, RSR labels may be inspected bitwise. Semantic transcript equivalence must therefore be strengthened to *literal-label synchrony*. For two lazy-label executions with encountered semantic domains D_0, D_1 and correspondence $\pi : D_0 \to D_1$, their label tables are *synchronous* if $\mathsf{LT}_1(\pi(x)) = \mathsf{LT}_0(x)$ for all $x \in D_0$, and their rejection sets are identical.

Lemma 13 (Label-Synchronous Transfer). *For each $b \in \{0, 1\}$, let $\mathsf{LT}_b^{\mathsf{init}} : \mathcal{D}_b^{\mathsf{init}} \hookrightarrow \Sigma_\lambda$ be the initial lazy-label table of execution b, where $\mathcal{D}_b^{\mathsf{init}}$ is exactly the set of semantic elements initially exposed to that execution. Suppose that the two semantic executions admit a type-preserving correspondence $\pi : \mathcal{D}_0 \longrightarrow \mathcal{D}_1$ that restricts to a bijection $\mathcal{D}_0^{\mathsf{init}} \to \mathcal{D}_1^{\mathsf{init}}$ and preserves initially exposed elements, action-query input-output relations, and semantic equality. If $\mathsf{LT}_1^{\mathsf{init}}(\pi(x)) = \mathsf{LT}_0^{\mathsf{init}}(x)$ for every $x \in \mathcal{D}_0^{\mathsf{init}}$, and the initial rejection sets are identical, then the executions can be coupled to have identical complete bit-level transcripts.*

Proof. Maintain that encountered semantic elements correspond under π, their labels are identical, the rejection sets coincide, and the algorithms have identical bit-level views. Hence they issue the same next query. An unallocated submitted string is rejected in both executions. Otherwise it labels corresponding semantic elements, and preservation of the action relation gives corresponding outputs. If those outputs were previously encountered, preservation of equality implies identical existing labels; if both are new, assign them one common fresh string outside the common table image and rejection set. The invariant therefore persists throughout the execution. This completes the proof of Lemma 13.

Anchored Transcripts for Lazy-Label Executions. For an execution starting from anchors $\mathbf{x} = (x_0, \ldots, x_d)$, annotate every valid initial-label and action-response occurrence by its anchored term as in Section 5.1. The equality-saturated transcript additionally records literal labels, and its positive relation set is defined as in Definition 25. Invalid-label queries have no anchored term and induce no semantic relation.

Lemma 14 (Label-Synchronous Gain-Graph Transfer). *Let $\mathcal{B}$ be a deterministic RSR-GGAM algorithm with fixed ordinary input and fixed random tape. Consider two lazy-label executions of $\mathcal{B}$ on respective semantic anchor tuples $\mathbf{x} = (x_0, \ldots, x_d)$ and $\mathbf{y} = (y_0, \ldots, y_d)$. Suppose that the domains of the two initial lazy-label tables are exactly $\{x_0, \ldots, x_d\}$ and $\{y_0, \ldots, y_d\}$, respectively, that these tables are synchronous under $x_i \leftrightarrow y_i$, and that the two initial rejection sets are identical. Let $\mathsf{Tr}_{\mathbf{x}}$ and $\mathsf{Tr}_{\mathbf{y}}$ be the complete equality-saturated lazy-label transcripts. Suppose that there exists a balanced gain graph Γ on vertex set $\{0, \ldots, d\}$ such that both $\mathbf{x}$ and $\mathbf{y}$ realize Γ, and such that $\mathsf{PosRel}(\mathsf{Tr}_{\mathbf{x}}) \cup \mathsf{PosRel}(\mathsf{Tr}_{\mathbf{y}}) \subseteq \mathrm{cl}(\Gamma)$. Then the two executions can be coupled so that their complete bit-level transcripts are identical.*

Proof. As in Lemma 13, couple corresponding occurrences by their anchored terms. Suppose corresponding terms are (i, g) and (j, h). If they are equal in the $\mathbf{x}$-execution, the induced positive relation lies in $\mathrm{PosRel}(\mathrm{Tr}_{\mathbf{x}}) \subseteq \mathrm{cl}(\Gamma)$; since $\mathbf{y}$ realizes Γ, the corresponding terms are equal there. The converse follows symmetrically. Thus equality is preserved and reflected.

Consequently, corresponding valid action queries have corresponding semantic outputs. Either both outputs were previously encountered or both are new; otherwise the resulting equality in one transcript would, by the same two-sided argument, force the corresponding equality in the other. In the new case assign one common fresh label to both outputs. Invalid submitted strings are identical in the two executions and are therefore rejected in both. Hence the semantic correspondence preserves initial elements, action relations, and equality, while the label tables and rejection sets remain synchronous. Lemma 13 now gives identical bit-level transcripts.

As usual, the reference to complete transcripts is well founded: a first divergence would be preceded by identical views, and the positive relation causing an equality in one execution would force the same equality in the other. This completes the proof of Lemma 14.

Corollary 9 (Gain-Graph Transfer with Random Labels). *Let R be a canonical relation set realized by both anchor tuples, and let Γ_R be its common gain graph. If $\mathsf{Rel}_{\mathbf{x}}(\mathsf{Tr}_{\mathbf{x}}) \subseteq \mathrm{cl}_{\mathbf{x}}(R)$ and $\mathsf{Rel}_{\mathbf{y}}(\mathsf{Tr}_{\mathbf{y}}) \subseteq \mathrm{cl}_{\mathbf{y}}(R)$, then the two lazy-label executions have identical complete bit-level transcripts under the coupling of Lemma 14.*

Proof. As in Corollary 4, relation-set containment implies

$$\mathsf{PosRel}(\mathsf{Tr}_{\mathbf{x}}) \cup \mathsf{PosRel}(\mathsf{Tr}_{\mathbf{y}}) \subseteq \mathrm{cl}(\Gamma_R).$$

Now, applying Lemma 14 completes the proof of Corollary 9.

7.3 Forgetting Private Labels

A scheme invocation may allocate labels that are not retained in its structured output, yet their literal values could still influence retained ordinary state. We now show that the private-label symmetry defined in Section 4.2 permits such transient labels to be safely forgotten.

Fix a completed lazy-label prefix satisfying private-label symmetry, with notation $(K, \tau_{\mathsf{sem}}, LT_{\mathsf{pub}}, LT_{\mathsf{priv}}, \mathsf{Rej}, D_{\mathsf{priv}})$ as in Section 4.2. The real continuation retains $(LT_{\mathsf{pub}}, LT_{\mathsf{priv}}, \mathsf{Rej})$, whereas the sanitized continuation retains only K, LT_{pub} and resets $(LT_{\mathsf{priv}}, \mathsf{Rej}) \mapsto (\emptyset, \emptyset)$. The continuation itself receives only the retained state and retained labels.

Lemma 15 (Private-Label Forgetting). *Suppose that the prefix satisfies private-label symmetry and that $|D_{\mathsf{pub}}| \le p$ and $|D_{\mathsf{priv}}| = a$, $|\mathsf{Rej}| = r$. Let $\mathcal{C}$ be any possibly computationally unbounded continuation that makes at most q oracle queries. If $M_\lambda > (p + a + r + q)$, then*

$$\Delta\Big(\mathsf{View}_{\mathcal{C}}^{\mathsf{real}}, \mathsf{View}_{\mathcal{C}}^{\mathsf{san}}\Big) \le \varepsilon_{\mathsf{forget}}(p, a, r, q),$$

where $\varepsilon_{\mathsf{forget}}(p, a, r, q) = \min\{1,\ (q(a + r)) \,/\, (M_\lambda - p - a - r - q)\}$.

Proof. Condition on the retained information in Definition 14. By symmetry, $\mathsf{LT}_{\mathsf{priv}}$ is a uniform injection into the unused label space. Reveal its entries only when semantically necessary and couple the real and sanitized continuations until either: (i) the continuation submits the still-hidden label of an unrevealed private element; or (ii) a fresh allocation in the sanitized execution selects either a hidden private label or a string from the forgotten rejection set.

Semantically reaching an element of D_{priv} is not itself bad: reveal its deferred label and return it in both executions. Likewise, if a submitted string belongs to the forgotten rejection set, both executions reject it. Conditioned on no prior failure, at most $(a + r)$ strings remain reserved as unrevealed private labels or forgotten rejected strings, while at least $(M_\lambda - p - a - r - q)$ strings remain possible. For any query, the two bad cases above are mutually exclusive, so its failure probability is at most $(a + r)/(M_\lambda - p - a - r - q)$. A union bound over at most q queries gives the claim. Before failure, both continuations have identical labels, oracle answers, and ordinary outputs. The coupling bound therefore implies the claimed statistical-distance bound. This completes the proof of Lemma 15.

Corollary 10 (Negligible Forgetting Loss). *If p, a, r, and q are polynomially bounded and $\ell(\lambda) = \Omega(\lambda)$, then $\varepsilon_{\mathsf{forget}}(p, a, r, q) = \mathsf{negl}(\lambda)$.*

Application to IBE. Definition 15 supplies the required symmetry after the setup-key-generation prefix, while Definition 16 supplies it at later sanitization boundaries. Hence transient labels can be forgotten between composed computations. The next lemma formalizes repeated resetting.

