How to Revoke Users in Encryption Schemes?

Sanjam Garg[*1], Dimitris Kolonelos[2], Benjamin Ostrovsky[†3], and Guru-Vamsi Policharla[4]

[1]UC Berkeley, sanjamg@berkeley.edu
[2]zkBricks, Inc., dimitriskolonelos@gmail.com
[3]Caltech, ben.ostrovsky@caltech.edu
[4]Commonware, Inc., guruvamsi.policharla@gmail.com

Abstract

Revocation is essential for keeping encryption secure as keys leak, devices are compromised or users leave the system. We revisit the problem of revocation for encryption schemes and give, to our knowledge, the first approach combining anonymous, collusion-resistant revocation with unchanged keys and access rules, succinct ciphertext overhead and a trustless revocation authority, i.e. that cannot decrypt. Revoking a user publishes one random-looking pseudonym, without telling encryptors whose key was revoked. A party can decrypt future ciphertexts only if it is unrevoked and satisfies the original access decryption rule.

Our core observation is that revocation is, at heart, witness encryption for the statement that one satisfies the access rule and that the pseudonym bound to one's key has not been revoked. Although general-purpose witness encryption remains far from practical, we observe that the statements arising from revocation are simple and algebraic, and we show that they fall within the special-purpose framework of Garg et al. (CRYPTO 2025), which compiles short arguments of knowledge (called "gadgets") into efficient witness encryption using bilinear groups. Our technical contribution is a pair of succinct non-membership gadgets for this framework, one for a single hidden pseudonym and one that certifies non-revocation jointly for all users selected by an access rule. Both gadgets are generic and modular, as they compose with the gadgets expressing a wide range of access rules, and thus provide a single revocation mechanism that is designed once and reused across different encryption schemes, without redesigning either the scheme or the revocation.

These gadgets yield the following applications. First, we give a black-box transformation adding revocation to arbitrary independently instantiated public-key encryption, with constant-size cryptographic ciphertext overhead. For unrevoked users, confidentiality continues to rely only on the security of the underlying scheme. Then, we demonstrate applications to advanced encryption scheme, constructing revocable schemes for distributed broadcast, registered attribute-based, and registered threshold encryption, with succinct cryptographic ciphertext components and threshold decryption shares. We prove adaptive security of our schemes in the generic bilinear-group and random-oracle models.

[*]Supported in part by the AFOSR Award FA9550-24-1-0156, and research grants from the Bakar Fund, Peder Sather Funds, the Stellar Development Foundation, Supra Inc., Category Labs, Input Output Global, and Byzantine Research Inc.
[†]Supported by the Taylor Lawrence SURF Fellowship at Caltech.

Contents

1 Introduction **3**
- 1.1 Our Contributions . . . 5

2 Technical Overview **7**
- 2.1 Non-membership to Revocation . . . 7
  - 2.1.1 Witness Encryption Framework. . . . 8
- 2.2 Constructing the Non-membership Arguments . . . 9

3 Preliminaries **10**
- 3.1 Polynomials . . . 10
- 3.2 Generic Group Model . . . 11
- 3.3 Commitment Schemes . . . 11
- 3.4 Arguments of Knowledge . . . 12
- 3.5 Witness Encryption . . . 13
  - 3.5.1 Witness Encryption Framework [GHK+25] . . . 14

4 Nonmembership Gadgets **15**
- 4.1 Nonmembership Gadget . . . 15
- 4.2 Indexed Non-membership Gadget . . . 17

5 Adding Revocation to Arbitrary Public-Key Encryption **22**
- 5.1 Definition . . . 23
- 5.2 Construction and Analysis . . . 24

6 Revocation for Advanced Encryption Schemes **26**
- 6.1 Revocable Distributed Broadcast Encryption . . . 26
  - 6.1.1 Definition . . . 26
  - 6.1.2 Construction and Analysis . . . 28
- 6.2 Revocable Attribute Based Encryption . . . 31
  - 6.2.1 Definition . . . 31
  - 6.2.2 Construction and Analysis . . . 33
- 6.3 Revocable Threshold Encryption . . . 35
  - 6.3.1 Definition . . . 35
  - 6.3.2 Construction and Analysis . . . 37

A Framework under Our Extended CRS **45**
- A.1 Setup and Public Inputs . . . 45
- A.2 Gadget Extraction . . . 46
  - A.2.1 Zero Check . . . 47
  - A.2.2 Indexed Inner Product . . . 47
- A.3 Normalization and Composition . . . 48
- A.4 IIP AuxGen Inversion Hardness . . . 49

1 Introduction

Encryption schemes allow a sender to encode a message so that only authorized parties can recover it. In public-key encryption [RSA78], for instance, each user generates a public key together with a secret decryption key and anyone can encrypt a message to that public key such that only the holder of the corresponding secret key can decrypt. This pair of keys is generated once but used for a long time: Senders keep encrypting to it, often for years, implicitly assuming that the corresponding secret key is still under the exclusive control of its legitimate owner. Over such an extended lifetime, this assumption cannot reasonably be expected to hold. Devices storing secret keys may be lost, stolen, or compromised, employees leave their organizations, members of a decryption committee may become corrupted and in some systems keys may simply expire. Ciphertexts created before such an event cannot be protected retroactively, but the greater damage lies ahead. The public key persists in address books, directories, and certificates, and senders continue to encrypt new messages under it, unaware that an unauthorized party can now read them. Analogous issues arise in more advanced encryption primitives, such as identity-based [BF01], attribute-based [SW05, BSW07], broadcast [FN94, BGW05], and threshold encryption [Des88, DF90].

In any long-lived encryption system, such events are not exceptional but inevitable. What keeps encryption meaningful in their presence is *revocation* [Mic96, NN98, ALO98]. After a short (public) update, a revoked key can no longer decrypt any ciphertext created afterwards, while all other users keep their decryption capabilities and carry on as before. Without revocation, every leaked key remains a liability for as long as senders keep encrypting to it.

Example 1: Revocation in case of lost keys. Suppose that Alice regularly encrypts messages to Bob and Bob's device, holding his secret key, is compromised. Bob wants future ciphertexts to be useless to the attacker and this should require no action from Alice, who may keep using the compromised public key without learning about the incident. That is, revocation should not expose Bob. Announcing which key was revoked reveals that Bob was successfully attacked and may identify the device on which the attack succeeded. First, this publicly attests that the device is exploitable, which turns it into a target for further attacks. Second, and more broadly, it potentially puts every other user running the same hardware or software at risk before the underlying weakness is fixed.

This motivates *revocation anonymity*. Ideally, revocation publishes only a random-looking *pseudonym* that nobody can link to Bob's key. Of course, some *revocation administrator* of the system must know this link, as otherwise only the owner of a key could revoke it, which is of little help if the owner has lost the device (and keys).[1] However, this authority should be the only party that knows the link, and it should not be able to decrypt.

Example 2: Selective revocation of devices. In practice, a user rarely holds a single key. A typical user owns several devices, such as a phone, a laptop, and a tablet, and each device holds its own secret key so that no key ever has to leave the device that generated it. An encryptor, however, wants to address the user rather than an individual device, so a ciphertext must be decryptable by any key in the set of device-keys associated with that user. Distributed broadcast encryption[2] [WQZD10, BZ14] captures this setting naturally, since parties generate their keys independently and a sender can encrypt to any set of public keys with succinct ciphertexts. Devices, however, are replaced, lost, or compromised far more often than users leave a system, which calls for *selective* revocation. It should be possible to revoke one or more devices of a user while the remaining devices keep decrypting, without the user resampling any key. Removing the revoked device from the recipient set is not a satisfactory solution. Functionally speaking, it forces every encryptor to track the status of each device, as for security, it reveals which device was revoked, resulting to similar attack vectors as discussed above. Instead, the encryptor should keep encrypting to the same set of keys, oblivious to which of them are still valid.

[1] On top of that, although unrelated to this example, in some cases we would like to revoke misbehaving users from a system.

[2] While (plain) Broadcast Encryption [FN94, BGW05] would functionally suffice, it introduces the notorious key-escrow problem, i.e., a single key-generation authority that can decrypt all ciphertexts of all users.

Although the literature on revocation is vast ([Mic96],[PTMW06],[NP01], [BGK08],[AI09],[LSW10] among hundreads of works), as we elaborate below, existing solutions remain unsatisfactory for encryption.

Pitfalls of prior approaches. In the classical approach, certificate revocation [Mic96, NN98, ALO98], a certificate authority publishes authenticated information on which certificates are revoked and asks encryptors not to encrypt to them. There, the revoked keys are identified in the clear, and some solutions require the authority to refresh the status of every unrevoked certificate in each period, which does not scale.

In what we call *indirect revocation*, an authority periodically issues key updates that only unrevoked users can use [BF01, BGK08, AI09, SSW12, SE13, ML19], re-certifies every unrevoked user [Gen03], or even takes part in every decryption [BDTW01]. In identity and attribute-based schemes, this authority can decrypt every ciphertext, and key updates grow with the number of revoked users and typically reveal who they are. Recent revocation mechanisms for registered encryption [FKdP23, Luo24, AAH+25] similarly exclude or delete the revoked keys publicly.

In *direct revocation*, the latter is enforced cryptographically via the decryption access structure. The encryptor excludes the revoked users explicitly, either by encoding them in the ciphertext [NP01, NNL01, BW06, AI09, LSW10, AAH+25] or by leaving them out of the set of recipients [BGW05, KMW23, Luo24]. The encryptor must then know who was revoked, and so does everyone who sees the ciphertext. Hiding the set of recipients instead is inherently expensive, as private broadcast encryption requires ciphertexts linear in the number of users [KS12].

Overall, the few schemes that do hide the revoked users [GMS12, BMN+21, BM24, Lee25] rely on an authority that issues all keys, and their ciphertexts or key updates grow with the number of revoked users. Finally, all these mechanisms are tailored to specific primitives, lacking modularity.

Revisiting revocation for encryption. We argue that revocation should be enforced by the cryptography itself rather than delegated to the encryptors. An encryptor need not know, and should not be able to learn, who has been revoked. It encrypts with respect to the current public revocation information, and the scheme guarantees that a coalition cannot decrypt unless its unrevoked members satisfy the ciphertext's access rule. The revocation update should be a single short, random-looking value, unlinkable to the actual information (public key or identity) of the user. Anonymity has been studied extensively for the recipients of a ciphertext, but hardly ever for revocation. Finally, since revoked users typically constitute only a small fraction of all users, we seek small ciphertext revocation components and online non-membership work to depend only on the revoked list. We account separately for registry-dependent setup, registration, and public helper data.

Our goals. Concretely, we put forth the following five criteria for revocation for encryption schemes.

1. **No re-keying.** Revoking a user should not require rekeying the whole system, which would be costly, interactive and prohibitively cumbersome for a large number of users.
2. **No decryption power.** The revocation authority should not be able to decrypt.
3. **Revocation Anonymity.** The public update may reveal that a revocation occurred, but not which user was revoked.
4. **Succinctness and Efficiency.** Revocation should add only constant-size user key material and succinct ciphertext components. We count the public revoked list and access-rule description separately, as well as shared parameters and registry-dependent helper data. After registry preprocessing, the additional work to encrypt or produce one user's non-membership proof should depend on the revoked-list size, rather than the registry size.
5. **Collusion-Resistance.** A coalition should not decrypt unless its unrevoked members satisfy the ciphertext's original access rule.[3]

[3]For threshold encryption, this qualification concerns the joint contributions of the unrevoked committee members.

To the best of our knowledge, no existing revocation mechanism for any encryption scheme achieves all of these properties simultaneously. Addressing this gap is the goal of the present work.

1.1 Our Contributions

We present a general approach for adding revocation to encryption schemes, that satisfies all the aforementioned properties. Our revocation mechanism is general and modular, it is designed once and enhances a broad class of encryption primitives. We instantiate it for public-key encryption and for several advanced encryption schemes. All our constructions are proven secure in the generic bilinear-group and random-oracle models [BBG05], assuming security of the underlying encryption scheme where used, with the registration and anonymity scope specified below.

Revocation through witness encryption. Our central observation is that revocation is, at its core, a witness encryption problem. A ciphertext should be decryptable exactly by those who can prove the statement

"I hold a secret key that satisfies the access rule of this ciphertext, and the pseudonym bound to this key is not in L."

where L is the list of revoked users. Witness encryption [GGSW13, GKP+13] lets the sender encrypt to such a statement directly, while the receiver decrypts with a proof that it computes locally. This enforces revocation cryptographically using public pseudonyms, without requiring the encryptor to know their associated identities. The anonymity guarantees are specified below. General-purpose witness encryption is far from practical. Our core technical observation is that the statements relevant to the above are simple and algebraic, which makes them amenable to recent efficient witness encryption techniques [GKPW24, GHK+25] based on bilinear groups. In particular, the framework of Garg et al. [GHK+25] compiles arguments of knowledge whose verification consists of pairing equations that are linear in the proof, called *gadgets*, into witness encryption, and preserves their succinctness. We thus reduce revocation to the design of new gadgets for the *set non-membership* relation, which can then be composed with the gadgets that express the access rule of the underlying encryption scheme.

Our setting. We start from an encryption scheme in which users generate their own keys, and enhance it with revocation. Besides its key pair, each user samples a random secret pseudonym y, which is bound to its public key by a perfectly hiding commitment $\mathsf{cm} = \mathsf{Com}(y; r)$. At registration, the user reveals the pseudonym and the opening of the commitment (only) to the revocation authority, which thereby learns the link between public keys and pseudonyms. To revoke a user, the authority appends the user's pseudonym y to a public bulletin board. Encryptors read the current list L of revoked pseudonyms from the board and encrypt with respect to it, and decryptors use the same list along with their secret keys.[4] Registration is the only interaction with the authority: encryptors and decryptors never contact it, and unrevoked users do not need to update their keys ever. The authority is trusted only to keep the link between the public key and pseudonym private, and to revoke users when, and only when, this is due. It holds no secret that helps decryption, and ciphertext confidentiality holds even against an adversary that knows all pseudonyms. We are in the structured reference string model, assuming all parties have access to a crs, while no party knows the corresponding trapdoor.

Non-membership gadgets. Our main technical contribution consists of two witness encryption gadgets, extending the framework of [GHK+25], certifying that pseudonyms, hidden inside public commitments, lie outside the revoked list L. Both build on the Bézout identity underlying non-membership proofs for bilinear-map accumulators [DT08, ATSM09]. For $P_L(Y) = \prod_{y \in L}(Y - y)$, a field element y lies outside L if and only if there exist a polynomial $u(Y)$ and a scalar v such that $u(Y) \cdot (Y - y) + v \cdot P_L(Y) = 1$. Our first gadget

[4] For a fixed list L, a succinct encoding of it can be stored. In the public-key construction, a recipient can also store its non-membership proof for that list. Processing the list and recomputing these data when it changes are separate costs.

concerns a single commitment. Since the corresponding y of a public key is hidden, the identity cannot be checked in the clear; we check it against the commitment $\mathsf{Com}(y; r)$ instead. The resulting proof consists of 5 group elements and 3 pairing equations, independently of $|L|$, and the reference string is linear in the maximum size of L.

Our second gadget is *indexed*: It takes the commitments of all n registered users together with a committed selector vector $\mathbf{b} \in \mathbb{F}_p^n$, and certifies that every selected user, that is, every i with $b_i \neq 0$, is unrevoked. Batching the n identities into a proof of constant size, verified by a constant number of pairing equations, is the main technical difficulty, for which we resort to bivariate polynomials and a corresponding reference string of size $O(n^2)$.

Revocable public-key encryption. Our first application adds revocation to *any* public-key encryption scheme (it needs not even be bilinear-group based), in a black-box way. A user's public key is augmented with the commitment to its pseudonym, a single group element. To encrypt, we secret share the message into two additive shares and encrypt one under the original public-key encryption scheme and the other with witness encryption compiled from our aforementioned non-membership gadget, both addressed to the same recipient. Apart from the public list L, a ciphertext consists of a ciphertext of the underlying scheme together with 7 group elements, regardless of the number of users and revoked users. The additional decryption work uses $O(|L| + 1)$ group operations, or $O(1)$ with a precomputed (for the same recipient and list) non-membership proof. We prove adaptive security under revocation, as well as perfect anonymity. Notably, for unrevoked recipients, confidentiality reduces to the security of the underlying scheme alone. Revocation never weakens the scheme it is added to, and the generic group and random oracle models are only needed to lock revoked users out.[5]

Revocation for advanced encryption schemes. Secret sharing suffices for a single recipient, but not for richer access rules. In broadcast encryption, for instance, revocation is trivially not secure against collusions. A revoked recipient could still recover one share using their secret key while any unrevoked user outside the recipient set recovers the other share. Authorization and non-revocation must therefore be enforced for the *same* users, which is exactly what our indexed gadget provides. By routing the same committed selector to it and to the gadgets implementing the access rule, every valid witness selects users that are simultaneously authorized and unrevoked. Applying this to the schemes of [GHK+25], we obtain revocable distributed broadcast encryption [WQZD10, BZ14], with constant-size ciphertexts and master public key (counting the explicit recipient set and revoked list separately, as is common in broadcast encryption) revocable registered attribute-based encryption [HLWW23] for DNF policies, where revocation adds a constant number of group elements per clause, and revocable registered threshold encryption [GHK+25], where decryption requires t unrevoked committee members and partial decryptions remain succinct. The last case requires extra care. Partial decryptions are public, so the non-membership proofs they contain could later be tested against newly published pseudonyms. We rerandomize these proofs while preserving their aggregation. The resulting threshold-anonymity guarantee is historical. Honest partials for contexts created before anonymous publication remain unlinkable afterwards. Honest partial-decryption queries may continue for those contexts with their original revoked lists, but the experiment permits no new contexts for that oracle after publication. Public computations remain unrestricted, and the challenge users remain uncorrupted and otherwise unrevealed. Under these anonymity notions, revocation adds a constant number of ciphertext group elements (per conjunction for DNF policies), apart from the public context. Once the encoding of the list, and registry helper data are prepared, the additional non-membership encryption uses constantly many group operations, and a proof for one selected user uses $O(|L| + 1)$ group operations. Revocation itself does not update user keys or master public keys. With M maximum number of revocation, the indexed constructions use an $O(M^2)$ reference string and $O(M^2)$ auxiliary data per user. DBE and threshold encryption retain $O(M^2)$ public data.

[5]Notably, our revocation scheme can be composed with post-quantum secure public encryption schemes and, while security of revocation naturally won't be secure against quantum adversaries, it remains secure against "harvest-now-decrypt-later" attacks.

2 Technical Overview

Our constructions build on the classic idea that revocation can be expressed as non-membership in a list alongside the usual conditions for decryption. That is, a user should be able to decrypt only if they can prove that their revocation identifier is not in a "revoked list." Towards this goal, we develop a succinct non-membership argument and a corresponding witness encryption scheme, then use these to construct revocable public-key encryption (PKE) [RSA78], distributed broadcast encryption (DBE) [WQZD10, BZ14], registered attribute-based encryption (R-ABE) [HLWW23] and registered threshold encryption [GHK+25]. Let us first establish the setting for revocation.

Setting. We consider a registered setting with a polynomial number of users and a public bulletin board. Users generate their own secret keys and register public information that will be used for encryption. To use non-membership for revocation, we first need to decide which values appear in the revoked list. Since we want anonymous revocation, publishing these values should allow encryptors to exclude revoked users without revealing their identities. Each user therefore samples a private revocation identifier, a uniformly random field element $y \in \mathbb{F}_p$, included in its secret key. We call this value the user's *pseudonym*.

We model the revocation process with a Revocation Authority (RA), which knows the identity-to-pseudonym map and can therefore revoke users by posting their pseudonyms to the bulletin board.[6] The values posted so far form the revoked set L, and subsequent encryptions should require the recipient's pseudonym to lie outside L. For simplicity, we assume only the RA writes to the bulletin board and is trusted to keep the identity-to-pseudonym map confidential. Each revocation adds only one field element—a pseudonym—to L.

To prevent a revoked user from choosing a fresh pseudonym, its public key includes $c = \mathsf{Com}(y; r)$, with fresh randomness r and opening (y, r) retained in its secret key. The commitment binds the pseudonym and hides it jointly with all public registration data. Non-revocation requires knowledge of an opening of this commitment with $y \notin L$. Moreover, we seek small user keys and succinct ciphertext revocation components, with online non-membership work depending only on the revoked-list size after registry preprocessing. The public list, shared parameters, and registry helper data are counted separately.

With the setting established, let us detail how non-membership gives revocation.

2.1 Non-membership to Revocation

Suppose that we have a succinct witness encryption scheme [GGSW13, GKP+13] for the following relation

$$\mathsf{NM} = \left\{ \ (\mathsf{ck}, c, L); (y, r) \ \middle| \ c = \mathsf{Com}_{\mathsf{ck}}(y; r) \wedge y \notin L \ \right\},$$

A user who knows such an opening can construct the required witness and decrypt locally. We will explain how to obtain this scheme from our non-membership argument below, but first, what can we do with it?

Revocable Public-Key Encryption. This already gives a simple construction of revocable public-key encryption (PKE). Take *any* IND-CPA-secure PKE scheme PKE, together with a witness-encryption scheme $\mathsf{WE}_{\mathsf{NM}}$. A user publishes an ordinary public key together with a commitment $c = \mathsf{Com}(y; r)$. To encrypt a message msg, choose a uniformly random share R and secret-share the message into R and $\mathsf{msg} - R$. Encrypt R under the ordinary public key and encrypt $\mathsf{msg} - R$ using non-membership encryption for c and L. That is, form

$$\mathsf{ct}_0 \leftarrow \mathsf{PKE}.\mathtt{Enc}(\mathsf{pk}, R), \qquad \mathsf{ct}_{\mathsf{NM}} \leftarrow \mathsf{WE}_{\mathsf{NM}}.\mathtt{Enc}\big(\mathsf{ed}_{\mathsf{NM}}, L, \mathsf{msg} - R\big),$$

where $\mathsf{ed}_{\mathsf{NM}}$ is a public encryption digest for c, and the two encryptions use independent randomness. An unrevoked recipient recovers both shares and adds them to obtain msg.

[6] The role of the RA is simply a modeling convenience. A user could post their own pseudonym to the same effect, though the posting must not reveal the *poster*, otherwise the hiding would not hold.

Revocable Broadcast Encryption. We, further, desire more complex encryption schemes, where encryptions need not be issued to one particular user. Take, for example, broadcast encryption, where an encryptor chooses a set S of recipients. Revocable broadcast encryption is a well-studied problem [NP01, NNL01, BW06, BMN+21]. In broadcast encryption, the cryptographic ciphertext components should have size sublinear in $|S|$.[7] However, applying the revocable public-key construction separately for each recipient loses this succinctness. It requires one non-membership ciphertext per recipient, making the ciphertext size grow linearly with $|S|$.

A tempting approach is to resort again to a secret-sharing technique, that is, to encrypt one share for S and make the other share decryptable by any unrevoked user. However, this admits a collusion attack: a revoked recipient in S can recover the first share, while an unrevoked user outside S can recover the second. Together they recover the message, although neither is an unrevoked recipient. We therefore need a relation that enforces authorization and non-revocation for the same users.

2.1.1 Witness Encryption Framework.

The framework of Garg et al. [GHK+25] allows us to do exactly this. In particular, the framework constructs witness encryption from *linearly verifiable* relations. A relation is linearly verifiable if it admits a non-interactive argument of knowledge whose verifier checks equations of the form $\mathbf{A} \circ \pi = \mathbf{c}$, where $\mathbf{A}$ and $\mathbf{c}$ are determined by the public inputs, π contains the proof elements, and $\circ$ denotes matrix-vector multiplication using bilinear pairings. The framework gives a compiler to turn such arguments into witness encryption, with succinct proofs yielding succinct ciphertexts.

To construct distributed broadcast encryption, the framework has each user generate a BLS [BLS04] signing key sk_i and publish its verification key pk_i. It uses a vector $b = (b_1, \ldots, b_n)$ to indicate which users participate. $b_i = 0$ means that user i is not selected. This *selector* must have at least one nonzero entry and may select only recipients in S. Its entries weight the public keys to form

$$\mathsf{pk}_b = \sum_{i=1}^{n} b_i \mathsf{pk}_i.$$

An *indexed inner-product* (IIP) argument checks this equality succinctly. A signature argument checks a signature under pk_b on a fresh public tag chosen by the sender. These arguments use the same aggregate key, and all checks on b use the same commitment to it.

The sender encrypts using witness encryption for these combined checks. A recipient $i \in S$ sets $b_i = 1$ and all other entries to zero, so $\mathsf{pk}_b = \mathsf{pk}_i$. It then signs the tag using sk_i and computes the proof needed to decrypt. This gives DBE without a separate ciphertext for each recipient. For more details on the IIP argument, see Appendix A.2.2.

Indexed Non-membership. To add revocation, we must also check that every user selected by this same b is unrevoked. This motivates an *indexed non-membership argument* for the relation

$$\mathsf{INM} = \left\{ (\mathsf{ck}, (c_1, \ldots, c_n), L); (b, y, r) \;\middle|\; \begin{array}{l} \forall i \in \{1, \ldots n\}, \\ b_i \neq 0 \implies (c_i = \mathsf{Com}_{\mathsf{ck}}(y_i; r_i) \wedge y_i \notin L) \end{array} \right\}.$$

Each selected position requires an opening to an unrevoked pseudonym. When $b_i = 0$, no condition is imposed on (y_i, r_i).

We make the broadcast and indexed non-membership arguments share a commitment to the same selector b. Both arguments must therefore use the same selected users. The combined argument checks that these users satisfy the broadcast-decryption condition and are unrevoked, ruling out the collusion above. Applying the witness-encryption compiler gives revocable DBE. For more details, see Section 6.1. The extensions to registered ABE and registered threshold encryption appear in Sections 6.2 and 6.3, respectively.

[7] In broadcast encryption the public recipient set is counted separately; the objective is to minimize the cryptographic overhead, i.e., the ciphertext size.

2.2 Constructing the Non-membership Arguments

We next explain how to construct the non-membership arguments. To this end, we rely on Bézout's identity, a technique previously used in bilinear-map accumulators by Damgård and Triandopoulos [DT08] and by Au et al. [ATSM09].

Non-membership. First, associate L with the polynomial $P_L(Y) = \prod_{y \in L}(Y - y)$, with $P_\emptyset(Y) = 1$. This polynomial has roots exactly at the elements of the revoked set. Thus $y \notin L$ exactly when $Y - y$ and $P_L(Y)$ are relatively prime. By Bézout's identity, this is equivalent to the existence of a polynomial $u(Y)$ and a scalar v satisfying

$$u(Y) \cdot (Y - y) + v \cdot P_L(Y) = 1.$$

We check this identity "in the exponent." Polynomials are represented by group encodings of their evaluations at a hidden point θ from the CRS, and multiplicative relations are checked using the pairing. For fixed y, verification is linear in $u(Y)$ and v. The factors $Y - y$ and $P_L(Y)$ belong to the statement. This makes this relation amenable to the witness encryption framework of Garg et al. [GHK+25].

To keep y hidden, we use a Pedersen commitment [Ped91], with an additional proof element canceling the commitment's blinding term in the check. Together with a check that the opening agrees with the registered commitment $c = \mathsf{Com}(y; r)$, this gives our non-membership argument. For more details, see Section 4.1.

Indexed Non-membership. For indexed non-membership, we use the same idea for each user, replacing the right-hand side by its selector coordinate

$$u_i(Y) \cdot (Y - y_i) + v_i \cdot P_L(Y) = b_i.$$

When $b_i \neq 0$, this is a scaled Bézout identity and requires $y_i \notin L$. When $b_i = 0$, setting $u_i = v_i = 0$ satisfies the identity without imposing any condition on y_i.

Checking these identities separately would require a proof whose size grows with the number of users

$$\begin{aligned} u_1(Y)(Y - y_1) + v_1 P_L(Y) &= b_1, \\ &\vdots \\ u_n(Y)(Y - y_n) + v_n P_L(Y) &= b_n. \end{aligned}$$

Instead, we interpolate over a second variable X, assigning a distinct point d_i to each user and writing $\mathcal{D} = \{d_1, \ldots, d_n\}$. For clarity, temporarily suppress the commitment blinding. Let $\tilde{p}, U, V, B$ be the polynomials of degree less than n in X satisfying $\tilde{p}(d_i, Y) = Y - y_i$, $U(d_i, Y) = u_i(Y)$, $V(d_i) = v_i$, and $B(d_i) = b_i$. That is, the unique polynomials that interpolate to p_i, u_i, v_i, b_i respectively at $X = d_i$. Writing $Z_{\mathcal{D}}(X) = \prod_{i=1}^{n}(X - d_i)$ for the vanishing polynomial of $\mathcal{D}$, the n identities above become

$$\tilde{p}(X, Y)U(X, Y) + V(X)P_L(Y) \equiv B(X) \pmod{Z_{\mathcal{D}}(X)}.$$

Working modulo $Z_{\mathcal{D}}(X)$ requires the equality to hold at each of the n points in $\mathcal{D}$, as an identity in Y. At $X = d_i$, it becomes exactly user i's Bézout identity.

We check this congruence in the exponent, using the registered commitments and canceling their blinding terms as above. Together with the opening checks, this gives a constant number of pairing equations and proof elements, while retaining the shared selector commitment that ties non-revocation to the broadcast argument. The proofs and additional ciphertext components have constant size, independent of both the number of users and $|L|$. For more details, see Section 4.2.

3 Preliminaries

Notation. With λ we will denote the security parameter, with negl a negligible function, (i.e. for *any* polynomial $f(x)$, $\mathsf{negl}(x) < 1/f(x)$ for all sufficiently large x), and with poly a polynomial function in its variables (they can be more than 1). We use $[\![n]\!]$ for the set $\{1, \ldots, n\}$, where $n > 0$ is a positive integer. We denote sets with calligraphic capital letters, e.g. $\mathcal{S}$. Bold symbols denote vectors or matrices, e.g. $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{A} = (a_{i,j})_{i,j}$. Subscripts of the form $(\mathbf{x}_{i,j})_{i\in\mathcal{S},\, j\in\mathcal{P}}$ range independently over all $i \in \mathcal{S}$ and $j \in \mathcal{P}$. Similarly, $(\mathbf{x}_{i,j})_{i,j\in\mathcal{S}}$ means that i and j each range over all of $\mathcal{S}$. For a tuple $\mathbf{x}$ and a set S of its positions, $(x_i)_S$ is shorthand for $(x_i)_{i\in S}$. For vector spaces U, V over the same field, $\mathrm{Hom}(U, V)$ denotes the set of linear maps from U to V. When writing relations and verification equations, we highlight the terms supplied by the prover, including witness commitments and proof elements

Bilinear Groups. A Bilinear Group $\mathcal{BG}$, generated as $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{BG}(1^\lambda)$, is specified by a field $\mathbb{F}_p$ of prime order $p = 2^{\Theta(\lambda)}$, three groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ (the first two we call "source groups" and the third "target group"), a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that we call "pairing" and one random generator g_1, g_2 for each source group. We use the *implicit notation*, i.e., $[x]_s := x \cdot g_s$ and more generally $[\mathbf{A}]_s$ represents a matrix of the corresponding group elements, for $s \in \{1, 2, T\}$. We denote the group operation additively, $[x]_s + [y]_s = [x + y]_s$, for $s \in \{1, 2, t\}$. By $[x]_1 \circ [y]_2 = [y]_2 \circ [x]_1 = [x \cdot y]_T$, we denote the pairing $e([x]_1, [y]_2)$. We note that the way it is defined $\circ$ is commutative, for instance $([a]_1, [b]_2)^\intercal \circ ([c]_2, [d]_1)$ is well-defined and gives the outcome $[ac + bd]_T$. In our constructions, we will sometimes omit writing explicitly the Bilinear Group in the algorithms' inputs, even though it is implicitly taken as input. For a group with fixed generator g, we write $[a]$ to mean $a \cdot g$, omitting the group subscript when clear from context.