Lemma 16 (Composable Resetting). *Let $\mathcal{C}$ be a polynomially bounded label-erasing composition having a set $\mathcal{J}$ of invocation boundaries at which sanitization is to be performed. Suppose that the composition satisfies Definition 16 at every such boundary, including in executions in which any subset of preceding boundaries has already been sanitized. For every $j \in \mathcal{J}$, let $|D_{\mathsf{live},j}| \le p_j$, $|D_{\mathsf{priv},j}| \le a_j$, and $|\mathsf{Rej}_j| \le r_j$, and let Q_j bound the total number of action-oracle queries made after boundary j. Assume $M_\lambda > p_j + a_j + r_j + Q_j$ for every $j \in \mathcal{J}$. Let $\mathsf{View}_{\mathcal{C}}^{\mathsf{share}}$ denote the view of the ordinary lazy-label execution, in which all invocations share one evolving lazy-label state. Let $\mathsf{View}_{\mathcal{C}}^{\mathsf{reset}}$ denote the execution obtained by performing, at every $j \in \mathcal{J}$, the following sanitization: $(\mathsf{LT}_{\mathsf{live},j}, \mathsf{LT}_{\mathsf{priv},j}, \mathsf{Rej}_j) \longmapsto (\mathsf{LT}_{\mathsf{live},j}, \varnothing, \varnothing)$. Then*

$$\Delta\Big(\mathsf{View}_{\mathcal{C}}^{\mathsf{share}}, \mathsf{View}_{\mathcal{C}}^{\mathsf{reset}}\Big) \le \sum_{j \in \mathcal{J}} \epsilon_{\mathsf{forget}}(p_j, a_j, r_j, Q_j).$$

Proof. Order J chronologically and let H_t sanitize exactly the first t boundaries. For adjacent hybrids H_{t-1}, H_t, condition on the common execution prefix up to the next boundary j. Definition 16 gives the private-label symmetry required by Lemma 15, even after preceding boundaries have already been sanitized. Since the remaining computation makes at most Q_j queries, $\Delta(H_{t-1}, H_t) \le \epsilon_{\mathsf{forget}}(p_j, a_j, r_j, Q_j)$. Summing over the hybrids proves the claim. Unreached boundaries may be padded by zero-query dummy blocks and contribute zero. This completes the proof of Lemma 16.

Corollary 11 (Negligible Resetting Loss). *If $|\mathcal{J}|$ and p_j, a_j, r_j, Q_j are all polynomially bounded, and $\ell(\lambda) = \Omega(\lambda)$, then $\sum_{j \in \mathcal{J}} \epsilon_{\mathsf{forget}}(p_j, a_j, r_j, Q_j) = \mathrm{negl}(\lambda)$.*

Proof. Apply Corollary 10 at every boundary and use that a polynomial sum of negligible functions is negligible. This completes the proof of Corollary 11.

7.4 Symmetry-Preserving Offline Candidate Selection

The offline search of Section 5.4 must not select according to literal private labels, since this could destroy private-label symmetry. We therefore enumerate setup-key-generation prefixes only through summaries that erase those labels.

Let Pref be a completed lazy-label prefix. Its retained symbolic summary is $\mathsf{Sum}(\mathsf{Pref}) = \big(K, \tau_{\mathsf{sem}}, \mathbf{P}, D_{\mathsf{priv}}, \mathsf{LT}_{\mathsf{pub}}, \mathsf{Rej}\big)$, where $\mathbf{P}$ is the semantic public-anchor tuple and the remaining notation is as in Section 7.3. Thus $\mathsf{Sum}(\mathsf{Pref})$ contains every quantity conditioned upon in Definition 14 but not $\mathsf{LT}_{\mathsf{priv}}$. For a summary S, let $\mathsf{San}(S)$ denote the operational state that retains $K, \mathsf{LT}_{\mathsf{pub}}$, deletes all private-table entries, and sets $\mathsf{Rej} = \emptyset$.

Definition 30 (RSR Consistency with Learned Relations). *Fix real public parameters $\mathsf{pp} = (\mathsf{pp}_{\mathsf{exp}}, P_1, \ldots, P_n)$ and a learned canonical relation set R. Let $S^\star$ be the summary of a candidate setup–key-generation prefix, with retained public parameters $\mathsf{pp}^\star$ and semantic public-anchor tuple $\mathbf{P}^\star$. We say that $S^\star$ is* RSR-consistent *with (pp, R) if: (i) $\mathsf{pp}^\star = \mathsf{pp}$ as canonical structured bitstring objects, including their explicit components, shapes, and designated literal label strings; and (ii) $\mathbf{P}^\star$ realizes every relation in R.*

Remark 13. Literal equality of the public labels is essential. Because setup and key generation may inspect label bits, a candidate public table cannot be renamed after the candidate prefix has been generated.

Fix a candidate summary $S^\star$, an identity id, and the explicit candidate key $\mathsf{sk}_{\mathsf{id}}^\star$ contained in its retained state. For $b \in \{0,1\}$, define

$$\overline{p}_{\mathsf{cor}}^\star(b) = \Pr\Big[\mathsf{Dec}\big(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}^\star, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, b)\big) = b \;\Big|\; \mathsf{San}(S^\star)\Big].$$

Let $\mathsf{Tr}_{\mathsf{id},b}^\star$ be the corresponding equality-saturated semantic transcript. Define

$$\overline{p}_{\mathsf{rel}}^\star(b; R) = \Pr\left[\mathsf{Rel}_{\mathbf{P}^\star}(\mathsf{Tr}_{\mathsf{id},b}^\star) \subseteq \mathrm{cl}_{\mathbf{P}^\star}(R) \,\middle|\, \mathsf{San}(S^\star)\right].$$

Definition 31 (Admissible Sanitized Candidate). *Let $\tau_{\mathsf{cor}}, \tau_{\mathsf{rel}} \in \mathbb{Q} \cap [0,1]$. A candidate summary $S^\star$ for identity id is $(\tau_{\mathsf{cor}}, \tau_{\mathsf{rel}})$-admissible relative to (pp, R) if all of the following hold:*

1. *$S^\star$ occurs with positive probability in the support of the lazy-label setup–key-generation prefix for id.*
2. *$S^\star$ is RSR-consistent with (pp, R) in the sense of Definition 30.*
3. *For every $b \in \{0,1\}$, $\overline{p}^\star_{\mathsf{cor}}(b) \geq \tau_{\mathsf{cor}}$.*
4. *For every $b \in \{0,1\}$, $\overline{p}^\star_{\mathsf{rel}}(b; R) \geq \tau_{\mathsf{rel}}$.*

Crucially, admissibility is determined entirely by $S^\star, (\mathsf{pp}, R)$, and continuations from $\mathsf{San}(S^\star)$, and is therefore independent of any representative private table.

Lemma 17 (Symmetry-Preserving Sanitized-Candidate Search). *There exists a deterministic algorithm*

$$\mathsf{FindSanitizedCandidate}\big(1^\lambda, \mathsf{gpar}_\lambda, L_0, \mathsf{pp}, \mathsf{id}, R, \tau_{\mathsf{cor}}, \tau_{\mathsf{rel}}\big)$$

with the following properties.

1. *It makes no query to the real group-action oracle and uses only unbounded local computation.*
2. *If an admissible sanitized candidate exists, it returns the first such summary in a fixed canonical ordering, together with the explicit candidate key contained in that summary. Otherwise, it returns $\bot$.*
3. *Selection preserves private-label symmetry. Namely, if $S^\star$ is the returned summary and $\mathsf{Pref}^\star$ is sampled from the conditional distribution of prefixes satisfying $\mathsf{Sum}(\mathsf{Pref}^\star) = S^\star$, then $\mathsf{LT}^\star_{\mathsf{priv}}$ is conditionally uniform over all injections $D^\star_{\mathsf{priv}} \hookrightarrow \Sigma_\lambda \setminus \big(\mathrm{im}(\mathsf{LT}^\star_{\mathsf{pub}}) \cup \mathsf{Rej}^\star\big)$.*

Proof. All setup/key-generation random tapes, lazy-label choices, and labels lie in finite effectively enumerable spaces. Enumerate all prefixes, project them to their summaries, aggregate equal summaries, and exhaustively enumerate the sanitized encryption-decryption continuations needed to compute $p^\star_{\mathsf{cor}}$ and $p^\star_{\mathsf{rel}}$ exactly. Discard inconsistent or inadmissible summaries and return the first remaining one. This is purely local and exhaustive, proving Items 1 and 2. For Item 3, the selection predicate depends only on $(K, \tau_{\mathsf{sem}}, \mathbf{P}, D_{\mathsf{priv}}, LT_{\mathsf{pub}}, \mathsf{Rej})$ and never on $\mathsf{LT}_{\mathsf{priv}}$. Private-label symmetry therefore remains valid after conditioning on the selected summary, giving the claimed uniform conditional distribution. This completes the proof of Lemma 17.