3.1 Polynomials

Throughout the paper, polynomials are over the scalar field $\mathbb{F}_p$. We write $\mathbb{F}_p[X]_{<d}$ for polynomials of degree less than d.

For Lagrange interpolation, let $\mathcal{S} = \{s_1, \ldots, s_M\} \subseteq \mathbb{F}_p$ be a set of distinct field elements. For each $i \in [\![M]\!]$, we define the Lagrange basis polynomial

$$L_i^{\mathcal{S}}(X) = \prod_{\substack{j\in[\![M]\!] \\ j\neq i}} \frac{X - s_j}{s_i - s_j}.$$

These polynomials satisfy $L_i^{\mathcal{S}}(s_i) = 1$ and $L_i^{\mathcal{S}}(s_j) = 0$ for each $j \in [\![M]\!] \setminus \{i\}$. Consequently, for $y_1, \ldots, y_M \in \mathbb{F}_p$, the unique polynomial $f(X) \in \mathbb{F}_p[X]_{<M}$ satisfying $f(s_i) = y_i$ for all $i \in [\![M]\!]$ is exactly $f(X) = \sum_{i\in[\![M]\!]} L_i^{\mathcal{S}}(X) y_i$. Similarly, for $f_1(\mathbf{Y}), \ldots, f_M(\mathbf{Y}) \in \mathbb{F}_p[Y]$ where $\mathbf{Y}$ is a tuple of indeterminates, the unique polynomial of degree less than M in X satisfying $F(s_i, \mathbf{Y}) = f_i(\mathbf{Y})$ for all $i \in [\![M]\!]$ is exactly $F(X, \mathbf{Y}) := \sum_{i\in[\![M]\!]} L_i^{\mathcal{S}}(X) f_i(\mathbf{Y})$.

For a finite set $\mathcal{V} \subseteq \mathbb{F}_p$, its vanishing polynomial is defined as

$$Z_{\mathcal{V}}(X) = \prod_{v\in\mathcal{V}} (X - v).$$

We recall a useful property: if $f(X)$ is divisible by $Z_{\mathcal{V}}(X)$, i.e. there is a quotient polynomial $Q(X)$ such that $f(X)/Z_{\mathcal{V}}(X) = Q(X)$, then $f(v_1) = \ldots = f(v_k) = 0$. More generally, for any polynomial $F(X, \mathbf{Y})$, $Z_{\mathcal{V}}(X) \mid F(X, \mathbf{Y}) \iff F(v, \mathbf{Y}) = 0$ for every $v \in \mathcal{V}$, where equality is as a polynomial in $\mathbf{Y}$.

We use $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \mathbb{F}_p$ for the fixed interpolation domain in our constructions. This may be roots-of-unity for efficiency, or simply arbitrary distinct field elements.

Generalized Univariate Sumcheck. [RZ21] constructed a generalized variant of the univariate sumcheck [BCR+19] that works over arbitrary domains. We recall their construction but generalize to support inner product arguments instead of just a sumcheck.

Lemma 1. *Let $\mathcal{D} = \{d_1, \ldots, d_n\} \subset \mathbb{F}_p$ be an arbitrary domain of size n and $Z_{\mathcal{D}}(X)$ be the vanishing polynomial over $\mathcal{D}$. Let $A(X) \in \mathbb{F}_p[X]$, be a polynomial of arbitrary degree, and $b = (b_1, \ldots, b_n) \in \mathbb{F}_p^n$. For any $x^* \in \mathbb{F}_p \setminus \mathcal{D}$, we define the polynomial $B(X) \coloneqq \sum_{i=1}^n b_i \cdot L_i^{\mathcal{D}}(x^*)^{-1} \cdot L_i^{\mathcal{D}}(X)$. We then have that for all $x^* \in \mathbb{F}_p \setminus \mathcal{D}$ and all $s \in \mathbb{F}_p$, $\sum_{i=1}^n b_i \cdot A(d_i) = s$ iff there exist polynomials $Q_Z(X)$, $Q_X(X)$ such that $\deg(Q_X(X)) \leq n-2$, and*

$$A(X) \cdot B(X) = s + Q_X(X) \cdot (X - x^*) + Q_Z(X) \cdot Z_{\mathcal{D}}(X).$$

In particular,

$$A(X)B(X) = \sum_{i=1}^{n} A(d_i)B(d_i)L_i^{\mathcal{D}}(X) + Q_Z(X)Z_{\mathcal{D}}(X). \tag{1}$$

Let $R = AB \bmod Z_{\mathcal{D}}$, so $\deg R < n$. The choice of B gives $R(x^*) = \sum_i b_i A(d_i)$. Thus the inner-product condition holds exactly when $R - s = (X - x^*)Q_X$ with $\deg Q_X \leq n - 2$. Adding the multiple of $Z_{\mathcal{D}}$ gives the global identity above.

3.2 Generic Group Model

Our gadget security proofs use the Generic (Bilinear) Group Model [Sho97, Mau05]. A 'generic' adversary does not have concrete representations of the elements of the group and can only use generic group operations. This model captures the possible 'algebraic' attacks that an adversary can perform.

In particular, we follow the Maurer's GGM [Mau05], which is extended to Bilinear Groups by [BBG05]. The adversary in GGM makes oracle queries for each generic group operation she wishes to perform and receives a handle for the resulting group element, instead of the actual element itself. We call the party that answers the queries the 'challenger'. The challenger keeps three lists L_1, L_2, L_T of all group elements resulted from the queries of the adversary together with their corresponding handles.

Throughout our generic-group security proofs, we work in the symbolic experiment, representing group elements by formal polynomials in the sampled secrets. An accidental collision occurs when two distinct formal polynomials representing elements of the same group evaluate to the same field element. For polynomially many polynomials of polynomially bounded degree over our exponentially large field, the standard collision bound is negligible [Sho97, BBG05]. We therefore exclude accidental collisions in the symbolic analysis.

For further discussion, see [Pol26, Chapter 2, Definition 2.1, pp. 11–12].

3.3 Commitment Schemes

Definition 1 (Commitment Scheme)**.** *A commitment scheme* $\mathsf{Com} = (\mathtt{CGen}, \mathtt{Commit})$ *consists of a PPT key-generation algorithm, which may be randomized, that outputs a public commitment key* ck*, and a deterministic PPT algorithm* $\mathtt{Commit}(\mathsf{ck}, m)$ *mapping messages $m \in \mathcal{M}$ to commitments in a commitment space $\mathcal{C}$. It is* computationally binding *if, for an honestly generated* ck*, every PPT adversary finds distinct messages $m \neq m'$ satisfying $\mathtt{Commit}(\mathsf{ck}, m) = \mathtt{Commit}(\mathsf{ck}, m')$ only with negligible probability. Any trapdoor used to generate* ck *is sampled honestly and is not given to the adversary.*[8]

Below we recall the structured commitment scheme of [GHK+25]. Let $\mathcal{D} = \{d_1, \ldots, d_n\} \subset \mathbb{F}_p$ and let $L_1^{\mathcal{D}}(X), \ldots, L_n^{\mathcal{D}}(X)$ be its Lagrange polynomials.

Construction 1 (Structured Commitment Scheme)**.** *Consider group $\mathbb{G}$ over scalar field $\mathbb{F}$. We construct a structured commitment scheme* $\mathrm{SCS}(\mathbb{G})$ *characterized by group $\mathbb{G}$ over message space $\mathbb{F}^n$ and commitment space $\mathbb{G}$ as follows.*

[8] For this work we do not require hiding.

- $\mathtt{CGen}_{\mathsf{SCS}}(\lambda, n, \tau \in \mathbb{F})$: *Output* $\mathsf{ck} = ([\tau^{a-1}])_{a \in [\![n]\!]}$.
- $\mathtt{Commit}_{\mathsf{SCS}}(\mathsf{ck}, w)$: *For* $w = (w_1, \ldots, w_n) \in \mathbb{F}_p^n$, *let* $W(X) = \sum_{i \in [\![n]\!]} L_i^{\mathcal{P}}(X) w_i$. *Output* $\overline{w} = [W(\tau)]$.

An *indexed relation* [CHM+20, COS20] $\mathcal{R}$ is a set of tuples $(\mathsf{pp}, \mathsf{idx}, u, w)$ with efficiently decidable membership, where pp are public parameters, idx is a public index, u is an instance (the "claim"), and w is a witness.

Definition 2 (Committed Relation)**.** *Let $\mathcal{R}$ be an indexed relation and let $\mathbb{W}$ denote the set of possible values of its witness w. Let $\mathsf{Com} = (\mathtt{CGen}, \mathtt{Commit})$ be a deterministic commitment scheme with message space $\mathbb{W}$. We define the committed relation $[\mathcal{R}]_{\mathsf{Com}}$ as follows.*

$$[\mathcal{R}]_{\mathsf{Com}} = \left\{ \begin{array}{l} \textit{parameters } (\mathsf{pp}, \mathsf{ck}), \\ \textit{index } \mathsf{idx}, \\ \textit{instance } (\overline{w}, u), \\ \textit{witness } w \end{array} \;\middle|\; \begin{array}{l} (\mathsf{pp}, \mathsf{idx}, u, w) \in \mathcal{R}, \\ \overline{w} = \mathtt{Commit}(\mathsf{ck}, w) \end{array} \right\}.$$

3.4 Arguments of Knowledge

We follow the argument-of-knowledge definition of [GHK+25]. We write ρ for the oracles available to the algorithms and adversary. For our later constructions, these are the generic bilinear-group oracles described in Section 3.2.

Definition 3 (Non-Interactive Argument of Knowledge)**.** *Consider an indexed relation $\mathcal{R}$. An argument of knowledge in oracle model ρ for $\mathcal{R}$ is defined by PPT algorithms $(\mathtt{Gen}, \mathtt{Prove}, \mathtt{Verify})$ called the generator, the prover, the verifier respectively and deterministic algorithms $\mathtt{AuxGen}$ and $\mathtt{Digest}$, all with access to ρ with the following interface.*

- $\mathtt{Gen}(\lambda, N, \mathsf{td}) \to (\mathsf{pp}, \mathsf{crs})$: *Takes as input the security parameter λ, size parameters N, and trapdoor $\mathsf{td} \in \mathcal{T}$. Deterministically outputs the relation parameters pp and common reference string crs deterministically.*
- $\mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i) \to (\mathsf{idx}_i, \mathsf{aux}_i)$: *Takes as input pp, crs, and a private value s_i. Deterministically outputs an index component idx_i and public auxiliary value aux_i.*
- $\mathtt{Digest}((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux}) \to (\mathsf{Pk}, \mathsf{Vk})$: *Takes as input pp, crs, index $\mathsf{idx} = (\mathsf{idx}_1, \ldots, \mathsf{idx}_n)$, and auxiliary values aux. Deterministically outputs prover key Pk and verifier key Vk deterministically.*
- $\mathtt{Prove}(\mathsf{Pk}, u, w) \to \pi$: *Takes as input the prover key Pk, public instance u, and witness w. Outputs a proof π. This algorithm may be randomized.*
- $\mathtt{Verify}(\mathsf{Vk}, u, \pi) \to \{0, 1\}$: *Takes as input the verifier key Vk, instance u, and proof π. Outputs a bit in $\{0, 1\}$ deterministically, with output 1 exactly when the proof is valid.*

For m private inputs, we write $\mathsf{idx} = (\mathsf{idx}_1, \ldots, \mathsf{idx}_m)$ and $\mathsf{aux} = (\mathsf{aux}_1, \ldots, \mathsf{aux}_m)$. All parties share the same oracles and their state. An argument of knowledge $(\mathtt{Gen}, \mathtt{AuxGen}, \mathtt{Digest}, \mathtt{Prove}, \mathtt{Verify})$ with trapdoor space $\mathcal{T}$ satisfies the following properties:

1. ***Perfect completeness.*** *For all $(u, w, (s_1, \ldots, s_m))$ with valid private inputs s_i, sample $\mathsf{td} \leftarrow_{\$} \mathcal{T}$ and compute $(\mathsf{pp}, \mathsf{crs}) \leftarrow \mathtt{Gen}(\lambda, N, \mathsf{td})$. For each $i \in [\![m]\!]$, compute $(\mathsf{idx}_i, \mathsf{aux}_i) \leftarrow \mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i)$. If $(\mathsf{pp}, \mathsf{idx}, u, w) \in \mathcal{R}$, compute $(\mathsf{pk}, \mathsf{vk}) \leftarrow \mathtt{Digest}((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux})$ and $\pi \leftarrow \mathtt{Prove}(\mathsf{pk}, u, w)$. Then*

$$\Pr[\mathtt{Verify}(\mathsf{Vk}, u, \pi) = 1] = 1.$$

2. ***Knowledge soundness.*** *There exists a PPT straight-line extractor E such that, for every PPT oracle prover $\mathtt{Prove}^*$ with well-formed outputs, given $\mathsf{td} \leftarrow_{\$} \mathcal{T}$, $(\mathsf{pp}, \mathsf{crs}) \leftarrow \mathtt{Gen}(\lambda, N, \mathsf{td})$, $((s_1, \ldots, s_m), u, \pi, \mathsf{tr}) \leftarrow$*

$\mathtt{Prove}^*(\mathsf{pp}, \mathsf{crs})$, $\{(\mathsf{idx}_i, \mathsf{aux}_i) \leftarrow \mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i)\}_{i \in [\![m]\!]}$, *and* $(\mathsf{Pk}, \mathsf{Vk}) \leftarrow \mathtt{Digest}((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux})$, *we have*

$$\Pr\left[\mathtt{Verify}(\mathsf{Vk}, u, \pi) = 1 \ \wedge (\mathsf{pp}, \mathsf{idx}, u, E((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux}, u, \pi, \mathsf{tr})) \notin \mathcal{R}\right] \leq \mathsf{negl}(\lambda),$$

where tr *is the recorded list of oracle queries and responses made by* $\mathtt{Prove}^*$.

Definition 4 (Succinctness)**.** *An argument of knowledge is* succinct *if the proof is constant, or sublinear in the size of the index and witness.*

Definition 5 (Linear Verifiability)**.** *A non-interactive argument of knowledge* Π *for a committed relation* $\mathcal{R}$ *is linearly verifiable over* $\mathrm{Hom}(W, V)$ *if there exists some linear transformation* $A_{\mathsf{vk},u} \in \mathrm{Hom}(W, V)$ *and* $b_{\mathsf{vk},u} \in V$ *such that*

$$\Pi.\mathtt{Verify}(\mathsf{vk}, (\overline{w}, u), \pi) = 1 \quad \iff \quad A_{\mathsf{vk},u}(\overline{w}, \pi) = b_{\mathsf{vk},u}.$$

Likewise, a committed relation $\mathcal{R}$ *is linearly verifiable over* $\mathrm{Hom}(W, V)$ *if there exists an argument of knowledge* Π *that is linearly verifiable over* $\mathrm{Hom}(W, V)$ *for* $\mathcal{R}$.

Definition 6 (Reduction)**.** *For relations* $\mathcal{R}_1$ *and* $\mathcal{R}_2$ *in NP, a reduction* Π *from* $\mathcal{R}_1$ *to* $\mathcal{R}_2$ *(denoted* $\Pi : \mathcal{R}_1 \to \mathcal{R}_2$*) consists of two polynomial-time, deterministic, polynomial-time invertible functions, the instance map* ϕ_{inst} *and the witness map* ϕ_{wit}*, such that for* $(u_2, w_2) \leftarrow \phi_{\mathrm{wit}}(\mathsf{pp}, \mathsf{idx}, u_1, w_1)$ *and* $u_2' \leftarrow \phi_{\mathrm{inst}}(u_1)$ *we have that* $u_2 = u_2'$ *and*

$$(\mathsf{pp}, \mathsf{idx}, u_2, w_2) \in \mathcal{R}_2 \text{ if and only if } (\mathsf{pp}, \mathsf{idx}, u_1, w_1) \in \mathcal{R}_1.$$

3.5 Witness Encryption

Witness encryption [GGSW13, GKP+13] encrypts a message relative to a public instance. A party holding a witness that satisfies the indexed relation can use the public decryption digest to produce a proof that enables message recovery. Similarly to Definition 3, we follow the syntax of [GHK+25].

Definition 7 (Witness Encryption)**.** *A witness-encryption scheme* $\mathsf{WE} = (\mathtt{Gen}, \mathtt{AuxGen}, \mathtt{Digest}, \mathtt{Enc}, \mathtt{Prove}, \mathtt{Dec})$ *in oracle model* ρ *for the indexed relation* $\mathcal{R}$*, with message space* $\mathcal{M}$ *and trapdoor space* $\mathcal{T}$ *equipped with an honest sampler, consists of the following six PPT algorithms, each with access to* ρ.

- $\mathtt{Gen}(\lambda, N, \mathsf{td}) \to (\mathsf{pp}, \mathsf{crs})$*: Takes as input the security parameter* λ*, length parameter* N*, and trapdoor* $\mathsf{td} \in \mathcal{T}$*. Outputs structured public parameters* pp *and a common reference string* crs *deterministically.*
- $\mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i) \to (\mathsf{idx}_i, \mathsf{aux}_i)$*: Takes as input* pp*,* crs*, and a private value* s_i*. Outputs an index component* idx_i *and public auxiliary value* aux_i *deterministically.*
- $\mathtt{Digest}((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux}) \to (\mathsf{ed}, \mathsf{dd})$*: Takes as input* pp*,* crs*, the assembled index* idx*, and auxiliary values* aux*. Outputs the public encryption digest* ed *and public decryption digest* dd *deterministically.*
- $\mathtt{Enc}(\mathsf{ed}, u, \mathsf{msg}) \to \mathsf{ct}$*: Takes as input the encryption digest* ed*, instance* u*, and message* $\mathsf{msg} \in \mathcal{M}$*. Outputs a ciphertext* ct*; this algorithm may be randomized.*
- $\mathtt{Prove}(\mathsf{dd}, u, w) \to \pi$*: Takes as input the decryption digest* dd*, instance* u*, and witness* w*. Outputs a proof* π.
- $\mathtt{Dec}(\mathsf{ct}, \pi) \to \mathsf{msg}$*: Takes as input a ciphertext* ct *and proof* π*. Outputs* msg *when* π *is valid.*

For m *private inputs, we write* $\mathsf{idx} = (\mathsf{idx}_1, \ldots, \mathsf{idx}_m)$ *and* $\mathsf{aux} = (\mathsf{aux}_1, \ldots, \mathsf{aux}_m)$*. All parties share the same oracles and their state throughout the experiment. A witness encryption scheme* $(\mathtt{Gen}, \mathtt{AuxGen}, \mathtt{Digest}, \mathtt{Enc}, \mathtt{Prove}, \mathtt{Dec})$ *over trapdoor space* $\mathcal{T}$ *for relation* $\mathcal{R}$ *satisfies the following properties:*

1. ***Perfect correctness.*** *Fix any polynomially bounded m, N, instance u, witness w, message $\mathsf{msg} \in \mathcal{M}$, private inputs $s_1, \ldots, s_m$, and λ large enough. Sample $\mathsf{td} \leftarrow_\$ \mathcal{T}$ and compute $(\mathsf{pp}, \mathsf{crs}) \leftarrow \mathtt{Gen}(\lambda, N, \mathsf{td})$. For each $i \in [\![m]\!]$, compute $(\mathsf{idx}_i, \mathsf{aux}_i) \leftarrow \mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i)$. Assume $(\mathsf{pp}, \mathsf{idx}, u, w) \in \mathcal{R}$, and compute $(\mathsf{ed}, \mathsf{dd}) \leftarrow \mathtt{Digest}((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux})$. Finally compute $\mathsf{ct} \leftarrow \mathtt{Enc}(\mathsf{ed}, u, \mathsf{msg})$ and $\pi \leftarrow \mathtt{Prove}(\mathsf{dd}, u, w)$. Then*

$$\Pr[\mathtt{Dec}(\mathsf{ct}, \pi) = \mathsf{msg}] = 1.$$

2. ***Extractable security.*** *There exists a PPT straight-line extractor $\mathcal{E}$ such that, for every security parameter λ, polynomially bounded m, N, and PPT stateful adversary $\mathcal{A}$, the following holds. Sample $\mathsf{td} \leftarrow_\$ \mathcal{T}$ and compute $(\mathsf{pp}, \mathsf{crs}) \leftarrow \mathtt{Gen}(\lambda, N, \mathsf{td})$. Run $((s_1, \ldots, s_m), u, (\mathsf{msg}_0, \mathsf{msg}_1), \mathsf{tr}_1) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs})$, where every s_i is a valid input to $\mathtt{AuxGen}$ and $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathcal{M}$ have equal length. For each $i \in [\![m]\!]$, compute $(\mathsf{idx}_i, \mathsf{aux}_i) \leftarrow \mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i)$, and compute $(\mathsf{ed}, \mathsf{dd}) \leftarrow \mathtt{Digest}((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux})$. Sample $b \leftarrow_\$ \{0, 1\}$, compute $\mathsf{ct} \leftarrow \mathtt{Enc}(\mathsf{ed}, u, \mathsf{msg}_b)$, and run $(b', \mathsf{tr}_2) \leftarrow \mathcal{A}(\mathsf{ct})$. Define ϵ by $\Pr[b' = b] = 1/2 + \epsilon$, and set $w^* \leftarrow \mathcal{E}((\mathsf{pp}, \mathsf{crs}), \mathsf{idx}, \mathsf{aux}, u, (\mathsf{msg}_0, \mathsf{msg}_1), \mathsf{ct}, (\mathsf{tr}_1, \mathsf{tr}_2))$. Then*

$$\Pr[(\mathsf{pp}, \mathsf{idx}, u, w^*) \in \mathcal{R}] \geq \epsilon - \mathsf{negl}(\lambda).$$

Here $\mathsf{tr}_1, \mathsf{tr}_2$ are the challenger-recorded oracle queries and responses from $\mathcal{A}$'s two calls.

Definition 8 (Succinctness)**.** *For fixed security parameter and message length, a witness-encryption scheme is* succinct *if its ciphertext size is sublinear in the combined size of the index and witness. That is, $|\mathsf{ct}| = o(|\mathsf{idx}| + |w|)$.*

3.5.1 Witness Encryption Framework [GHK+25]

The framework of Garg et al. [GHK+25] combines linearly verifiable arguments of knowledge and compiles them into witness encryption. Its building blocks, called gadgets, are committed relations equipped with such arguments. We use it to combine authorization and non-membership for the same users.

A *routing* specifies which coordinates each gadget checks. Shared coordinates require the gadgets to agree on the corresponding values.

Definition 9 (Routed Product)**.** *Let $\mathcal{R}_1, \ldots, \mathcal{R}_k$ be indexed relations. A routing C specifies, for each relation i, coordinate sets P_i, I_i, J_i, W_i for its parameters, index, instance, and witness. The routed product $(\mathcal{R}_1 \times \cdots \times \mathcal{R}_k)_C$ contains precisely the tuples $(\mathsf{pp}, \mathsf{idx}, u, w)$ for which $(\mathsf{pp}|_{P_i}, \mathsf{idx}|_{I_i}, u|_{J_i}, w|_{W_i}) \in \mathcal{R}_i$ for every i. Different relations may share coordinates.*

The gadgets must also use the same setup, auxiliary generation, and commitment algorithms for shared coordinates. This is the intercompatibility requirement.

Definition 10 (Intercompatibility)**.** *Let Π_i be arguments of knowledge for $[\mathcal{R}_i]_{\mathsf{Com}_i}$, where $\mathsf{Com}_i = (\mathsf{CGen}_i, \mathsf{Commit}_i)$, with trapdoor spaces $\mathcal{T}_i$. They are C-intercompatible if there exist common tuples of algorithms $\mathtt{Gen}', \mathtt{AuxGen}', \mathsf{CGen}', \mathsf{Commit}'$, with trapdoor spaces $(\mathcal{T}'_j)_j$, such that each gadget uses the components selected by its routing, with $\mathtt{Gen}_i = \mathtt{Gen}'|_{P_i}$, $\mathcal{T}_i = \prod_{j \in P_i} \mathcal{T}'_j$, $\mathtt{AuxGen}_i = \mathtt{AuxGen}'|_{I_i}$, $\mathsf{CGen}_i = \mathsf{CGen}'|_{W_i}$, and $\mathsf{Commit}_i = \mathsf{Commit}'|_{W_i}$. Compatibility requires shared coordinates to use the same setup and commitment algorithms.*

Theorem 1 (Witness Encryption from Compatible Gadgets)**.** *Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be pairing-friendly groups over $\mathbb{F}_p$. Suppose $\mathcal{R}$ reduces to $(\mathcal{R}_1 \times \cdots \times \mathcal{R}_k)_C$. Suppose further that the committed relations admit C-intercompatible non-interactive arguments of knowledge with linear verification over $\mathrm{Hom}(U_i^{\ell_i}, \mathbb{G}_T^{\ell'_i})$, where $U_i \in \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{F}_p\}$.[9] Then, there exists a compiler (in particular, the one of [GHK+25, Construction 2]) that yields extractable witness encryption for $\mathcal{R}$ in the generic group and random oracle models. Furthermore, if the component arguments are succinct, the resulting witness-encryption scheme is succinct.*

[9] The input space $U_i^{\ell_i}$ may also be a product $\mathbb{G}_1^{a_i} \times \mathbb{G}_2^{b_i}$, for nonnegative integers a_i, b_i, as noted in [GHK+25, footnote 11].

Encryption and decryption. The compiler [GHK+25, Construction 2] constructs encryption and decryption algorithms from the combined verification equations. Both algorithms use "digests." The encryption digest contains the public data needed to verify the proof for each gadget. Together with an instance u, these data determine the linear system $A \circ \mathbf{z} = \mathbf{t}$ used by encryption. The decryption digest contains the corresponding public proving data. A party holding a witness uses this digest, together with the instance, to compute the witness commitments and component proofs that form $\mathbf{z}$. Shared proof components occur only once in $\mathbf{z}$.

To illustrate, suppose $A \in \mathbb{G}_1^{q \times \ell}$, $\mathbf{z} \in \mathbb{G}_2^{\ell}$, and $\mathbf{t} \in \mathbb{G}_T^{q}$. Encryption samples $\mathbf{s} \leftarrow_{\$} \mathbb{F}_p^q$ and outputs

$$\mathsf{ct}_1 = \mathbf{s}^\top A, \qquad \mathsf{ct}_2 = \mathsf{msg} + H(\mathbf{s}^\top \mathbf{t}),$$

where the random oracle H outputs masks in the message space (with XOR for bit strings). A party holding a valid witness uses the public decryption digest to form $\mathbf{z}$ and recovers

$$\mathsf{msg} = \mathsf{ct}_2 - H(\mathsf{ct}_1 \circ \mathbf{z}), \qquad \mathsf{ct}_1 \circ \mathbf{z} = \mathbf{s}^\top (A \circ \mathbf{z}) = \mathbf{s}^\top \mathbf{t}.$$

For the mixed source-group coordinates used by our gadgets, the same computation applies blockwise with the appropriate pairing, and Theorem 1 supplies extractable security under its stated hypotheses.

4 Nonmembership Gadgets

We construct gadgets which prove nonmembership in a "revoked set" L for one committed field element in Section 4.1 and for an indexed collection selected by a common vector in Section 4.2. Both gadgets satisfy the following hiding requirement:

Definition 11 (Perfect Hiding)**.** *Fix honestly generated public parameters and a number n of index entries. A gadget with private inputs $(y_i, r_i)_{i \in [\![n]\!]}$ is* perfectly hiding *if, for any two vectors $\boldsymbol{y}^{(0)}, \boldsymbol{y}^{(1)} \in \mathbb{F}_p^n$, the following public views have identical distributions. For $b \in \{0,1\}$, independently sample $r_i \leftarrow_{\$} \mathbb{F}_p$ and generate the public indices, auxiliary data, and digests from $(y_i^{(b)}, r_i)_{i \in [\![n]\!]}$.*

Below, for a a revoked set $L \subseteq \mathbb{F}_p$, we write $P_L(Y) := \prod_{z \in L}(Y - z)$ and $P_\emptyset(Y) := 1$.

4.1 Nonmembership Gadget

Definition 12 (Nonmembership relation)**.** *Let $\mathsf{Com}(y; r)$ be a commitment to y with randomness r under commitment key ck, which we leave implicit. Define*

$$\mathsf{NM} = \left\{ \left(\begin{array}{l} \textit{parameters } \mathsf{ck}, \\ \textit{index } c, \\ \textit{instance } L, \\ \textit{witness } (y, r) \end{array} \right) \;\middle|\; \begin{array}{l} c = \mathsf{Com}(y; r), \\ y \notin L \end{array} \right\},$$

where $y, r \in \mathbb{F}_p$ and $L \subseteq \mathbb{F}_p$ with $|L| \leq n$.