Lemma 18 (Sanitized Two-World RSR Transfer). *Let S_0 and S_1 be two retained summaries whose public parameters are literally identical and whose semantic public-anchor tuples $\mathbf{P}_0$ and $\mathbf{P}_1$ both realize a canonical relation set R. Consider two continuations of the same deterministic RSR-GGAM algorithm, with identical ordinary inputs and fixed random tape, starting from $\mathsf{San}(S_0)$ and $\mathsf{San}(S_1)$. Let Tr_0 and Tr_1 be their complete lazy-label transcripts. If $\mathsf{Rel}_{\mathbf{P}_0}(\mathsf{Tr}_0) \subseteq \mathrm{cl}_{\mathbf{P}_0}(R)$ and $\mathsf{Rel}_{\mathbf{P}_1}(\mathsf{Tr}_1) \subseteq \mathrm{cl}_{\mathbf{P}_1}(R)$, then the two continuations can be coupled with identical complete bit-level transcripts.*

Proof. The sanitized states have empty rejection sets. Their public tables assign literally identical strings to corresponding public anchors. Apply Corollary 9 to complete the proof of Lemma 18.

7.5 Impossibility of Restricted Composably Label-Sanitizable RSR-GGAM-IBE

We now lift the restricted attack of Section 5 to the RSR-GGAM using the lazy-label, resetting, and symmetry-preserving machinery above.

Postponing Challenge Encryption. The relation-learning, candidate-search, and validation phases do not inspect the challenge ciphertext. We may therefore postpone challenge encryption.

Lemma 19 (Challenge Postponement). *Consider the fixed-representation RSR-GGAM security experiment. Suppose that the second stage of an adversary can be written as $\mathcal{A}_2 = \mathcal{A}_{\mathsf{post}} \circ \mathcal{A}_{\mathsf{pre}}$, where $\mathcal{A}_{\mathsf{pre}}$ does not take the challenge ciphertext or the challenge bit as input. Then the joint distribution of the experiment is unchanged if the execution order $\Big(C^\star \leftarrow \mathsf{Enc}^{O^{\sigma_X}_\star}(pp, \mathsf{id}^\star, b); \mathcal{A}^{O^{\sigma_X}_\star}_{\mathsf{pre}}\Big)$ is replaced by $\Big(\mathcal{A}^{O^{\sigma_X}_\star}_{\mathsf{pre}}; C^\star \leftarrow \mathsf{Enc}^{O^{\sigma_X}_\star}(pp, \mathsf{id}^\star, b).\Big)$*

Proof. Condition on σ_X and all other randomness. The oracle $\mathcal{O}^{\sigma_X}_\star$ is a fixed stateless function, and the two computations use independent random tapes and do not read or modify each other's state. Hence they commute pointwise; averaging proves the claim. This completes the proof of Lemma 19.

Thus challenge encryption and final candidate-key decryption may be treated as one final composed computation after all challenge-independent pre-computation. For convenience, set ϵ_{lbl} and $\epsilon_{\mathsf{forget}}$ to 1 whenever their denominator conditions fail. Let Q_{cor} bound the action queries in one composed Setup-KeyGen-Enc-Dec execution, let q_{cor} bound the post-prefix Enc-Dec queries, and let $a_{\mathsf{cor}}, r_{\mathsf{cor}}$ bound the private labels and rejected strings in the setup–key-generation prefix. Define

$$\delta_{\mathsf{cor}}(\lambda) = 2\epsilon_{\mathsf{lbl}}\big(1, Q_{\mathsf{cor}}(\lambda)\big) + \epsilon_{\mathsf{forget}}\big(n_{\mathsf{pp}}(\lambda)+1, a_{\mathsf{cor}}(\lambda), r_{\mathsf{cor}}(\lambda), q_{\mathsf{cor}}(\lambda)\big).$$

Let $J_{\mathrm{atk}}(\lambda)$ index the sanitization boundaries of a maximal challenge-independent attack schedule, padding unreached boundaries by zero-query dummy blocks as in Lemma 16. For $j \in J_{\mathrm{atk}}$, let p_j, a_j, r_j, Q_j bound, respectively, the live labels, forgotten labels, rejection-set size, and remaining oracle queries. Set

$$\delta_{\mathsf{reset}}(\lambda) = \sum_{j \in \mathcal{J}_{\mathrm{atk}}(\lambda)} \epsilon_{\mathsf{forget}}(p_j(\lambda), a_j(\lambda), r_j(\lambda), Q_j(\lambda))\,.$$

If Q_{atk} bounds all real-oracle queries of the restricted attack, including designated-label validation, define

$$\delta_{\mathsf{comp}}(\lambda) = 2\epsilon_{\mathsf{lbl}}\big(1, Q_{\mathrm{atk}}(\lambda)\big) + \delta_{\mathsf{reset}}(\lambda).$$

As before, every error term is interpreted using the extended convention that it equals 1 whenever its required denominator condition fails.

The Fixed-Prefix Sanitized Correctness Experiment. Fix (id, b). Sample a uniform fixed representation σ_X, run Setup and KeyGen(id) against it, form the retained summary of the setup–key-generation prefix, sanitize it, and then run Enc-Dec from a fresh lazy-label continuation initialized with the retained public-label table. Let $\epsilon_{\mathsf{san}}(\lambda; \mathsf{id}, b)$ be the resulting error probability. For a fixed-representation prefix Pref, its induced table is $\mathsf{LT}^{\sigma}_{\mathsf{Pref}} = \sigma_X \restriction_{D_{\mathsf{Pref}}}$, where D_{Pref} contains all encountered semantic elements. We also define the transcript-derived rejection set

$$\mathsf{Rej}^{\sigma}_{\mathsf{Pref}} = \{L \in \Sigma_\lambda : \text{an action-oracle query on } L \text{ is rejected during } \mathsf{Pref}\}\,.$$

Except on the coupling failure already charged to ϵ_{lbl}, Corollary 8 ensures that every accepted designated label belongs to this induced table, so no additional provenance loss is needed.

Lemma 20 (Correctness Transfer to the Fixed-Prefix Sanitized Experiment). *For every* $\mathsf{id} \in \mathsf{ID}_\lambda$ *and* $b \in \{0,1\}$, $\epsilon_{\mathsf{san}}(\lambda; \mathsf{id}, b) \le \epsilon_{\mathsf{corr}}(\lambda) + \delta_{\mathsf{cor}}(\lambda)$.

Proof. Starting from ordinary fixed-representation correctness: (i) replace the fixed representation by a shared lazy-label execution, costing at most $\epsilon_{\mathsf{lbl}}(1, Q_{\mathsf{cor}})$ by Lemma 12; (ii) sanitize the setup-key-generation prefix, costing at most $\epsilon_{\mathsf{forget}}((n_{\mathsf{pp}}+1), a_{\mathsf{cor}}, r_{\mathsf{cor}}, q_{\mathsf{cor}})$ by Definition 15 and Lemma 15; and (iii) replace the lazy prefix by the fixed-representation prefix while applying the same sanitizer and continuation, costing another $\epsilon_{\mathsf{lbl}}(1, Q_{\mathsf{cor}})$ by Lemma 12 and data processing. The final experiment is exactly the one above. This completes the proof of Lemma 20.

Conditioning on a Fixed Setup World. A setup world ω fixes the group-action instance, full representation σ_X, setup random tape, and setup output, but no later randomness. Let $\mathrm{err}^{\mathsf{san}}_{\omega}(\mathsf{id}, b)$ be the conditional error obtained by running KeyGen(id) against the fixed representation in ω, sanitizing the resulting prefix, and using fresh lazy-label randomness thereafter. Lemma 20 gives

$$\mathbb{E}_{\omega}[\mathrm{err}^{\mathsf{san}}_{\omega}(\mathsf{id}, b)] \le \epsilon_{\mathsf{corr}}(\lambda) + \delta_{\mathsf{cor}}(\lambda).$$

For later use, let $\gamma_\omega(\mathsf{id})$ be the conditional failure probability of the Lemma 12 coupling between this fixed-representation setup–key-generation prefix and its lazy-label counterpart. Then $\mathbb{E}_\omega[\gamma_\omega(\mathsf{id})] \le \epsilon_{\mathsf{lbl}}\big(1, Q_{\mathsf{cor}}(\lambda)\big)$.

Lemma 21 (Good Sanitized Setup Worlds). *Let* $S \subseteq \mathsf{ID}_\lambda$ *be finite and let* $\alpha > 0$. *Then*

$$\Pr_{\omega}\left[\exists\, \mathsf{id} \in S,\ b \in \{0,1\} : \mathrm{err}^{\mathsf{san}}_{\omega}(\mathsf{id}, b) > \alpha\right] \le \frac{2|S|\big(\epsilon_{\mathsf{corr}}(\lambda) + \delta_{\mathsf{cor}}(\lambda)\big)}{\alpha}.$$

Proof. Applying Markov's inequality separately to every identity-message pair and taking a union bound over the $2|S|$ pairs immediately yields a proof of Lemma 21.