Construction 2 (Non-membership gadget)**.** *We instantiate the commitment as*

$$\mathsf{Com}(y; r) = [\theta]_1 - y[1]_1 + r[\alpha]_1,$$

where $\mathsf{ck} = (\mathcal{BG}, [\theta]_1, [\alpha]_1)$. The witness encryption algorithms are instantiated as follows

- $\mathtt{Gen}(\lambda, n, \mathsf{td})$*: Sample bilinear groups and $\mathsf{td} = (\theta, \alpha) \leftarrow (\mathbb{F}_p^\times)^2$. Deterministically output*

$$\mathsf{pp} \leftarrow (\mathcal{BG}, n, [\theta]_1, [\alpha]_1, ([\theta^j]_1)_{j=0}^n),$$
$$\mathsf{crs} \leftarrow ([\theta^j]_1, [\theta^j]_2)_{j=0}^n.$$

  Erase td*.* pp *is implicit input to all preceding algorithms.*

- $\texttt{AuxGen}(\mathsf{crs}, (y, r))$*: Compute* $\mathsf{idx} \leftarrow \mathsf{Com}(y; r)$ *and output* $(\mathsf{idx}, \perp)$.
- $\texttt{Digest}(\mathsf{crs}, \mathsf{idx}, \mathsf{aux})$*: Output*

$$\mathsf{Vk} \leftarrow (\mathsf{idx}, [\theta]_1, [\alpha]_1, [1]_1, [1]_2, [1]_T),$$
$$\mathsf{Pk} \leftarrow (\mathsf{crs}, \mathsf{idx}, \mathsf{aux}).$$

- $\texttt{Prove}(\mathsf{Pk}, (\overline{c}, L), (y, r))$*: For a valid opening with* $\overline{c} = \mathsf{idx}$ *and* $y \notin L$*, compute* $v = P_L(y)^{-1}$*,* $u(Y) = \frac{1 - P_L(Y)v}{Y - y}$*, and* $w(Y) = ru(Y)$*, and output*

$$\pi = (\overline{u}, \overline{v}, \overline{w}, \overline{y}, \overline{r}) = ([u(\theta)]_2, [v]_2, [w(\theta)]_2, [y]_2, [r]_2).$$

- $\texttt{Verify}(\mathsf{Vk}, (\overline{c}, L), \pi)$*: Parse* $\pi = (\overline{u}, \overline{v}, \overline{w}, \overline{y}, \overline{r})$*. Compute* $[P_L(\theta)]_1$ *from* L *and* $([\theta^j]_1)_{j=0}^n$ *in* pp*. Check*

  1. $\overline{c} \circ [1]_2 = \mathsf{idx} \circ [1]_2$
  2. $\mathsf{idx} \circ \overline{u} - [\alpha]_1 \circ \overline{w} + [P_L(\theta)]_1 \circ \overline{v} = [1]_T$
  3. $[1]_1 \circ \overline{y} - [\alpha]_1 \circ \overline{r} = ([\theta]_1 - \mathsf{idx}) \circ [1]_2$.

  Accept exactly when all three checks hold.

For the framework, regard the same map $\mathsf{Com}(y; r)$ as a deterministic commitment to the entire pair (y, r), with no additional randomness, and denote this scheme by C_{NM}. The committed relation $[\mathsf{NM}]_{\mathsf{C}_{\mathsf{NM}}}$ has public instance $(\overline{c}, L)$ and witness (y, r), where $\overline{c} = \mathsf{Com}(y; r)$. It requires $\overline{c} = c$ and $y \notin L$.

Lemma 2 (Nonmembership Gadget)**.** *Construction 2 is a linearly verifiable constant-size non-interactive argument of knowledge for* $[\mathsf{NM}]_{\mathsf{C}_{\mathsf{NM}}}$ *in the generic bilinear-group model.*

Proof. For perfect completeness, take any valid opening (y, r) of c, including an opening different from the one used by $\texttt{AuxGen}$. Observe its polynomial $p(Y, Z) = Y - y + rZ$ satisfies $[p(\theta, \alpha)]_1 = c$. Moreover, the prover's polynomials satisfy

$$p(Y, Z)u(Y) - Zw(Y) + P_L(Y)v = 1. \tag{2}$$

Evaluating this identity at (θ, α) gives check 2. Check 3 follows directly from $c = [\theta - y + \alpha r]_1$. Check 1 holds by definition of the committed relation. Therefore the construction is perfectly complete.

For linear verifiability, note all checks are linear in the supplied witness commitment and proof elements. The committed witness is one element of $\mathbb{G}_1$, and the proof is five elements of $\mathbb{G}_2$, so the argument has constant size and is therefore succinct.

The deterministic commitment to (y, r) is computationally binding in this model. Indeed, two distinct pairs with the same encoding would give $y - y' = \alpha(r - r')$, where necessarily $r \neq r'$, revealing the honestly sampled hidden scalar α. The public powers of θ do not reveal α in the generic model, except with negligible collision probability.

For knowledge soundness, the public source-$\mathbb{G}_2$ inputs are $\{[\theta^j]_2\}_{j=0}^n$. The extractor obtains polynomials $u(Y), v(Y), w(Y), \eta(Y), \rho(Y) \in \mathbb{F}_p[Y]_{\leq n}$ from the recorded oracle queries, representing the five proof elements (where $\eta(Y)$ and $\rho(Y)$ represent $\overline{y}$ and $\overline{r}$, respectively). Note the extracted $v(Y)$ need not be constant, though an honest $v(Y)$ is. Write (y_0, r_0) for the inputs to $\texttt{AuxGen}$. Outside a negligible generic-group collision event, check 3 gives

$$\eta(Y) - Z\rho(Y) = y_0 - r_0 Z.$$

Comparing coefficients shows that $\eta(Y) = y_0$ and $\rho(Y) = r_0$ are constant. The extractor outputs $(y, r) = (\eta(0), \rho(0))$. Check 2 gives

$$(Y - y + rZ)u(Y) - Zw(Y) + P_L(Y)v(Y) = 1.$$

Setting $Z = 0$ and $Y = y$ yields $P_L(y)v(y) = 1$, hence $y \notin L$, including when $L = \emptyset$. Checks 3 and 1 show that (y, r) opens both c and $\overline{c}$. Thus it is a valid witness for $[\mathsf{NM}]_{\mathsf{C}_{\mathsf{NM}}}$. Extraction is polynomial time and requires no rewinding. □

Lemma 3 (Perfect hiding)**.** *Construction 2 satisfies perfect hiding as defined by Definition 11.*

Proof. The index is perfectly hiding by the standard Pedersen commitment argument [Ped91]: for fixed setup, uniform r makes $\theta - y + \alpha r$ uniform over $\mathbb{F}_p$, since $\alpha \neq 0$. The auxiliary output is empty, and the digests are deterministic functions of the public setup and index, so their joint distribution is also independent of y. Independent masks give the same guarantee jointly for any collection of indices. □

4.2 Indexed Non-membership Gadget

We now consider a vector of commitments to revocation pseudonyms. A selector $b \in \mathbb{F}_p^n$ identifies the users whose nonmembership must be proved. Specifically, every coordinate with $b_i \neq 0$ must have a known opening to a pseudonym outside L. Coordinates with $b_i = 0$ require no opening, and their witness entries are unrestricted.

Definition 13 (Indexed nonmembership relation)**.** *Let* $\mathsf{Com}(y; r)$ *be a commitment to a field element* $y \in \mathbb{F}_p$ *using randomness* $r \in \mathbb{F}_p$*. Define*

$$\mathsf{INM} = \left\{ \left(\begin{array}{l} \textit{parameters } \mathsf{ck}, \\ \textit{index } (c_1, \ldots, c_n), \\ \textit{instance } L, \\ \textit{witness } (b, y, r) \end{array} \right) \;\middle|\; \begin{array}{l} \forall i \in [\![n]\!], \\ b_i \neq 0 \implies \big(\mathsf{Com}(y_i; r_i) = c_i \;\wedge\; y_i \notin L\big) \end{array} \right\}.$$

Here $b, y, r \in \mathbb{F}_p^n$ *and* $L \subseteq \mathbb{F}_p$ *with* $|L| \leq n$ *.*

In our construction, the opening check involves the products $b_i y_i$ and $b_i r_i$. We use these products as witness coordinates so that the check is linear in the committed witness:

Definition 14 (Weighted indexed nonmembership relation)**.** *Define* $\mathsf{INM}^{\mathsf{wt}}$ *by replacing each witness* (b, y, r) *for* INM *with* (b, t, s)*, where* $t_i = b_i y_i$*, and* $s_i = b_i r_i$*. The public parameters, index, and instance are unchanged. In particular, every witness for* $\mathsf{INM}^{\mathsf{wt}}$ *has* $t_i = s_i = 0$ *whenever* $b_i = 0$*.*

Conversely, from any witness (b, t, s) *for* $\mathsf{INM}^{\mathsf{wt}}$*, we obtain a witness* (b, y, r) *for* INM *by setting*

$$(y_i, r_i) = \begin{cases} (t_i/b_i, s_i/b_i), & b_i \neq 0, \\ (0, 0), & b_i = 0. \end{cases}$$

Construction 3 (Indexed Nonmembership Gadget)**.** *Fix bilinear groups* $\mathcal{BG}$ *and trapdoor* $\mathsf{td} = (\tau, \theta, \alpha)$*, where* $\tau \leftarrow_{\$} \mathbb{F}_p$ *and* $\theta, \alpha \leftarrow_{\$} \mathbb{F}_p^{\times}$*. Instantiate the commitment as* $\mathsf{ck} = ([\theta]_1, [\alpha]_1)$ *and* $\mathsf{Com}(y; r) = [\theta - y + \alpha r]_1$*.*

Write $\mathsf{idx}_i = c_i$ *and, for the openings used in honest registration, let* $p_i(Y, Z) = Y - y_i + r_i Z$*. As before,* Y, Z *are formal variables evaluated at* θ, α*. A third variable* X*, evaluated at* τ*, indexes the users. Fix a public domain* $\mathcal{D} = (d_1, \ldots, d_n) \subseteq \mathbb{F}_p$ *of distinct elements, its Lagrange basis polynomials* $L_i(X)$*, and its vanishing polynomial* $Z_{\mathcal{D}}(X)$*. For* $b, t, s \in \mathbb{F}_p^n$ *as in Definition 14, write*

$$B(X) = \sum_i b_i L_i(X), \qquad T(X) = \sum_i t_i L_i(X), \qquad S(X) = \sum_i s_i L_i(X),$$

Let $\tilde{p}(X, Y, Z) = \sum_i p_i(Y, Z) L_i(X)$ *and define* Q_i *by*

$$Z_{\mathcal{D}}(X) Q_i(X, Y, Z) = (\tilde{p}(X, Y, Z) - p_i(Y, Z)) L_i(X).$$

Use the encodings $(\overline{B}, \overline{T}, \overline{S}) = ([B(\tau)]_2, [T(\tau)]_2, [S(\tau)]_2)$ *below.*

- $\mathsf{Gen}(\lambda, n, \mathsf{td})$*: Given* $\mathsf{td} = (\tau, \theta, \alpha)$*, deterministically output*

$$\begin{aligned} \mathsf{pp} &\leftarrow (\mathcal{BG}, n, \mathcal{D}, [\theta]_1, [\alpha]_1, ([\theta^j]_1)_{j=0}^n), \\ \mathsf{crs} &\leftarrow \big([\tau^a \theta^b]_1, [\alpha \tau^a \theta^b]_1, [\tau^a \theta^b]_2\big)_{0 \leq a, b \leq n}. \end{aligned}$$

- $\texttt{AuxGen}(\mathsf{crs}, (y_i, r_i))$ *: Using homomorphic operations over* crs*, output*

$$\begin{aligned}\mathsf{idx}_i &\leftarrow [p_i(\theta,\alpha)]_1,\\ \mathsf{aux}_i &\leftarrow \left([p_i(\theta,\alpha)\tau^j\theta^k]_1\right)_{0\le j,k<n}.\end{aligned}$$

- $\texttt{Digest}(\mathsf{crs}, \mathsf{idx}, \mathsf{aux})$*: For each* $i \in [\![n]\!]$*, let* $Q_i(X,Y,Z)$ *be the unique polynomial satisfying*

$$Z_{\mathcal{D}}(X)Q_i(X,Y,Z) = (\tilde{p}(X,Y,Z) - p_i(Y,Z))L_i(X).$$

The difference $\tilde{p}(X,Y,Z) - p_i(Y,Z)$ *is independent of* Y*, so* Q_i *is also independent of* Y *. Using* aux*, homomorphically compute* $\widehat{\mathsf{aux}}_{i,h} := [\theta^h Q_i(\tau,\theta,\alpha)]_1$ *for all* $i \in [\![n]\!]$ *and* $0 \le h < n$*. Write* $\widehat{\mathsf{aux}} = (\widehat{\mathsf{aux}}_{i,h})_{i,h}$*.*

Using crs*, compute helper keys*

$$\mathsf{hk}_i := \left([L_i(\tau)\theta^c\tau^d]_2\right)_{0\le c\le n,\ 0\le d\le 1}.$$

Output

$$\begin{aligned}\mathsf{Vk} &\leftarrow \left([\tilde{p}(\tau,\theta,\alpha)]_1, [\alpha]_1, [1]_1, [Z_{\mathcal{D}}(\tau)]_2, [\tau]_1, [\theta^n]_1, [\theta]_1\right),\\ \mathsf{Pk} &\leftarrow (\mathsf{crs}, \mathsf{idx}, \widehat{\mathsf{aux}}, (\mathsf{hk}_i)_{i\in[\![n]\!]}).\end{aligned}$$

- $\texttt{Prove}(\mathsf{Pk}, ((\overline{B},\overline{T},\overline{S}), L), (b,t,s))$*: Given a valid witness whose commitment is* $(\overline{B},\overline{T},\overline{S})$*, recover* $(y_i, r_i) = (t_i/b_i, s_i/b_i)$ *whenever* $b_i \neq 0$*, and set* $y_i = r_i = 0$ *otherwise. For each selected coordinate, set*

$$v_i = b_i P_L(y_i)^{-1}, \qquad u_i(Y) = \frac{b_i - P_L(Y)v_i}{Y - y_i}, \qquad w_i(Y) = r_i u_i(Y).$$

For each unselected coordinate, set $u_i(Y) = v_i = w_i(Y) = 0$*. Let*

$$U(X,Y) = \sum_i u_i(Y)L_i(X), \quad V(X) = \sum_i v_i L_i(X), \quad W(X,Y) = \sum_i w_i(Y)L_i(X).$$

Form $\overline{U} = [U(\tau,\theta)]_2$*,* $\overline{V} = [V(\tau)]_2$*, and* $\overline{W} = [W(\tau,\theta)]_2$*. Compute the quotient encodings from the retained auxiliary data using*

$$\begin{aligned}Q_{\mathrm{nm}}(X,Y,Z) &= \sum_i u_i(Y)Q_i(X,Y,Z),\\ Q_{\mathrm{open}}(X,Y,Z) &= \sum_i b_i Q_i(X,Y,Z).\end{aligned} \tag{3}$$

For $F \in \{B,T,S,U,V,W\}$*, compute* $\widehat{F}_X(X,Y) = XF(X,Y)$*. Also compute*

$$\widehat{F}_Y(X,Y) = \begin{cases} Y^n F(X), & F \in \{B,V\},\\ YF(X,Y), & F \in \{U,W\}.\end{cases}$$

Here a univariate polynomial is regarded as independent of Y*. Using* Pk*, output*

$$\pi \leftarrow \begin{pmatrix} \overline{U},\overline{V},\overline{W}, [Q_{\mathrm{nm}}(\tau,\theta,\alpha)]_1, [Q_{\mathrm{open}}(\tau,\theta,\alpha)]_1,\\ ([\widehat{F}_X(\tau,\theta)]_2)_{F\in\{B,T,S,U,V,W\}},\\ ([\widehat{F}_Y(\tau,\theta)]_2)_{F\in\{B,V,U,W\}}\end{pmatrix}.$$

- $\texttt{Verify}(\mathsf{Vk}, ((\overline{B}, \overline{T}, \overline{S}), L), \pi)$: *Parse the proof as*

$$\pi = \big(\overline{U}, \overline{V}, \overline{W}, \overline{Q}_{\mathrm{nm}}, \overline{Q}_{\mathrm{open}}, (\overline{F}_X)_{F\in\{B,T,S,U,V,W\}}, (\overline{F}_Y)_{F\in\{B,V,U,W\}}\big).$$

Set $\overline{p} = [\tilde{p}(\tau,\theta,\alpha)]_1$ *from* Vk *. Compute* $\overline{P}_L = [P_L(\theta)]_1$ *from* L *and* $([\theta^j]_1)_{j=0}^n$ *in* pp*. Check*

1. $\overline{p} \circ \overline{U} - [\alpha]_1 \circ \overline{W} + \overline{P}_L \circ \overline{V} = [1]_1 \circ \overline{B} + \overline{Q}_{\mathrm{nm}} \circ [Z_{\mathcal{D}}(\tau)]_2$.
2. $(\overline{p} - [\theta]_1) \circ \overline{B} + [1]_1 \circ \overline{T} - [\alpha]_1 \circ \overline{S} = \overline{Q}_{\mathrm{open}} \circ [Z_{\mathcal{D}}(\tau)]_2$.
3. $[\tau]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_X$ *for* $F \in \{B, T, S, U, V, W\}$.
4. $[\theta^n]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_Y$ *for* $F \in \{B, V\}$.
5. $[\theta]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_Y$ *for* $F \in \{U, W\}$.