Theorem 9 (Impossibility of Restricted Composably Label-Sanitizable RSR-GGAM-IBE). *Let $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a restricted key-label-oblivious and composably label-sanitizable RSR-GGAM-IBE scheme over an efficiently indexable identity-space family $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda\in\mathbb{N}}$. Suppose that the public parameters contain at most $n_{\mathsf{pp}}(\lambda)$ distinct designated non-origin labels, where n_{pp} is publicly known, efficiently computable, and polynomially bounded. Let $\epsilon_{\mathsf{corr}}(\lambda)$ denote the standard averaged correctness error of Π, and set $k(\lambda) = n_{\mathsf{pp}}(\lambda) + 1$. Then there exists a single pseudoefficient RSR-GGAM adversary $\mathcal{A}$ such that, for every security parameter satisfying $|\mathsf{ID}_\lambda| \geq k(\lambda)$, we have*

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) \geq \frac{1}{40k(\lambda)} - 33k(\lambda)\big(\epsilon_{\mathsf{corr}}(\lambda) + \delta_{\mathsf{cor}}(\lambda)\big) - \delta_{\mathsf{comp}}(\lambda) - e^{-\Omega(\lambda)}.$$

Proof. Fix λ, write $n = n_{\mathsf{pp}}(\lambda)$,$k = (n+1)$, $\eta = 1/(640k)$, and use the adversary of Section 5.5, replacing its candidate search by $\mathsf{FindSanitizedCandidate}$ from Lemma 17 with $\tau_{\mathsf{cor}} = 19/20$ and $\tau_{\mathsf{rel}} = 4/5$. All RSR computations use the instrumented transcript convention of Section 3.3.

We first analyze the *fixed-prefix resetting experiment* $H_{\mathsf{fix\text{-}reset}}$. By Lemma 19, all challenge-independent pre-computation occurs first. Setup and extraction-key prefixes use one fixed representation; after sanitization, every composed encryption–decryption sample begins from the retained public-label table with empty rejection set and fresh lazy-label randomness. Only its ordinary output and canonical semantic relations survive to the next block. Offline candidate continuations similarly begin from their sanitized summaries, and challenge encryption together with final decryption forms the last composed block. Let $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_k\}$ be the identities returned by the public indexer. Define $\mathsf{GoodSetup}$ by $\mathrm{err}^{\mathsf{san}}_\omega(\mathsf{id}, b) \leq \eta$ for $\mathsf{id} \in S, b \in \{0,1\}$ and $\gamma_\omega(\mathsf{id}) \leq 1/(8k)$ for $\mathsf{id} \in S$. Lemma 21, Markov's inequality, and $2\epsilon_{\mathsf{lbl}}(1, Q_{\mathsf{cor}}) \leq \delta_{\mathsf{cor}}$ give

$$\Pr[\neg\mathsf{GoodSetup}] \leq 1284k^2(\epsilon_{\mathsf{corr}} + \delta_{\mathsf{cor}}). \tag{12}$$

Fix $\omega \in \mathsf{GoodSetup}$. Because every reset-state sample starts from the same retained public table with fresh independent randomness, the repeated-sampling analysis of Section C applies verbatim. For completeness, augment the analysis with independent sanitized genuine-key summaries $Z_1, \ldots, Z_k$, where Z_i is obtained by running $\mathsf{KeyGen}(\mathsf{id}_i)$ against the fixed representation in ω and then sanitizing its prefix. These summaries are hidden analysis variables; only their explicit keys are used by the adversary. Since the fixed representation oracle is stateless and all key-generation and continuation tapes are independent, the complete hypothetical sequence $(Z_1, E_1, R_1), \ldots, (Z_k, E_k, R_k)$ is defined independently of the random challenge position Y, and its prefix before Y has exactly the distribution generated in $H_{\mathsf{fix\text{-}reset}}$. Hence the random-prefix argument applies, and since the public-anchor relation rank is at most n,

$$\Pr[\mathsf{GoodPrefix} \mid \omega] \geq \frac{1}{2k}. \tag{13}$$

Condition now on $\mathsf{GoodPrefix}$ and the preceding history, and put $Z := Z_Y$. The conditional distribution of Z remains its setup-wise sanitized key-generation distribution. Exactly as in Claim 3 of Section 5.5, setup-wise correctness gives

$$\Pr[\bar{p}^{\mathsf{cor}}_Z(b) \geq 19/20 \ \forall b] \geq 1 - \frac{1}{16k},$$

while the repeated-sampling calculation with $L = 8k$ gives

$$\Pr[\bar{p}^{\mathsf{rel}}_Z(b; R_{Y-1}) \geq 4/5 \ \forall b] \geq \frac{1}{4k}.$$

Moreover $\omega \in \mathsf{GoodSetup}$ implies that the fixed-to-lazy prefix coupling for id_Y fails with probability at most $1/(8k)$. Therefore

$$\Pr[Z \text{ satisfies both thresholds and the prefix coupling succeeds}]$$

$$\geq \frac{1}{4k} - \frac{1}{16k} - \frac{1}{8k} = \frac{1}{16k} > 0.$$

On this event the fixed and lazy setup–key-generation prefixes have identical retained summaries; hence Z occurs in the support required by Definition 31, has literally identical public parameters, realizes R_{Y-1}, and meets both admissibility thresholds. Lemma 17 therefore ensures that the offline search returns an admissible summary. Let $\mathsf{sk}^\star$ be its explicit candidate key. Candidate-world correctness is at least $19/20$, while candidate-world and real-world relation containment are each at least $4/5$. Both continuations start from sanitized states with identical public labels and empty rejection sets, so Lemma 18 couples their complete bit-level transcripts whenever both relation-containment events occur. Their joint probability is at least $4/5 + 4/5 - 1 = 3/5$. Consequently, for either message bit,

$$\Pr[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}^\star, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_Y, b)) = b \mid \mathsf{GoodPrefix}, \omega] \geq \frac{11}{20}. \tag{14}$$

All validation trials and the final challenge computation are independent samples from this same reset-state distribution. Thus the validation analysis of Section 5.5 applies unchanged. Using (13)–(14),

$$\Pr[\mathcal{A} \text{ wins in } H_{\mathsf{fix\text{-}reset}} \mid \mathsf{GoodSetup}] \geq \frac{1}{2} + \frac{1}{40k} - e^{-\Omega(\lambda)}.$$

Outside $\mathsf{GoodSetup}$, the validation safeguard gives

$$\Pr[\mathcal{A} \text{ wins in } H_{\mathsf{fix\text{-}reset}} \mid \neg\mathsf{GoodSetup}] \geq \frac{1}{2} - e^{-\Omega(\lambda)}.$$

Combining these inequalities with (12),

$$\mathsf{Adv}^{\mathsf{sel}}_{H_{\mathsf{fix\text{-}reset}},\mathcal{A}}(\lambda) \geq \frac{1}{40k} - 33k(\epsilon_{\mathsf{corr}} + \delta_{\mathsf{cor}}) - e^{-\Omega(\lambda)}. \tag{15}$$

It remains to return to the ordinary RSR experiment. Let $H_{\mathsf{LL\text{-}share}}$ use one shared lazy-label state and let $H_{\mathsf{LL\text{-}reset}}$ sanitize at all boundaries in J_{atk}. Lemma 12, Lemma 16, and a final fixed-prefix/lazy-prefix replacement give, respectively,

$$\begin{aligned}
&\Delta(\mathsf{View}_{\mathsf{RSR}}, \mathsf{View}_{H_{\mathsf{LL\text{-}share}}}) \leq \epsilon_{\mathsf{lbl}}(1, Q_{\mathrm{atk}}),\\
&\Delta(\mathsf{View}_{H_{\mathsf{LL\text{-}share}}}, \mathsf{View}_{H_{\mathsf{LL\text{-}reset}}}) \leq \delta_{\mathsf{reset}},\\
&\Delta(\mathsf{View}_{H_{\mathsf{LL\text{-}reset}}}, \mathsf{View}_{H_{\mathsf{fix\text{-}reset}}}) \leq \epsilon_{\mathsf{lbl}}(1, Q_{\mathrm{atk}}).
\end{aligned}$$

The last bound follows by coupling only the setup/extraction prefix and then applying the same sanitizer and fresh continuations; challenge postponement is exact. Therefore $\Delta(\mathsf{View}_{\mathsf{RSR}}, \mathsf{View}_{H_{\mathsf{fix\text{-}reset}}}) \leq \delta_{\mathsf{comp}}$. Combining with (15) proves the stated advantage bound. The offline search is purely local, while all remaining phases invoke only polynomially many polynomial-query scheme computations. Hence $\mathcal{A}$ is pseudoefficient; the indexer fallback makes it one uniform adversary defined at every parameter. This completes the proof of Theorem 9.