For the framework, define

$$\begin{aligned}\texttt{Commit}_{\mathsf{INM}}(b,t,s) &:= \big(\texttt{Commit}_{\mathsf{SCS}(\mathbb{G}_2)}(b), \texttt{Commit}_{\mathsf{SCS}(\mathbb{G}_2)}(t), \texttt{Commit}_{\mathsf{SCS}(\mathbb{G}_2)}(s)\big)\\ &= ([B(\tau)]_2, [T(\tau)]_2, [S(\tau)]_2),\end{aligned}$$

with the public commitment key from setup left implicit, and denote this scheme by $\mathsf{C}_{\mathsf{INM}}$. Since it is a product of three standard structured commitments, it inherits their computational binding. The committed relation $[\mathsf{INM}^{\mathsf{wt}}]_{\mathsf{C}_{\mathsf{INM}}}$ has public instance $((\overline{B}, \overline{T}, \overline{S}), L)$ and witness (b, t, s). Its first commitment is the standard selector commitment shared with authorization gadgets.

The auxiliary data aux contain $O(n^3)$ group elements, while $\widehat{\mathsf{aux}}$ contains $O(n^2)$ and is what the prover key uses. The $O(n^3)$ aux may be discarded after digest generation for a fixed registry. Both quotients use only the retained entries, for Q_{open} uses $h = 0$ and Q_{nm} uses $0 \le h < n$ because $\deg_Y u_i < n$.

The displayed construction uses CRS bound n. For a reusable CRS with larger bound $N \ge n$, replace every X-shift by X^{N-n+1}, the Y-shifts for B, V by Y^N, and those for U, W by Y^{N-n+1}. The verifier uses the corresponding encodings $[\tau^{N-n+1}]_1$, $[\theta^N]_1$, and $[\theta^{N-n+1}]_1$. The helper keys are shifted accordingly, and all required monomials remain within the range $0 \le a, b \le N$. Thus the degree bounds refer to the current domain size n even when the CRS is reused.

To allow later registrations without storing every user's auxiliary data, we update the existing digest using only the new user's auxiliary data:

Remark 1 (Incremental registration)**.** *Fix the interpolation domain of Construction 3 to all M slots, with $n = M$, and initialize* $\texttt{Digest}$ *with $p_i(Y, Z) = Y$ at every slot.*

After validating a new user for an unused slot j, replace its registered index and write $\Delta p_j(Y, Z) = p_j(Y, Z) - Y$. By equation (1) of Lemma 1, the corresponding polynomial updates are

$$\begin{aligned}\tilde{p}(X,Y,Z) &\leftarrow \tilde{p}(X,Y,Z) + L_j(X)\Delta p_j(Y,Z),\\ Q_i(X,Y,Z) &\leftarrow Q_i(X,Y,Z) + \frac{(L_j(X) - \delta_{ij})L_i(X)}{Z_{\mathcal{D}}(X)}\Delta p_j(Y,Z).\end{aligned}$$

Here δ_{ij} is 1 if $i = j$ and 0 otherwise. The displayed quotient is a polynomial because its numerator vanishes on $\mathcal{D}$. Update the stored encodings of these polynomials homomorphically, using

$$[\tau^a\theta^h \Delta p_j(\theta,\alpha)]_1 = [\tau^a\theta^h p_j(\theta,\alpha)]_1 - [\tau^a\theta^{h+1}]_1, \qquad 0 \le a, h < M.$$

The first term is supplied in the new user's auxiliary data, and the second is available in the CRS. The new user's auxiliary data can then be discarded. When an application also needs $C_h = [\theta^h \tilde{p}(\tau,\theta,\alpha)]_1$, maintain these encodings using the first update above, starting from $C_h = [\theta^{h+1}]_1$. The authorization gadget must force $b_i = 0$ at unused slots.

This retains $O(M^2)$ group elements, including the fixed helper keys, rather than all users' $O(M^3)$ auxiliary data at full capacity. A direct update takes $O(M^3)$ group operations.

Lemma 4. *Construction 3 is perfectly complete.*

Proof. Fix a witness (b, t, s) for $\mathsf{INM}^{\mathsf{wt}}$ and recover a witness (b, y, r) for INM as in Definition 14. For a selected coordinate, $P_L(y_i) \neq 0$, so the prover's polynomials satisfy

$$(Y - y_i + r_i Z)u_i(Y) - Z w_i(Y) + P_L(Y)v_i = b_i.$$

Also $t_i = b_i y_i$ and $s_i = b_i r_i$ give

$$(Y - y_i + r_i Z - Y)b_i + t_i - Z s_i = 0.$$

The witness may use any valid openings. For a selected coordinate, equality of its commitment with the registered index implies

$$-y_i + \alpha r_i = -y_i^{\mathrm{reg}} + \alpha r_i^{\mathrm{reg}},$$

where the superscript reg marks the opening used during registration, while (y_i, r_i) denotes the opening supplied in the proof's witness. Hence the witness polynomial $Y - y_i + r_i\alpha$ equals the registered $p_i(Y, \alpha)$. Both displayed identities therefore hold with the registered polynomial after setting $Z = \alpha$. At an unselected coordinate they also hold, because $b_i = t_i = s_i = u_i(Y) = v_i = w_i(Y) = 0$, regardless of its registered opening. Multiplying the local identities by $L_i(X)$, summing, and using the registered helper polynomials $Q_i(X, Y, Z)$ gives the following identities at $Z = \alpha$, with the quotients defined in equations (3). The first expresses weighted opening consistency and the second expresses nonmembership:

$$(\tilde{p}(X, Y, Z) - Y)B(X) + T(X) - Z S(X) = Z_{\mathcal{D}}(X) Q_{\mathrm{open}}(X, Y, Z). \tag{4}$$

$$\tilde{p}(X, Y, Z)U(X, Y) - Z W(X, Y) + P_L(Y)V(X) = B(X) + Z_{\mathcal{D}}(X) Q_{\mathrm{nm}}(X, Y, Z). \tag{5}$$

Here $U(X, Y), V(X), W(X, Y)$ and the two quotients are computed by `Prove` from the witness (b, t, s). Thus the retained auxiliary encodings suffice even when the witness uses different valid openings.

Since $\deg_Y u_i < n$, all encodings of $u_i(Y)Q_i(X, Y, Z)$ are linear combinations of the entries with $h < n$. The opening quotient uses only $h = 0$. All six polynomials have X-degree below n, B, T, S, V are independent of Y, and U, W have Y-degree below n, so their shifted encodings are available from the helper keys. Evaluating the two identities at $(X, Y, Z) = (\tau, \theta, \alpha)$ and applying bilinearity proves checks 1 and 2. The definitions of the shifts prove checks 3–5. These are the maximum-degree checks of [GHK+25]. Appendix A establishes their compatibility with the extended CRS. □

Theorem 2. *Construction 3 is a linearly verifiable succinct non-interactive argument of knowledge for* $[\mathsf{INM}^{\mathsf{wt}}]_{\mathsf{C}_{\mathsf{INM}}}$. *The conversion between witnesses in Definition 14 gives completeness and extraction for the original relation* INM.

Proof. Completeness follows by Lemma 4. The verification equations are linear in the committed witness and proof elements. Their number and the proof size are constant, while the witness contains $3n$ field elements, giving succinctness.

For knowledge soundness, use the independent, honestly sampled setup scalars and the prover's full oracle transcript. Write X, Y, Z for the formal variables corresponding to τ, θ, α. In the displayed construction every $\mathbb{G}_2$ input is a known scalar combination of $[\tau^a\theta^b]_2$ for $0 \leq a, b \leq n$. Registration and `AuxGen` supply only $\mathbb{G}_1$ elements. Generic group operations preserve this form, and pairings output only in $\mathbb{G}_T$. In any gadget composition, the polynomial parts of additional $\mathbb{G}_2$ inputs must likewise lie in $\mathbb{F}_p[X, Y]$ with known coefficients and individual degrees at most the CRS bound. Additional independent $\mathbb{G}_2$ elements , such as those supplied by the signature gadget, have no copies multiplied by the required trapdoor powers. Their shifted terms lie outside the public $\mathbb{G}_2$ span, so the shift checks force their coefficients in the six unshifted encodings to vanish.

Outside negligible generic-group collision, the coefficientwise degree argument ofs [GHK+25, Lemma 3] therefore gives $\deg_X F < n$ for $F \in \{B, T, S, U, V, W\}$ from check 3. Check 4 makes B, V independent of Y, and check 5 gives $\deg_Y U, \deg_Y W < n$. With a larger CRS bound N, the shifts X^{N-n+1}, Y^N, and Y^{N-n+1}

give exactly the same conclusions. All these polynomials are independent of Z, since the $\mathbb{G}_2$ CRS has no Z terms. The extractor obtains their coefficients from the recorded generic operations.

Recall $y_i^{\mathrm{reg}}, r_i^{\mathrm{reg}}$ denote the openings used in honest registration. Checks 1 and 2 imply the formal polynomial identities (5) and (4), with $p_i(Y,Z) = Y - y_i^{\mathrm{reg}} + r_i^{\mathrm{reg}} Z$. Let $b_i = B(d_i)$. Evaluating the opening identity at $X = d_i$ gives

$$(-y_i^{\mathrm{reg}} + r_i^{\mathrm{reg}} Z) b_i + T(d_i, Y) - Z S(d_i, Y) = 0.$$

Comparing coefficients of Z and its constant term yields $S(d_i, Y) = b_i r_i^{\mathrm{reg}}$ and $T(d_i, Y) = b_i y_i^{\mathrm{reg}}$. In particular, both values are independent of Y. Because T, S have X-degree below n, interpolation on the n domain points shows that $T, S \in \mathbb{F}_p[X]_{<n}$. This is why separate Y-degree checks for these two polynomials are unnecessary.

The extractor sets $t_i = T(d_i)$ and $s_i = S(d_i)$. Evaluating the nonmembership identity at $X = d_i$ gives

$$(Y - y_i^{\mathrm{reg}} + r_i^{\mathrm{reg}} Z) U(d_i, Y) - Z W(d_i, Y) + P_L(Y) V(d_i) = b_i. \tag{6}$$

Setting $Y = y_i^{\mathrm{reg}}$ and $Z = 0$ yields $P_L(y_i^{\mathrm{reg}}) V(d_i) = b_i$. If $b_i \neq 0$, then $y_i^{\mathrm{reg}} \notin L$, and the extractor recovers the actual opening as $y_i = t_i / b_i = y_i^{\mathrm{reg}}$ and $r_i = s_i / b_i = r_i^{\mathrm{reg}}$.

Thus $\mathsf{Com}(y_i; r_i) = \mathsf{idx}_i$. If $b_i = 0$, the opening identity gives $t_i = s_i = 0$, and we set $y_i = r_i = 0$ as in Definition 14. Note the extractor need not know the registration openings in advance, as it computes only the coefficients of B, T, S and these ratios.

Finally, the X-degree bounds show that B, T, S are the interpolants of the extracted vectors. Hence

$$(\overline{B}, \overline{T}, \overline{S}) = \mathtt{Commit}_{\mathsf{INM}}(b, t, s).$$

The extractor outputs a valid witness for $[\mathsf{INM}^{\mathsf{wt}}]_{\mathsf{C}_{\mathsf{INM}}}$ and recovers a witness for INM as in Definition 14, completing the proof. □

Lemma 5 (Perfect hiding of the indexed gadget)**.** *Construction 3 satisfies perfect hiding as defined in Definition 11.*

Proof. Fix the setup scalars (τ, θ, α) and any pseudonym vector $(y_i)_i$. Since $\alpha \neq 0$, the independent uniform masks r_i make $z_i = -y_i + \alpha r_i$ independent and uniform over $\mathbb{F}_p$, regardless of $(y_i)_i$. The indices $[\theta + z_i]_1$ and auxiliary entries $[(\theta + z_i)\tau^j \theta^k]_1$ depend on the openings only through these z_i. The digests are deterministic functions of the setup and these public data. Thus the joint distribution of all indices, auxiliary data, and digests is identical for every pseudonym vector. □

We recall the Hard-to-Invert AuxGen property of Garg et al. [GHK+25, Definition 15]. In our construction, the secret input to $\mathtt{AuxGen}$ is $s_i = (y_i, r_i) \in \mathbb{F}_p^2$. We use the following game.

Definition 15 (Hardness of $\mathtt{AuxGen}$)**.** *Let* $\mathsf{WE} = (\mathtt{Gen}, \mathtt{AuxGen}, \mathtt{Digest}, \mathtt{Enc}, \mathtt{Prove}, \mathtt{Dec})$ *be a Witness Encryption scheme. Consider the following game between a PPT adversary* $\mathcal{A}$ *and a challenger* $\mathcal{C}$*.*

1. *$\mathcal{C}$ samples the public setup and* td *honestly, computes* $(\mathsf{pp}, \mathsf{crs}) \leftarrow \mathtt{Gen}(\lambda, N, \mathsf{td})$*, and sends* $(\mathsf{pp}, \mathsf{crs})$ *to* $\mathcal{A}$*. $\mathcal{C}$ also initializes a set of inverted indices* $\mathsf{Cor} \leftarrow \emptyset$ *and a counter* $K \leftarrow 0$*.*
2. *$\mathcal{A}$ makes polynomially many queries of the following types, choosing each after seeing the previous responses.*
   (a) $\mathtt{AuxGen}$ **query.** *$\mathcal{C}$ sets* $i \leftarrow K+1$ *and samples a uniformly random* s_i*, runs* $(\mathsf{idx}_i, \mathsf{aux}_i) \leftarrow \mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i)$ *and sends* $(\mathsf{idx}_i, \mathsf{aux}_i)$ *to* $\mathcal{A}$*. $\mathcal{C}$ also updates* $K \leftarrow K + 1$*.*
   (b) **Inversion query** *$\mathcal{A}$ chooses an index* i *and sends it to* $\mathcal{C}$*. If* $i \in [\![K]\!]$*, $\mathcal{C}$ updates* $\mathsf{Cor} \leftarrow \mathsf{Cor} \cup \{i\}$ *and sends* s_i *to* $\mathcal{A}$*.*
3. *The adversary outputs* (i, s_i^*)*.*

We say that WE *has a Hard-to-Invert* AuxGen *if for any PPT adversary* $\Pr[i \in [\![K]\!] \setminus \mathsf{Cor} \wedge \mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), s_i^*) = (\mathsf{idx}_i, \mathsf{aux}_i)] = \mathsf{negl}(\lambda)$.

Lemma 6 (Hardness of INM AuxGen)**.** *Construction 3 has a Hard-to-Invert* AuxGen *algorithm against generic adversaries.*

Proof. Let X, Y, Z be formal variables corresponding to τ, θ, α respectively. For fixed α, the map $(y_i, r_i) \mapsto (-y_i + r_i\alpha, r_i)$ is a bijection on $\mathbb{F}_p^2$. Hence for every AuxGen query i, the uniformly random scalar $-y_i + r_i\alpha$ may be represented by a fresh formal variable S_i.

All public outputs for user i depend on (y_i, r_i) only through S_i. In a symbolic simulation, process queries sequentially. On the first inversion query for i, sample $(y_i, r_i) \leftarrow_{\$} \mathbb{F}_p^2$ and substitute $S_i = -y_i + r_i Z$ in every recorded expression, keeping the existing handles, so repeated queries return the same pair. The bijection above couples this deferred sampling to honest registration, except when distinct expressions collide under evaluation or substitution. For polynomially many queries this event is negligible, and unrevealed S_i remain independent in the simulation.

At the end of step 2 the adversary's view in the symbolic GGM is generated by the CRS together with

$$\begin{aligned}&\big\{(y_i, r_i), [Y - y_i + r_i Z]_1, ([(Y - y_i + r_i Z)X^j Y^k]_1)_{0 \le j,k < n}\big\}\\ &\quad \text{for each } i \in \mathsf{Cor},\\ &\big\{[Y + S_i]_1, ([(Y + S_i)X^j Y^k]_1)_{0 \le j,k < n}\big\}\\ &\quad \text{for each } i \in [\![K]\!] \setminus \mathsf{Cor}.\end{aligned}$$

The digests are deterministic public functions of these data. Suppose the adversary outputs $(i, (y_i^*, r_i^*))$ for some $i \in [\![K]\!] \setminus \mathsf{Cor}$. For this to be a valid preimage, the corresponding indices must agree, and hence it must be that $[Y - y_i^* + r_i^* Z]_1 = [Y + S_i]_1$. This requires $S_i = -y_i^* + r_i^* Z$. For an uncorrupted target, S_i is independent of the proposed scalars in the symbolic simulation, so this final equality holds with probability $1/|\mathbb{F}_p|$. Adding the negligible collision probability gives negligible success in the real experiment. □

Efficiency The gadget uses $O(n^2)$ CRS elements. The auxiliary data aux_i contain $O(n^2)$ group elements per user , and Digest compacts their $O(n^3)$ total size into $O(n^2)$ prover data. The committed witness contains three $\mathbb{G}_2$ elements. The additional proof contains thirteen $\mathbb{G}_2$ elements and two $\mathbb{G}_1$ elements, for a total of sixteen $\mathbb{G}_2$ and two $\mathbb{G}_1$ elements. Verification checks twelve pairing equalities.

Forming the coefficients of P_L from the raw list takes $O(|L|^2+1)$ field operations by direct multiplication. Given these coefficients, its CRS encoding takes $O(|L| + 1)$ group operations. For one selected coordinate, computing the Bézout coefficients and their encodings takes $O(|L| + 1)$ field and group operations using the prepared helper row. We count these costs separately from setup, auxiliary generation, and digest generation, which depend on the domain size. The public list and recipient or policy descriptions are not included in the constant-size proof and ciphertext-component bounds.

5 Adding Revocation to Arbitrary Public-Key Encryption

We add revocation to IND-CPA-secure PKE by secret-sharing the message. Each registered public key pairs an ordinary PKE key with a pseudonym commitment. We encrypt one share under the ordinary key and the other with non-membership witness encryption for that same commitment and the revoked set. An unrevoked recipient recovers both shares and adds them to recover the message.

This construction addresses one registered recipient. The proof uses PKE security when that recipient is unrevoked and non-membership security when it is revoked, even if its secret key is exposed. We give the definition and construction below, including the compatibility conditions needed when the PKE scheme uses the frameworks bilinear groups (so that even with these extra elements, an adversary cannot learn anything more).

5.1 Definition

Definition 16 (Revocable PKE)**.** *A revocable PKE scheme with message space $\mathcal{M}$ and pseudonym space $\mathcal{Y}$ has PPT algorithms $\Sigma = (\mathtt{Setup}, \mathtt{KGen}, \mathtt{Reg}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Revoke})$ with the following syntax.*

1. $\mathtt{Setup}(1^\lambda, n) \to \mathsf{pp}$*: On input security parameter λ and maximum number of users n, output public parameters* pp *and initialize $L = \emptyset$.* pp *are implicit input to all algorithms.*
2. $\mathtt{KGen}(\mathsf{pp}) \to (\mathsf{pk}, \mathsf{sk})$*: On input public parameters* pp*, output a key pair* $(\mathsf{pk}, \mathsf{sk})$ *.*
3. $\mathtt{Reg}(\mathsf{pk}) \to (\mathsf{ed}, \mathsf{dd})$*: On input an honestly generated public key* pk*, deterministically output public encryption and decryption digests* $(\mathsf{ed}, \mathsf{dd})$ *using the public parameters.*
4. $\mathtt{Enc}(\mathsf{ed}, L, \mathsf{msg}) \to \mathsf{ct}$*: On input an encryption digest* ed*, revoked set L with $|L| \leq n$, and message $\mathsf{msg} \in \mathcal{M}$, output a ciphertext* ct *for the registered key and L.*
5. $\mathtt{Dec}(\mathsf{sk}, \mathsf{dd}, L, \mathsf{ct}) \to \mathsf{msg}$ *or* $\perp$*: On input a secret key* sk*, its public decryption digest* dd*, revoked set L, and ciphertext* ct*, output the decrypted message* msg *or* $\perp$ *on failure.*
6. $\mathtt{Revoke}(y) \to y$*: On input a user's revocation pseudonym y, output and publish y, and update $L \leftarrow L \cup \{y\}$.*

The ciphertext fixes its recipient and revoked set L as public context. The scheme satisfies the following properties:

1. ***Correctness.*** *For any $n = \mathsf{poly}(\lambda)$, honestly generated public parameters and keys $(\mathsf{pk}_i, \mathsf{sk}_i)_{i \in [\![n]\!]}$, and registration outputs $(\mathsf{ed}_i, \mathsf{dd}_i) \leftarrow \mathtt{Reg}(\mathsf{pk}_i)$, write y_i for user i's pseudonym. For every message $\mathsf{msg} \in \mathcal{M}$, revoked set L with $|L| \leq n$, and user i with $y_i \notin L$,*

$$\Pr\big[\mathtt{Dec}(\mathsf{sk}_i, \mathsf{dd}_i, L, \mathtt{Enc}(\mathsf{ed}_i, L, \mathsf{msg})) = \mathsf{msg}\big] \geq 1 - \mathsf{negl}(\lambda).$$

   The probability is over the honest algorithms' randomness. Correctness is perfect *when recovery succeeds with probability one.*

2. ***Perfect hiding.*** *For any two fixed vectors of user pseudonyms, honest key generation and registration with those pseudonyms give identically distributed joint public views. This view consists of* pp *and all triples $(\mathsf{pk}_i, \mathsf{ed}_i, \mathsf{dd}_i)$, including any CRS and public auxiliary data. All remaining key-generation randomness is sampled honestly and independently across users.*

Definition 17 (Adaptive IND-CPA security for revocable PKE)**.** *Consider the following game between PPT adversary $\mathcal{A}$ and challenger $\mathcal{C}$:*

1. *$\mathcal{A}$ chooses a polynomially bounded number of users n and sends it to $\mathcal{C}$.*
2. *$\mathcal{C}$ samples* $\mathsf{pp} \leftarrow \mathtt{Setup}(1^\lambda, n)$*, generates* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathtt{KGen}(\mathsf{pp})$ *and* $(\mathsf{ed}_i, \mathsf{dd}_i) \leftarrow \mathtt{Reg}(\mathsf{pk}_i)$ *for each $i \in [\![n]\!]$, and sends* pp *and all triples* $(\mathsf{pk}_i, \mathsf{ed}_i, \mathsf{dd}_i)$ *to $\mathcal{A}$. Write y_i for user i's associated pseudonym. $\mathcal{C}$ also initializes $L = \mathsf{Cor} = \emptyset$, where* Cor *records corrupted users.*
3. *$\mathcal{A}$ may adaptively make polynomially many queries:*
   - ***Revoke.*** *On input $i \in [\![n]\!]$, $\mathcal{C}$ runs $\mathtt{Revoke}(y_i)$ and returns its output, updating the public revoked set L.*
   - ***Corrupt.*** *On input $i \in [\![n]\!]$, $\mathcal{C}$ returns the full secret key* sk_i *and updates* $\mathsf{Cor} \leftarrow \mathsf{Cor} \cup \{i\}$.
4. *$\mathcal{A}$ submits a challenge recipient $i^* \in [\![n]\!]$ and equal-length messages* $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathcal{M}$*. $\mathcal{C}$ sets $L^* \leftarrow L$, samples $b \leftarrow_\$ \{0, 1\}$, and returns* $\mathsf{ct}^* \leftarrow \mathtt{Enc}(\mathsf{ed}_{i^*}, L^*, \mathsf{msg}_b)$ *to $\mathcal{A}$.*
5. *$\mathcal{A}$ continues making Revoke and Corrupt queries, then outputs a bit b'. The challenge recipient, registration outputs, and revoked set L^* remain fixed, even after later revocations.*

$\mathcal{A}$ is admissible *if, in every execution, $i^* \in \mathsf{Cor} \implies y_{i^*} \in L^*$. Define*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\Sigma,\mathcal{A}}(\lambda) := \left|\Pr[b' = b] - \frac{1}{2}\right|.$$

The scheme satisfies adaptive IND-CPA security if this advantage is negligible in λ for every admissible PPT $\mathcal{A}$. The probability is over the randomness of $\mathcal{A}$ and $\mathcal{C}$.

5.2 Construction and Analysis

Construction 4 (Non-membership encryption)**.** *Compile all three checks of Construction 2 using Theorem 1 and the identity reduction on* NM*. The resulting* $\mathsf{WE}_{\mathsf{NM}}$ *has message space $\mathbb{G}_T$ and random oracle $H_{\mathsf{NM}} : \mathbb{G}_T \to \mathbb{G}_T$, with a domain separate from the PKE oracles. All algorithms take* pp *implicitly.*

- $\mathsf{WE}_{\mathsf{NM}}.\mathtt{Gen}$*: Use* `Gen` *from Construction 2.*
- $\mathsf{WE}_{\mathsf{NM}}.\mathtt{AuxGen}$*: Use* `AuxGen` *from Construction 2.*
- $\mathsf{WE}_{\mathsf{NM}}.\mathtt{Digest}(\mathsf{crs}, \mathsf{idx}, \bot)$*: Run the gadget's* `Digest` *and output* $\mathsf{ed} = \mathsf{Vk}$ *and* $\mathsf{dd} = \mathsf{Pk}$*. Both digests are independent of L.*
- $\mathsf{WE}_{\mathsf{NM}}.\mathtt{Enc}(\mathsf{ed}, L, \mathsf{msg})$*: Compute $P = [P_L(\theta)]_1$ from L and the powers in* pp*. Sample independent $s_0, s_1, s_2 \leftarrow_{\$} \mathbb{F}_p$ and set*

$$h_c = [s_0]_2, \qquad h_u = s_1\mathsf{idx}, \quad h_v = s_1 P,$$
$$h_w = -s_1[\alpha]_1, \quad h_y = [s_2]_1, \quad h_r = -s_2[\alpha]_1.$$

  Compute

$$K = s_0\big(\mathsf{idx} \circ [1]_2\big) + [s_1]_T + s_2\big(([\theta]_1 - \mathsf{idx}) \circ [1]_2\big)$$

  and return $(h_c, h_u, h_v, h_w, h_y, h_r, d)$, where $d = \mathsf{msg} + H_{\mathsf{NM}}(K)$.
- $\mathsf{WE}_{\mathsf{NM}}.\mathtt{Prove}(\mathsf{dd}, L, (y, r))$*: Compute $\overline{c} = \mathsf{Com}(y; r)$ and run the gadget's* $\mathtt{Prove}(\mathsf{Pk}, (\overline{c}, L), (y, r))$ *to obtain $\pi = (\overline{u}, \overline{v}, \overline{w}, \overline{y}, \overline{r})$. Output $\Pi_{\mathsf{NM}} = (\overline{c}, \pi)$.*
- $\mathsf{WE}_{\mathsf{NM}}.\mathtt{Dec}(\mathsf{ct}, \Pi_{\mathsf{NM}})$*: Parse the ciphertext and proof as above, and return*

$$d - H_{\mathsf{NM}}\big(\overline{c} \circ h_c + h_u \circ \overline{u} + h_v \circ \overline{v} + h_w \circ \overline{w} + h_y \circ \overline{y} + h_r \circ \overline{r}\big).$$

For a valid witness, checks 1, 2, and 3, multiplied by s_0, s_1, s_2 respectively, show that the decryption hash input equals K. This proves perfect correctness. When $L = \emptyset$, the non-membership proof uses $u = w = 0$ and $v = 1$, but the opening check is still included. The gadget's knowledge soundness and Theorem 1 give extractable security for NM in the generic group and random oracle models.