Corollary 12 (No Large-Identity-Space Restricted Composably Label-Sanitizable RSR-GGAM-IBE). *Assume that the random-string representation satisfies the standard label-slack condition. Let Π be a fully selective-secure restricted key-label-oblivious and composably label-sanitizable RSR-GGAM-IBE scheme over an efficiently indexable identity-space family $\mathsf{ID} = \{\mathsf{ID}_\lambda\}_{\lambda\in\mathbb{N}}$. Suppose that Π has negligible standard correctness error and polynomially bounded action-query complexity. Then $|\mathsf{ID}_\lambda| \leq n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ. In particular, no such scheme with polynomial-size public parameters can support a super-polynomial efficiently indexable identity space.*

Proof. Under label slack and the polynomial bounds, $\delta_{\mathsf{cor}}(\lambda) = \mathrm{negl}(\lambda)$. Moreover, every term $\epsilon_{\mathsf{forget}}(p, a, r, q)$ occurring in δ_{reset} is negligible: apply label slack with $p' = (p + a + r)$ and $q' = q(a + r + 1)$, which dominates

$$\epsilon_{\mathsf{forget}}(p, a, r, q) = \min\left\{1,\ \frac{q(a+r)}{2^{\ell(\lambda)} - p - a - r - q}\right\}.$$

There are only polynomially many reset boundaries, so $\delta_{\mathsf{reset}} = \mathrm{negl}(\lambda)$; Lemma 12 similarly gives $\epsilon_{\mathsf{lbl}}(1, Q_{\mathrm{atk}}) = \mathrm{negl}(\lambda)$, and therefore $\delta_{\mathsf{comp}}(\lambda) = \mathrm{negl}(\lambda)$. Since $k = (n_{\mathsf{pp}} + 1)$ is polynomial, Theorem 9 gives

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}}(\lambda) \geq \frac{1}{40k(\lambda)} - \mathrm{negl}(\lambda)$$

whenever $|\mathsf{ID}_\lambda| \geq k(\lambda)$. Such parameters cannot occur infinitely often under full selective security. This completes the proof of Corollary 12.

7.6 Tightness of the Public-Label Bound

We show that the preceding bound is tight even against arbitrary representation-sensitive RSR-GGAM adversaries. The construction is the random-label analogue of Section 6.3.

The Construction. Let $\mathsf{GA}_\lambda = (G_\lambda, X_\lambda, \star_\lambda, \widetilde{x}_\lambda)$ be a finite abelian regular group action, and write $N_\lambda = |G_\lambda| = |X_\lambda|$ and $\Sigma_\lambda = \{0,1\}^{\ell(\lambda)}$. Let $n = n(\lambda)$ be polynomially bounded and set $\mathsf{ID}_\lambda = [n]$. Given a random injection $\sigma_X : X_\lambda \hookrightarrow \Sigma_\lambda$, write $L_0 = [\widetilde{x}_\lambda]_{\sigma_X}$. We define $\Pi^{\mathsf{RSR}}_{\mathsf{match}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ as follows (subscripts λ are omitted for brevity).

$\mathsf{Setup}^{\mathcal{O}^{\sigma_X}_\star}(1^\lambda, \mathsf{gpar}_\lambda, L_0)$**:** Sample independently $a_1, \ldots, a_n \leftarrow G$, and compute $L_i \leftarrow \mathcal{O}^{\sigma_X}_\star(a_i, L_0) = [a_i \star \widetilde{x}]_{\sigma_X}$ for each $i \in [n]$. Output $\mathsf{msk} = (a_1, \ldots, a_n)$ and $\mathsf{pp} = (n; L_1, \ldots, L_n)$. As in Section 4.2, repeated labels and occurrences of L_0 are represented by the canonical explicit shape, so pp contains at most n distinct non-origin designated labels.

$\mathsf{KeyGen}^{\mathcal{O}^{\sigma_X}_\star}(\mathsf{pp}, \mathsf{msk}, i)$**:** Output the explicit group element $\mathsf{sk}_i = a_i$.

$\mathsf{Enc}^{\mathcal{O}^{\sigma_X}_\star}(\mathsf{pp}, i, b)$**:** Sample $r \leftarrow G$ and compute $R \leftarrow \mathcal{O}^{\sigma_X}_\star(r, L_0)$. If $b = 0$, set $Z \leftarrow \mathcal{O}^{\sigma_X}_\star(r, L_i)$. If $b = 1$, sample $s \leftarrow G$ independently and set $Z \leftarrow \mathcal{O}^{\sigma_X}_\star(s, L_0)$. Output $C = (R, Z)$.

$\mathsf{Dec}^{\mathcal{O}^{\sigma_X}_\star}(\mathsf{pp}, \mathsf{sk}_i, C)$**:** Parse $\mathsf{sk}_i = a_i$ and $C = (R, Z)$, compute $D \leftarrow \mathcal{O}^{\sigma_X}_\star(a_i, R)$, and output 0 if $D = Z$, and 1 otherwise.

Theorem 10 (Matching n-Identity Construction in the Sanitizable RSR-GGAM). *Let $\Pi^{\mathsf{RSR}}_{\mathsf{match}}$ be the scheme above. Then:*

1. *It supports the identity space $\mathsf{ID}_\lambda = [n(\lambda)]$ using at most $n(\lambda)$ distinct non-origin designated public labels.*
2. *It is restricted key-label-oblivious and composably label-sanitizable.*
3. *Its pointwise correctness error is at most $1/N_\lambda$.*
4. *For every fully selective RSR-GGAM adversary $\mathcal{A}$ making at most $q = q(\lambda)$ action-oracle queries across its two stages, let $Q = (q + 2n + 4)$. Whenever $|\Sigma_\lambda| > Q + 1$,*

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi^{\mathsf{RSR}}_{\mathsf{match}},\mathcal{A}}(\lambda) \leq \frac{2\binom{q+n+3}{2}}{N_\lambda} + 2\epsilon_{\mathsf{lbl}}(1, Q).$$

Consequently, if N_λ^{-1} is negligible and the representation family satisfies the standard label-slack condition, then $\Pi^{\mathsf{RSR}}_{\mathsf{match}}$ is fully selective-secure against pseudoefficient RSR-GGAM adversaries.

Proof. Item 1 is immediate from setup. For Item 2, every secret key is the explicit group element a_i, and KeyGen neither outputs nor inspects labels. Moreover, the scheme algorithms treat every label opaquely: labels are only copied, supplied to the action oracle, or compared for equality. Hence setup-key-generation has private-label symmetry, and at every later sanitization boundary a permutation of transient labels fixing the retained labels and rejection set induces a probability-preserving permutation of executions. This remains true after arbitrary earlier sanitizations, proving composable label sanitizability. Equivalently, the construction is globally label-opaque; Section 7.8 formalizes this criterion.

For correctness (Item 3), on $b = 0$, abelianness gives

$$\mathcal{O}^{\sigma_X}_\star(a_i, R) = [(a_i \circ r) \star \widetilde{x}]_{\sigma_X} = [(r \circ a_i) \star \widetilde{x}]_{\sigma_X} = Z.$$

For $b = 1$, decryption outputs 0 exactly when $s = a_i \circ r$, which occurs with probability $1/N_\lambda$.

It remains to prove security (Item 4). Let $\mathsf{View}^{\mathsf{RSR}}_b$ and $\mathsf{View}^{\mathsf{LL}}_b$ be the complete fixed- and lazy-representation views for challenge bit b. Including designated-output validation, the experiment makes at most $Q = q + 2n + 4$ action queries, so Lemma 12 gives

$$\Delta(\mathsf{View}^{\mathsf{RSR}}_b, \mathsf{View}^{\mathsf{LL}}_b) \leq \epsilon_{\mathsf{lbl}}(1, Q) \qquad (b \in \{0,1\}).$$

We compare the two lazy-label experiments by a label-synchronous planted-parallelogram coupling. Condition on the adversary's common random tape, first-stage transcript, and all non-challenge setup exponents. Writing $a = a_{i^\star}$, the challenge-dependent semantic elements are

$$L_{i^\star} = a \star \widetilde{x}, \qquad R = r \star \widetilde{x}, \qquad Z = \begin{cases} (r \circ a) \star \widetilde{x}, & b = 0, \\ s \star \widetilde{x}, & b = 1. \end{cases}$$

All remaining semantic elements and explicit side information are independent of a, r, s. Resolve all challenge-independent equalities identically in both experiments, and assign the same literal labels to corresponding challenge-dependent formal terms. Until a non-forced equality involving a challenge-dependent term occurs, the two executions therefore have identical complete bit-level transcripts, even though the adversary may inspect labels arbitrarily. For any fixed non-forced pair, conditioning on the background and all but one of a, r, s makes the equality determine the remaining uniform group element; its probability is therefore at most $1/N_\lambda$. Designated-output validation need not be counted in this collision transcript: it applies the identity action to an already exposed valid label and creates no new semantic relation. Thus there are at most $T = (q + n + 3)$ relevant input or oracle-response occurrences. A union bound over pairs in both challenge experiments yields

$$\Delta(\mathsf{View}_0^{\mathsf{LL}}, \mathsf{View}_1^{\mathsf{LL}}) \le \frac{2\binom{T}{2}}{N_\lambda}.$$

The triangle inequality with the two lazy-label replacements proves the stated advantage bound. This completes the proof of Theorem 10.

7.7 A Tight Impossibility in the Presence of Twists

We finally consider the stronger GGAM interface with a twist operation [DHK+23].

Twist-Augmented Sanitizable RSR-GGAM. Relative to the distinguished origin $\widetilde{x}$, define

$$\mathsf{tw}\big(a \star \widetilde{x}\big) = a^{-1} \star \widetilde{x}.$$

Regularity makes this map well defined. Moreover, tw is an involution and, by commutativity of the acting group, satisfies $\mathsf{tw}(g \star x) = g^{-1} \star \mathsf{tw}(x)$. The twist-augmented RSR-GGAM additionally provides the oracle

$$\mathcal{O}_{\mathsf{tw}}^{\sigma_X}(L) = \begin{cases} \sigma_X\big(\mathsf{tw}\big(\sigma_X^{-1}(L)\big)\big), & L \in \mathrm{im}(\sigma_X), \\ \bot, & \text{otherwise.} \end{cases}$$

All scheme algorithms and adversaries may query both oracles, and query complexity counts both kinds of query. The lazy-label implementation uses the same $\mathsf{LT}, \mathsf{Rej}$: an invalid input is rejected and recorded, while a valid twist response reuses or lazily allocates the label of the resulting semantic element. Thus each twist query exposes at most one new semantic element and allocates at most one new label.

Oriented Anchored Terms. Let $P_0 = \widetilde{x}$ and $P_i = p_i \star \widetilde{x}$ for $i \in [n]$ be the semantic public-anchor tuple, where $p_0 = e$. For each $i \in [n]$, we introduce two oriented anchors $P_i^+ = p_i \star \widetilde{x}$ and $P_i^- = p_i^{-1} \star \widetilde{x}$, and identify $P_0^+ = P_0^- = \widetilde{x}$. Every set element encountered in a computation using actions and twists has an oriented anchored term $(i, s, g) \in \{0, \dots, n\} \times \{+1, -1\} \times G$ with semantics

$$[\![(i, s, g)]\!]_{\mathbf{P}} = \big(g \circ p_i^s\big) \star \widetilde{x}.$$

Indeed, action and twist queries induce the syntactic updates

$$(i, s, g) \xmapsto{a\star(\cdot)} (i, s, a \circ g), \qquad (i, s, g) \xmapsto{\mathsf{tw}} (i, -s, g^{-1}).$$

Hence twists reverse anchor orientation but never combine two independent anchors.

The Odd-Order Refinement. The preceding doubling of anchors is unnecessary when the squaring map $\mathsf{sq}\colon G \longrightarrow G$, specified as $\mathsf{sq}(g) = g^2$, is bijective. For a finite abelian group, this is equivalent to the absence of nontrivial 2-torsion; in particular, it holds for every cyclic group of odd order N. An equality between the oriented terms (i, s, g) and (j, t, h) reveals the signed multiplicative relation

$$p_i^s \circ p_j^{-t} = g^{-1} \circ h.$$

Oriented and Signed Relation Systems. Let $\mathcal{M}_{\mathbf{P}}^{\mathsf{or}}$ denote the graphic relation matroid on the oriented anchors. When sq is bijective, let $\mathcal{M}_{\mathbf{P}}^{\pm}$ be the rooted signed-frame matroid whose canonical signed relations are

$$e_{i,j}^{s,t} = (i, j, s, t, \delta_{i,j}^{s,t}), \qquad \delta_{i,j}^{s,t} = p_i^s \circ p_j^{-t}.$$

We write $\mathrm{Rel}_{\mathbf{P}}^{\mathsf{or}}(\mathrm{Tr})$ and $\mathrm{Rel}_{\mathbf{P}}^{\pm}(\mathrm{Tr})$ for the relations revealed by an equality-saturated twist-augmented transcript.

Lemma 22 (Twisted Anchored Transcripts). *Let $\mathcal{B}$ be a deterministic generic-action algorithm with access to $\mathcal{O}_\star$ and $\mathcal{O}_{\mathsf{tw}}$, and let* Tr *be its equality-saturated transcript relative to a public-anchor tuple $\mathbf{P} = (P_0, P_1, \ldots, P_n)$, where $P_0 = \widetilde{x}$. Then the following statements hold.*

1. *Every valid input or oracle-response occurrence has a syntactically determined oriented anchored term (i, s, g) with semantics $[\![(i, s, g)]\!]_{\mathbf{P}} = (g \circ p_i^s) \star \widetilde{x}$.*
2. *In general,* $\mathrm{rank}_{\mathbf{P}}^{\mathsf{or}}\big(\mathrm{Rel}_{\mathbf{P}}^{\mathsf{or}}(\mathrm{Tr})\big) \le 2n$.
3. *If $\mathsf{sq}\colon G \to G$ is bijective, then* $\mathrm{rank}_{\mathbf{P}}^{\pm}\big(\mathrm{Rel}_{\mathbf{P}}^{\pm}(\mathrm{Tr})\big) \le n$.
4. *In either relation system, its corresponding closure characterizes all positive equalities implied by the observed relations.*
5. *Let $\mathbf{P}$ and $\mathbf{P}'$ be two anchor tuples that realize a common relation set R. If, in both executions, every positive transcript relation belongs to the appropriate closure of R, then the executions have identical equality and oracle-input/output structures up to an oriented-handle renaming. In the lazy-label model, synchronous allocation strengthens this to equality of the complete bit-level transcripts.*

Proof. Item 1 follows inductively from

$$(i, s, g) \mapsto (i, s, a \circ g), \qquad (i, s, g) \mapsto (i, -s, g^{-1}),$$

so neither oracle combines anchors. For Item 2, regard the two orientations of every non-origin anchor as distinct virtual anchors and apply the ordinary gain-graph argument of Lemma 1, giving rank at most $2n$. Assume now that sq is bijective. In a connected signed component, tree relations express every anchor coordinate in terms of one coordinate or its inverse. A balanced unrooted component leaves one free coordinate; an unbalanced cycle yields an equation $p_i^2 = c$, which uniquely determines p_i; and a component containing the rooted coordinate $p_0 = e$ is already pinned. Thus the rooted signed-frame rank is at most n, and the same elimination argument identifies its closure.

Finally, couple two executions as in Lemma 4. Corresponding action queries apply the same explicit shift, while corresponding twists flip the same orientation and invert the same shift. A first divergence could therefore only arise from an equality holding in one execution but not the other, which the two-sided closure hypothesis excludes. Label-synchronous allocation then gives identical literal transcripts as in Lemma 14. This completes the proof of Lemma 22.

Theorem 11 (RSR-GGAM-IBE Impossibility with Twists). *Let Π be a restricted key-label-oblivious and composably label-sanitizable twist-augmented RSR-GGAM-IBE scheme over a family of finite abelian regular group actions, satisfying the remaining hypotheses of Theorem 9. Let $n = n_{\mathsf{pp}}(\lambda)$. The following two statements hold.*

1. *In general, set $r_{\mathsf{tw}}(\lambda) = 2n_{\mathsf{pp}}(\lambda)$, and $k_{\mathsf{tw}}(\lambda) = 2n_{\mathsf{pp}}(\lambda) + 1$. There exists a single pseudoefficient fully selective adversary $\mathcal{A}_{\mathsf{tw}}^{\mathsf{gen}}$ for which the bound below holds whenever $|\mathsf{ID}_\lambda| \ge k_{\mathsf{tw}}(\lambda)$.*
2. *Suppose that $\mathsf{sq}\colon G_\lambda \to G_\lambda$ is bijective for all sufficiently large λ. Set $r_{\mathsf{tw}}(\lambda) = n_{\mathsf{pp}}(\lambda)$ and $k_{\mathsf{tw}}(\lambda) = (n_{\mathsf{pp}}(\lambda) + 1)$. There exists a single pseudoefficient fully selective adversary $\mathcal{A}_{\mathsf{tw}}^{\mathsf{odd}}$ for which the bound below holds at every sufficiently large λ satisfying $|\mathsf{ID}_\lambda| \ge k_{\mathsf{tw}}(\lambda)$.*

In either case, writing $\mathcal{A}_{\mathsf{tw}}$ for the corresponding adversary,

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi,\mathcal{A}_{\mathsf{tw}}}(\lambda) \geq \frac{1}{40k_{\mathsf{tw}}(\lambda)} - 33k_{\mathsf{tw}}(\lambda)\big(\epsilon_{\mathsf{corr}}(\lambda) + \delta^{\mathsf{tw}}_{\mathsf{cor}}(\lambda)\big) - \delta^{\mathsf{tw}}_{\mathsf{comp}}(\lambda) - e^{-\Omega(\lambda)}.$$

Here $\delta^{\mathsf{tw}}_{\mathsf{cor}}$ and $\delta^{\mathsf{tw}}_{\mathsf{comp}}$ are defined by the same expressions as δ_{cor} and δ_{comp} in Theorem 9, with all query bounds counting both action and twist queries.

Proof. Run the attack of Theorem 9 using the relation system from Lemma 22. In the general case its rank is at most $2n$; under bijective squaring it is at most n. Thus the repeated-sampling and random-prefix arguments apply with rank r_{tw}, and Sections 7.4–7.5 carry over after replacing $\mathrm{Rel}_{\mathbf{P}}, \mathrm{cl}_{\mathbf{P}}$ by their oriented or signed counterparts. Candidate inverse orientations are determined by the public anchors, so consistency, admissibility, and offline candidate search remain summary-measurable. At the representation layer, each twist query exposes at most one semantic element and one label. Hence lazy-label equivalence and private-label forgetting remain valid when query counts include both oracle types. Label-synchronous transfer couples corresponding twist responses with one literal label, and the twist oracle is equivariant under label permutations; therefore resetting and symmetry-preserving candidate selection are unchanged. Challenge postponement also applies because both fixed representation oracles are stateless. Substituting the rank bound into the proof of Theorem 9 gives the claim. This completes the proof of Theorem 11.

Corollary 13 (Identity Bound in the Presence of Twists). *Under the standard label-slack condition and the asymptotic hypotheses of Corollary 12, every fully selective-secure restricted key-label-oblivious and composably label-sanitizable twist-augmented RSR-GGAM-IBE scheme satisfies $|\mathsf{ID}_\lambda| \leq 2n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ. If, additionally, $\mathsf{sq}\colon G_\lambda \to G_\lambda$ is bijective for all sufficiently large λ, then $|\mathsf{ID}_\lambda| \leq n_{\mathsf{pp}}(\lambda)$ for all sufficiently large λ.*

Proof. The label-slack and polynomial-boundedness assumptions make $\delta^{\mathsf{tw}}_{\mathsf{cor}}$ and $\delta^{\mathsf{tw}}_{\mathsf{comp}}$ negligible exactly as in Corollary 12. Any infinite violation of the corresponding identity bound would therefore give the adversary of Theorem 11 inverse-polynomial advantage infinitely often. This completes the proof of Corollary 13.

Tightness for Odd-Order Actions. The matching construction remains secure against adversaries with access to the twist oracle when squaring is bijective.

Corollary 14 (Matching Construction with Twists). *Suppose that $\mathsf{sq}\colon G_\lambda \to G_\lambda$ is bijective for all sufficiently large λ. The matching construction of Section 7.6 is fully selective-secure in the twist-augmented RSR-GGAM under the same asymptotic assumptions as Theorem 10. More precisely, every fully selective adversary making at most q action and twist queries satisfies*

$$\mathsf{Adv}^{\mathsf{sel}}_{\Pi^{\mathsf{RSR}}_{\mathsf{match}},\mathcal{A}}(\lambda) \leq \frac{2\binom{q+n+3}{2}}{N_\lambda} + 2\epsilon_{\mathsf{lbl}}(1, q+2n+4).$$

Proof. Extend the coupling of Theorem 10 using the oriented updates

$$(i,s,g) \mapsto (i,s,a\circ g), \qquad (i,s,g) \mapsto (i,-s,g^{-1}).$$

After resolving all challenge-independent background equalities identically, the experiments have the same literal transcript until a non-forced equality involving a challenge-dependent oriented term occurs. Conditioning on the background and all but one hidden uniform group element z, such an equality has the form $z^\alpha = c$ for $\alpha \in \{\pm 1, \pm 2\}$. The corresponding power maps are bijective for $\alpha = \pm 1$, and also for $\alpha = \pm 2$ by bijectivity of squaring, so each equality has probability at most $1/N_\lambda$. As in Theorem 10, the collision transcript has at most $(q+n+3)$ occurrences and the fixed-to-lazy comparison makes at most $(q+2n+4)$ queries, yielding the displayed bound. This completes the proof of Corollary 14.

7.8 A Syntactic Criterion for Label Sanitizability

We give a syntactic sufficient condition for the composable label sanitizability of Definition 16. Informally, any label erased at a sanitization boundary must have been treated opaquely during its lifetime: it may be copied, compared with valid labels, or supplied to the action (and, when present, twist) oracle, but its literal bits must not affect surviving computation.

Erasure-Safe Boundary Normal Form. Fix a label-erasing composition $\mathcal{C}$ and an execution prefix ρ ending immediately before one of its sanitization boundaries. Let LT be the label table at this point, and write $\mathrm{dom}(\mathsf{LT}) = D_{\mathsf{live}} \dot\cup D_{\mathsf{priv}}$, $\mathsf{LT}_{\mathsf{live}} = \mathsf{LT}|_{D_{\mathsf{live}}}$, and $\mathsf{LT}_{\mathsf{priv}} = \mathsf{LT}|_{D_{\mathsf{priv}}}$, as in Definition 16. Labels of elements in D_{priv} are *boundary-private*; their relevant lifetime extends from first appearance to the boundary.

We use an auxiliary provenance typing: a string received in a designated label position or returned by a successful action/twist query is *label-typed*, and copying preserves this type. Ordinary strings remain untyped even if their value happens to equal a valid label. Casting a label-typed value to ordinary data counts as inspecting its representation. The permitted *opaque operations* on a label-typed value are: (i) storing, copying, or forwarding it in a label-typed position; (ii) testing it for equality with another label-typed valid label; (iii) supplying it, from label-typed storage, as the label operand of a group-action query (or, when the twist interface is present, of a twist query); (iv) receiving or returning it in a designated label position; and (v) recording it in an instrumented transcript solely for the extraction of canonical semantic relations, provided that the extraction depends only on semantic annotations and equality patterns and not on literal label bits. The equality outcomes and other renaming-invariant outcomes of these operations may be used in ordinary computation and may affect control flow.

Definition 32 (Erasure-Safe Boundary Normal Form). *The execution prefix ρ is in* erasure-safe boundary normal form *at the given boundary if, throughout the relevant lifetime of each such label, every boundary-private label is used only through the opaque operations listed above.*

Equivalently, the literal representation of a boundary-private label, such as its bits, substrings, hash, ordering position, etc., cannot influence ordinary state, explicit group operands, query scheduling, retained labels, output shape, the rejection set, or any other surviving non-label output. In particular, successful access to a private element must use a label-typed operand; accidental equality with an untyped string is not a permitted use. Dependence on renaming-invariant semantic information is unrestricted.

Definition 33 (Syntactic Label Sanitizability). *An RSR-GGAM-IBE scheme is* syntactically label-sanitizable *if every reachable execution of every label-erasing composition covered by Definition 16 is in erasure-safe boundary normal form at every sanitization boundary, including boundaries reached after any number of earlier sanitizations.*

Boundary Renaming and Conditional Symmetry. We next show that erasure-safe boundary normal form implies the conditional symmetry required by composable label sanitizability. Fix a label-erasing composition $\mathcal{C}$, a sanitization boundary, and a retained summary $S = (K, \tau_{\mathsf{sem}}, \mathsf{LT}_{\mathsf{live}}, \mathsf{Rej}, D_{\mathsf{priv}})$ having positive probability. Define the set of labels fixed by the summary as $F(S) = \mathrm{im}(\mathsf{LT}_{\mathsf{live}}) \cup \mathsf{Rej}$, and let $\mathsf{Adm}(S) = \mathrm{Inj}\big(D_{\mathsf{priv}}, \Sigma_\lambda \setminus F(S)\big)$ be the set of private label tables compatible with S.

Lemma 23 (Boundary-Renaming Lemma). *Suppose that every execution prefix of $\mathcal{C}$ compatible with S is in erasure-safe boundary normal form at the boundary under consideration. Then, for every $\theta_0, \theta_1 \in \mathsf{Adm}(S)$, there is a probability-preserving bijection between executions reaching $(S, \mathsf{LT}_{\mathsf{priv}} = \theta_0)$ and executions reaching $(S, \mathsf{LT}_{\mathsf{priv}} = \theta_1)$.*

Proof. Since θ_0, θ_1 are injections on the same finite domain, the map $\theta_0(x) \longmapsto \theta_1(x)$ for $x \in D_{\mathsf{priv}}$ extends to a permutation π of Σ_λ fixing $F(S)$ pointwise. Given an execution in the first fiber, apply π exactly to label-typed occurrences representing private elements; leave semantic values, ordinary strings and randomness, live labels, and rejected strings unchanged.

This preserves validity of the execution. Rejected untyped strings lie in Rej and are fixed by π; erasure-safe normal form excludes successful untyped access to private elements and comparisons of private label-typed values with ordinary strings. Copying and equality commute with the permutation. For a well-typed action or twist query, the explicit group operand and semantic input/output are unchanged, while private labels are renamed consistently. Canonical relation extraction is unchanged because it depends only on semantic information and equality patterns. Hence control flow, ordinary retained state, semantic transcript, live table, rejection set, and query schedule remain fixed.

Finally, at every lazy allocation, π bijectively maps $\Sigma_\lambda \setminus (\mathrm{im}(LT) \cup \mathsf{Rej})$ in the original execution to the corresponding available set in the renamed execution. Thus corresponding allocation outcomes have equal probability; all other randomness is unchanged. Applying π^{-1} gives the inverse map. This completes the proof of Lemma 23.

Corollary 15 (Conditional Private-Table Symmetry). *Under the hypotheses of Lemma 23,* $\Pr[\mathsf{LT}_{\mathsf{priv}} = \theta \mid S] = 1/|\mathsf{Adm}(S)|$ *for every* $\theta \in \mathsf{Adm}(S)$.

Proof. Lemma 23 shows that all fibers indexed by $\theta \in \mathsf{Adm}(S)$ have equal probability. These fibers are disjoint and exhaust the executions compatible with S, so conditioning on S makes $\mathsf{LT}_{\mathsf{priv}}$ uniform over $\mathsf{Adm}(S)$. This completes the proof of Corollary 15.

Theorem 12 (Syntactic Sanitizability Criterion). *Every syntactically label-sanitizable RSR-GGAM-IBE scheme, possibly with the twist interface of Section 7.7, is composably label-sanitizable in the sense of Definition 16.*

Proof. Fix any label-erasing composition, reachable sanitization boundary, and positive-probability retained summary S. By Definition 33, every compatible prefix is in erasure-safe boundary normal form, so Corollary 15 gives exactly the conditional uniformity of the private label table required by Definition 16. Because the syntactic condition quantifies over boundaries reached after arbitrary earlier sanitizations, the property is composable. This completes the proof of Theorem 12.

Applications and Scope. A particularly simple way to enforce Definition 32 is to treat *every* label opaquely, independently of whether it will eventually be retained.

Definition 34 (Global Label Opacity). *An RSR-GGAM algorithm is* globally label-opaque *if every label it receives or creates remains in label-typed storage and is used only through the opaque operations listed in Section 7.8. An RSR-GGAM-IBE scheme is globally label-opaque if each of its scheme-side algorithms obeys this instruction characterization on every well-formed input and under every sequence of action-oracle (and, when applicable, twist-oracle) responses.*

Proposition 1. *Every globally label-opaque RSR-GGAM-IBE scheme is syntactically label-sanitizable and, consequently, composably label-sanitizable.*

Proof. At every sanitization boundary, global opacity ensures that all boundary-private labels have been used only through the permitted opaque operations. The label-erasing controller likewise uses transient labels only through forwarding, equality, and canonical relation extraction, and designated-output validation uses labels only opaquely. Hence every reachable prefix is in erasure-safe boundary normal form. Applying Theorem 12 completes the proof of Proposition 1.

Opaque Embeddings of Type-Safe Schemes. In particular, the canonical random-label embedding of a TS-GGAM scheme is globally label-opaque: typed handles are represented by random labels but the scheme algorithms retain only the typed-handle interface. Thus such embeddings satisfy composable label sanitizability by Proposition 1.

The Matching Construction. The matching construction of Section 7.6 is also globally label-opaque: setup only publishes designated labels, encryption only copies labels and uses them as oracle operands, and decryption only performs an action query and equality test. Neither scheme execution nor designated-output validation inspects literal label bits.

Corollary 16. *The matching n-identity RSR-GGAM-IBE scheme is syntactically label-sanitizable, and hence, composably label-sanitizable.*

Its key-label-obliviousness condition also holds because KeyGen outputs only the explicit group element a_i and neither outputs nor inspects a label-typed value.

A Minimal Counterexample. Finally, the criterion is sufficient but not necessary. For example, if an algorithm stores a bit of a private label in a surviving explicit output and then erases the label, conditioning on that output restricts the erased label's distribution and violates the required symmetry. Conversely, a computation could inspect a label and erase every trace of the inspection while still satisfying the semantic definition. The syntactic condition is therefore an auditable sufficient criterion rather than a characterization.

References

ABB10. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Berlin, Heidelberg, May / June 2010.

ABP26. Shweta Agrawal, Andrea Basso, and Sikhar Patranabis. Identity-based encryption from isogenies. Cryptology ePrint Archive, Paper 2026/1457, 2026.

ABR26. Marius A. Aardal, Andrea Basso, and Doreen Riepel. The algebraic isogeny model: A general model with applications to SQIsign and key exchanges. In Joan Daemen and Emmanuel Thomé, editors, *EUROCRYPT 2026, Part IV*, volume 16545 of *LNCS*, pages 604–635. Springer, Cham, May 2026.

ADMP20. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Cham, December 2020.

AEK+22. Michel Abdalla, Thorsten Eisenhofer, Eike Kiltz, Sabrina Kunzweiler, and Doreen Riepel. Password-authenticated key exchange from group actions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 699–728. Springer, Cham, August 2022.

AFV11. Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Berlin, Heidelberg, December 2011.

AMPS21. Navid Alamati, Hart Montgomery, Sikhar Patranabis, and Pratik Sarkar. Two-round adaptively secure MPC from isogenies, LPN, or CDH. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part II*, volume 13091 of *LNCS*, pages 305–334. Springer, Cham, December 2021.

AY20. Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 13–43. Springer, Cham, May 2020.

BB04a. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, Berlin, Heidelberg, May 2004.

BB04b. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Berlin, Heidelberg, August 2004.

BCE+25. Giacomo Borin, Maria Corte-Real Santos, Jonathan Komada Eriksen, Riccardo Invernizzi, Marzio Mula, Sina Schaeffler, and Frederik Vercauteren. Qlapoti: Simple and efficient translation of quaternion ideals to isogenies. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 174–205. Springer, Singapore, December 2025.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Berlin, Heidelberg, August 2001.

BGW05. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, Berlin, Heidelberg, August 2005.

BGZ23. Dan Boneh, Jiaxin Guan, and Mark Zhandry. A lower bound on the length of signatures based on group actions and generic isogenies. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 507–531. Springer, Cham, April 2023.

BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 464–492. Springer, Cham, December 2020.

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Cham, December 2019.

BLSV18. Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 535–564. Springer, Cham, April / May 2018.

BPR+08. Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *49th FOCS*, pages 283–292. IEEE Computer Society Press, October 2008.

BSW06. Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 573–592. Springer, Berlin, Heidelberg, May / June 2006.

BY91. Gilles Brassard and Moti Yung. One-way group actions. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 94–107. Springer, Berlin, Heidelberg, August 1991.

CDG+17. Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 33–65. Springer, Cham, August 2017.

CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Berlin, Heidelberg, May / June 2010.

CLM+18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Cham, December 2018.

Coc01. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *Cryptography and Coding*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363, Berlin, Heidelberg, 2001. Springer.

Cou06. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006.

DEF+25. Pierrick Dartois, Jonathan Komada Eriksen, Tako Boris Fouotsa, Arthur Herlédan Le Merdy, Riccardo Invernizzi, Damien Robert, Ryan Rueger, Frederik Vercauteren, and Benjamin Wesolowski. PEGASIS: Practical effective class group action using 4-dimensional isogenies. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part I*, volume 16000 of *LNCS*, pages 67–99. Springer, Cham, August 2025.

DEIV26. Pierrick Dartois, Jonathan Komada Eriksen, Riccardo Invernizzi, and Frederik Vercauteren. sfqt-sfPegasis: Simpler and faster effective class group actions. In Joan Daemen and Emmanuel Thomé, editors, *EUROCRYPT 2026, Part IV*, volume 16545 of *LNCS*, pages 636–665. Springer, Cham, May 2026.

DFK+23. Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: Scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 345–375. Springer, Cham, May 2023.

DG17a. Nico Döttling and Sanjam Garg. From selective IBE to full IBE and selective HIBE. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 372–408. Springer, Cham, November 2017.

DG17b. Nico Döttling and Sanjam Garg. Identity-based encryption from the Diffie-Hellman assumption. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 537–569. Springer, Cham, August 2017.

DG19. Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Cham, May 2019.

DGHM18. Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, and Daniel Masny. New constructions of identity-based and key-dependent message secure encryption schemes. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 3–31. Springer, Cham, March 2018.

DHK+23. Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Generic models for group actions. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 406–435. Springer, Cham, May 2023.

DM20. Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 187–212. Springer, Cham, May 2020.

GKW18. Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 660–670. ACM Press, June 2018.

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

GS25. Elif Ozbay Gurler and Patrick Struck. How (not) to build identity-based encryption from isogenies. In Christina Boura, Atefeh Mashatan, and Ali Miri, editors, *SAC 2025*, volume 16207 of *LNCS*, pages 589–615. Springer, Cham, August 2025.

GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

KLLQ23. Shuichi Katsumata, Yi-Fu Lai, Jason T. LeGrow, and Ling Qin. CSI-Otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 729–761. Springer, Cham, August 2023.

LR24. Antonin Leroux and Maxime Roméas. Updatable encryption from group actions. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *PQCrypto 2024, Part II*, pages 20–53. Springer, Cham, June 2024.

Mau05. Ueli Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding 2005*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.

OZ24. Emmanuela Orsini and Riccardo Zanotto. Simple two-message OT in the explicit isogeny model. *CiC*, 1(1):15, 2024.

PR23. Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766, 2023.

PRV12. Periklis A. Papakonstantinou, Charles W. Rackoff, and Yevgeniy Vahlis. How powerful are the DDH hard groups? Cryptology ePrint Archive, Report 2012/653, 2012.

RS06. Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006.

SGS21. Gili Schul-Ganz and Gil Segev. Generic-group identity-based encryption: A tight impossibility result. In Stefano Tessaro, editor, *ITC 2021*, volume 199 of *LIPIcs*, pages 26:1–26:23. Schloss Dagstuhl, July 2021.

Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Berlin, Heidelberg, August 1984.

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Berlin, Heidelberg, May 2005.

Wat05. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Berlin, Heidelberg, May 2005.

Wee22. Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Cham, May / June 2022.

Zha22. Mark Zhandry. To label, or not to label (in generic groups). In *Advances in Cryptology – CRYPTO 2022*, Lecture Notes in Computer Science. Springer, 2022.