Construction 5 (Revocable public-key encryption by secret sharing)**.** *Use message space $\mathbb{G}_T$ and pseudonym space $\mathcal{Y} = \mathbb{F}_p$. Let* $\mathsf{PKE} = (\mathsf{PKE}.\mathtt{KGen}, \mathsf{PKE}.\mathtt{Enc}, \mathsf{PKE}.\mathtt{Dec})$ *be IND-CPA secure in the stated oracle environment, with parameters and keys generated independently of the gadget setup. Its plaintext space supports a fixed-length injective encoding* encode *of $\mathbb{G}_T$, with inverse* decode *rejecting malformed encodings. Polynomial-length bit strings suffice.*

- $\mathtt{Setup}(1^\lambda, n)$*: Sample bilinear groups $\mathcal{BG}$ and independent trapdoors* $\mathsf{td} = (\theta, \alpha) \leftarrow_{\$} (\mathbb{F}_p^\times)^2$*. Compute*

$$(\mathsf{pp}_{\mathsf{NM}}, \mathsf{crs}_{\mathsf{NM}}) \leftarrow \mathsf{WE}_{\mathsf{NM}}.\mathtt{Gen}(\lambda, n, (\theta, \alpha)),$$

  erase the trapdoors, and independently generate any public parameters required by PKE*. Output* $\mathsf{pp}_{\mathsf{NM}}$*,* $\mathsf{crs}_{\mathsf{NM}}$*, and the PKE public parameters together as* pp *, and initialize $L = \emptyset$.*

- $\mathtt{KGen}(\mathsf{pp})$*: Independently sample* $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{PKE}.\mathtt{KGen}$ *and* $y \leftarrow_\$ \mathcal{Y}$*,* $r \leftarrow_\$ \mathbb{F}_p$*, and compute* $c \leftarrow [\theta - y + \alpha r]_1$*. Output* $\mathsf{pk} \leftarrow (\mathsf{pk}_0, c)$ *and* $\mathsf{sk} \leftarrow (\mathsf{sk}_0, y, r)$*.*
- $\mathtt{Reg}(\mathsf{pk})$*: Parse* $\mathsf{pk} = (\mathsf{pk}_0, c)$ *and compute* $(\mathsf{ed}_{\mathsf{NM}}, \mathsf{dd}_{\mathsf{NM}}) \leftarrow \mathsf{WE}_{\mathsf{NM}}.\mathtt{Digest}(\mathsf{crs}_{\mathsf{NM}}, c, \bot)$*. Output the public digests* $\mathsf{ed} \leftarrow (\mathsf{pk}_0, \mathsf{ed}_{\mathsf{NM}})$ *and* $\mathsf{dd} \leftarrow \mathsf{dd}_{\mathsf{NM}}$*, independent of* L*.*
- $\mathtt{Enc}(\mathsf{ed}, L, \mathsf{msg})$*: Parse* $\mathsf{ed} = (\mathsf{pk}_0, \mathsf{ed}_{\mathsf{NM}})$*, sample a uniform share* $R \leftarrow_\$ \mathbb{G}_T$*, and output* $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_{\mathsf{NM}})$*, where*

$$\begin{aligned} \mathsf{ct}_0 &\leftarrow \mathsf{PKE}.\mathtt{Enc}(\mathsf{pk}_0, \mathsf{encode}(R)), \\ \mathsf{ct}_{\mathsf{NM}} &\leftarrow \mathsf{WE}_{\mathsf{NM}}.\mathtt{Enc}(\mathsf{ed}_{\mathsf{NM}}, L, \mathsf{msg} - R). \end{aligned}$$

  The encryptions use independent randomness. The ciphertext fixes L *as public context, either explicitly or by reference.*
- $\mathtt{Dec}(\mathsf{sk}, \mathsf{dd}, L, \mathsf{ct})$*: Parse* $\mathsf{sk} = (\mathsf{sk}_0, y, r)$ *and* $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_{\mathsf{NM}})$*, and return* $\bot$ *if* $y \in L$*. Otherwise compute*

$$\begin{aligned} R &\leftarrow \mathsf{decode}(\mathsf{PKE}.\mathtt{Dec}(\mathsf{sk}_0, \mathsf{ct}_0)), \\ \Pi_{\mathsf{NM}} &\leftarrow \mathsf{WE}_{\mathsf{NM}}.\mathtt{Prove}(\mathsf{dd}, L, (y, r)), \\ S &\leftarrow \mathsf{WE}_{\mathsf{NM}}.\mathtt{Dec}(\mathsf{ct}_{\mathsf{NM}}, \Pi_{\mathsf{NM}}). \end{aligned}$$

  Return $\bot$ *if a computation fails. Otherwise output* $R + S$*.*
- $\mathtt{Revoke}(y) \to y$*: Publish* y *and update* $L \leftarrow L \cup \{y\}$*.*

Lemma 7 (Correctness and pseudonym hiding)**.** *Construction 5 satisfies Definition 16. Its correctness error is at most that of* PKE*, so perfect PKE correctness gives perfect correctness.*

Proof. For $y \notin L$, the gadget recovers $\mathsf{msg} - R$ perfectly. Thus decryption recovers msg whenever PKE recovers R.

Definition 11 gives hiding because revocation inputs are independent across users and of other key material, and registration uses public data. Adaptively disclosing some users' pseudonyms or full keys therefore leaves the remaining pseudonyms hidden. Swapping two undisclosed pseudonyms and adjusting their masks preserves the public view, including later disclosures about other users. □

Theorem 3 (Adaptive IND-CPA security)**.** *Construction 5 satisfies Definition 17 in the generic group and random oracle models under the following condition. If* PKE *uses the gadget's groups, its public inputs must be independent of the fresh witness-encryption randomness. Their formal exponents have polynomially bounded degree, with source-*$\mathbb{G}_2$ *exponents in* $\mathbb{F}_p[Y, \boldsymbol{\zeta}]$ *and additional secret symbols* $\boldsymbol{\zeta}$ *independent of the setup trapdoors. This must remain true after permitted scalar disclosures, excluding* Z*-dependence and inverses of nonconstant polynomials in* Y*. Independently instantiated bit-string PKE adds no such source inputs.*

Proof. Let i^* and L^* be the challenge recipient and revoked list, chosen before the challenge randomness. We consider two cases.

If $y_{i^*} \in L^*$, give $\mathcal{A}$ all secret keys, registration openings, and the PKE-encrypted share R. Write $y = y_{i^*}$, $r = r_{i^*}$, and $p(Y, Z) = Y - y + rZ$. Let T_0, T_1, T_2, Y, Z represent $s_0, s_1, s_2, \theta, \alpha$. The additional PKE secrets are represented by $\boldsymbol{\zeta}$. The hash input K has formal exponent

$$T_0 p(Y, Z) + T_1 + T_2\big(Y - p(Y, Z)\big).$$

Set $T_0 = T_2 = 0$ in any candidate oracle query and compare coefficients of T_1. The original source-$\mathbb{G}_2$ exponents lie in $\mathbb{F}_p[Y, \boldsymbol{\zeta}]$ and the only fresh $\mathbb{G}_2$ ciphertext element is $[T_0]_2$, so this coefficient has the form

$$p(Y, Z) q_1(Y, \boldsymbol{\zeta}) + P_{L^*}(Y) q_2(Y, \boldsymbol{\zeta}) - Z q_3(Y, \boldsymbol{\zeta}),$$

where $q_j \in \mathbb{F}_p[Y, \boldsymbol{\zeta}]$. This vanishes at $Y = y, Z = 0$, whereas K has coefficient 1. No formal query therefore equals K. Independent oracle outputs add fresh target-group symbols and preserve this obstruction. Outside negligible generic-collision and oracle-guessing events, the unqueried uniform mask $H_{\mathsf{NM}}(K)$ hides $\mathsf{msg}_b - R$ even given all keys and R.

If $y_{i^*} \notin L^*$, admissibility forbids corrupting i^* even after the challenge. Guess i^*, embed the PKE challenge key, and generate all other keys and revocation openings independently. Replace the encryption of $\mathsf{encode}(R)$ by one of $\mathsf{encode}(0_{\mathbb{G}_T})$, keeping the non-membership ciphertext. IND-CPA security justifies this with a polynomial guessing loss. Return an independent random bit on a wrong guess, the other branch, or target corruption. On the correct guess in this branch, target corruption is impossible in the real experiment and negligible after replacement, since it is detectable. Now R appears only in the uniform share $\mathsf{msg}_b - R$, making the view independent of b. □

Efficiency. The construction adds $O(n)$ public group elements to PKE. Apart from the public context L, a ciphertext contains one PKE encryption of an encoded group element, five elements of $\mathbb{G}_1$, one of $\mathbb{G}_2$, and one of $\mathbb{G}_T$. Revocation affects future ciphertexts since existing ciphertexts retain their original list L.

6 Revocation for Advanced Encryption Schemes

To add revocation, route the same selector and its commitment to the authorization relation $\mathcal{R}$ and indexed non-membership, with the same registered positions and commitment key:

$$\mathcal{R}_{\mathsf{rev}} = (\mathcal{R} \times \mathsf{INM}^{\mathsf{wt}})_{\mathcal{C}}.$$

The authorization relation enforces its access rule, including nontriviality when required (i.e. the decryptors are not the zero set), while $\mathsf{INM}^{\mathsf{wt}}$ requires every selected user to be unrevoked. Theorem 1 compiles these checks into witness encryption when the gadgets are linearly verifiable and $\mathcal{C}$-intercompatible, with extraction under the full public setup and auxiliary view.

6.1 Revocable Distributed Broadcast Encryption

6.1.1 Definition

We use the threshold-one registered syntax of [GHK+25, Definition 19 and Section 7.3], adding pseudonym-hiding revocation for honestly generated keys.

Identities are public labels, distinct from secret keys and pseudonyms. Registration outputs a master public key mpk and public decryption digest dd, initially $\perp$.

Definition 18 (Revocable Distributed Broadcast Encryption)**.** *A revocable DBE scheme Σ with message space $\mathcal{M}$, identity space $\mathcal{ID}$, and pseudonym space $\mathcal{Y}$ has seven PPT algorithms. Each user samples $y_i \in \mathcal{Y}$. The revocation authority retains the user-to-pseudonym map and publishes $L \subseteq \mathcal{Y}$, initially empty. Public parameters* pp *are implicit inputs.*

1. $\mathtt{Setup}(1^\lambda, M) \to (\mathsf{pp}, \mathsf{crs})$*: Generate public parameters and a CRS for security parameter λ and capacity $M > 0$.*
2. $\mathtt{KGen}(\mathsf{crs}) \to (\mathsf{aux}, \mathsf{pk}, \mathsf{sk})$*: Generate a public key, secret key, and public auxiliary data.*
3. $\mathtt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux}) \to \{0, 1\}$*: Output 1 if the key and auxiliary data are well formed, and 0 otherwise.*
4. $\mathtt{Reg}(\mathsf{crs}, \mathsf{dd}, \mathsf{id}, \mathsf{pk}, \mathsf{aux}) \to (\mathsf{mpk}', \mathsf{dd}')$ *or $\perp$: Validate the key and auxiliary data and register them under $\mathsf{id} \in \mathcal{ID}$. Deterministically output the updated master key and public digest, or $\perp$ if validation fails, the identity is already registered, or registration would exceed M users.*

5. $\texttt{Enc}(\mathsf{mpk}, S, L, \mathsf{msg}) \to \mathsf{ct}$*: Encrypt* $\mathsf{msg} \in \mathcal{M}$ *to registered recipients* $S \subseteq \mathcal{ID}$ *whose pseudonyms lie outside* $L \subseteq \mathcal{Y}$*, where* $|L| \leq n$ *and* n *is this key's registry size. Include* S, L *as public context.*

6. $\texttt{Dec}(\mathsf{crs}, \mathsf{dd}, (\mathsf{id}_i, \mathsf{sk}_i)_{i \in I}, S, \mathsf{ct}) \to \mathsf{msg}$ *or* $\perp$*: Decrypt locally using the labelled keys and digest for the encryption key. Output the message or* $\perp$*.*

7. $\texttt{Revoke}(y) \to y$*: Publish and output* y*, and update* $L \leftarrow L \cup \{y\}$*.*

Existing ciphertexts use their original registry's outputs after later registrations. The scheme satisfies the following properties.

1. ***Key validation.*** *For honest* $(\mathsf{pp}, \mathsf{crs})$*,* $\texttt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux}) = 1$ *exactly when some* $(\mathsf{aux}, \mathsf{pk}, \mathsf{sk})$ *lies in the support of* $\texttt{KGen}(\mathsf{crs})$*. Such pairs can be registered under unused identities when capacity remains.*

2. ***Perfect correctness.*** *For every honest setup and registry, every set* S *of registered recipient identities, revoked set* L*, message* msg*, and collection of honestly generated keys indexed by* I*, if some* $i \in I$ *satisfies* $\mathsf{id}_i \in S$ *and* $y_i \notin L$*, then*

$$\texttt{Dec}\big(\mathsf{crs}, \mathsf{dd}, (\mathsf{id}_i, \mathsf{sk}_i)_{i \in I}, S, \texttt{Enc}(\mathsf{mpk}, S, L, \mathsf{msg})\big) = \mathsf{msg}$$

   with probability one over the honest algorithms' randomness. The master key and digest are the outputs for the ciphertext's registry.

3. ***Perfect pseudonym hiding.*** *The scheme satisfies Definition 20 below.*

4. ***Compactness.*** *The ciphertext has size at most* $|\mathsf{msg}| + \mathrm{poly}(\lambda)$*, excluding the explicit public recipient set and revoked list. The master public key has size* $\mathrm{poly}(\lambda, \log M)$ *beyond the shared public parameters. The registry, auxiliary data, and reusable public decryption digest are counted separately.*

Definition 19 (Adaptive IND-CPA Security for Revocable DBE)**.** *Consider the following experiment between PPT* $\mathcal{A}$ *and challenger* $\mathcal{C}$*, using the algorithms of Definition 18.*

1. $\mathcal{A}$ *chooses a polynomially bounded capacity* M *and* $1 \leq n \leq M$*.* $\mathcal{C}$ *samples* $(\mathsf{pp}, \mathsf{crs}) \leftarrow \texttt{Setup}(1^\lambda, M)$ *and sends* $(\mathsf{pp}, \mathsf{crs})$ *to* $\mathcal{A}$*.*

2. $\mathcal{A}$ *chooses distinct identities* $\mathsf{id}_1, \ldots, \mathsf{id}_n \in \mathcal{ID}$*. Starting with* $\mathsf{dd}_0 = \perp$*, for* $i = 1, \ldots, n$*,* $\mathcal{C}$ *independently samples* $(\mathsf{aux}_i, \mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \texttt{KGen}(\mathsf{crs})$ *and computes*

$$(\mathsf{mpk}_i, \mathsf{dd}_i) \leftarrow \texttt{Reg}(\mathsf{crs}, \mathsf{dd}_{i-1}, \mathsf{id}_i, \mathsf{pk}_i, \mathsf{aux}_i).$$

   It sends every $(\mathsf{pk}_i, \mathsf{aux}_i, \mathsf{mpk}_i, \mathsf{dd}_i)$ *to* $\mathcal{A}$ *and sets* $(\mathsf{mpk}, \mathsf{dd}) = (\mathsf{mpk}_n, \mathsf{dd}_n)$*. The registry and these outputs remain fixed. Let* y_i *denote user* i*'s pseudonym, and initialize* $L = R = \mathsf{Cor} = \emptyset$*.*

3. $\mathcal{A}$ *makes polynomially many adaptive* $\mathsf{Revoke}(i)$ *and* $\mathsf{Corrupt}(i)$ *queries for* $i \in [\![n]\!]$*. On* $\mathsf{Revoke}(i)$*,* $\mathcal{C}$ *returns* $\texttt{Revoke}(y_i)$*, updates* $L \leftarrow L \cup \{y_i\}$*, and sets* $R \leftarrow R \cup \{i\}$*. On* $\mathsf{Corrupt}(i)$*,* $\mathcal{C}$ *returns* sk_i *and sets* $\mathsf{Cor} \leftarrow \mathsf{Cor} \cup \{i\}$*.*

4. $\mathcal{A}$ *submits a recipient set* $S^* \subseteq \{\mathsf{id}_1, \ldots, \mathsf{id}_n\}$ *and equal-length messages* $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathcal{M}$*.* $\mathcal{C}$ *sets* $L^* \leftarrow L$*, samples* $b \leftarrow_\$ \{0, 1\}$*, and sends* $\mathsf{ct}^* \leftarrow \texttt{Enc}(\mathsf{mpk}, S^*, L^*, \mathsf{msg}_b)$ *to* $\mathcal{A}$*.*

5. *Queries may continue before* $\mathcal{A}$ *outputs* b'*. The registry,* S^**, and* L^* *remain fixed for the challenge, including after subsequent revocations.*

An adversary is admissible if, in every execution, every i *in the final* Cor *with* $\mathsf{id}_i \in S^*$ *satisfies* $y_i \in L^*$*. Adaptive IND-CPA security requires* $|\Pr[b' = b] - 1/2|$ *to be negligible in* λ *for every admissible PPT* $\mathcal{A}$*, over the randomness of* $\mathcal{A}$ *and* $\mathcal{C}$*.*

Definition 20 (Perfect Pseudonym Hiding for Revocable DBE)**.** *Fix honest public parameters and distinct public labels* $\mathsf{id}_1, \ldots, \mathsf{id}_n \in \mathcal{ID}$*. Honest key generation conditioned on any two possible pseudonym vectors in* $\mathcal{Y}^n$ *gives identically distributed public views, comprising the CRS, all public keys and auxiliary data, and every registration output.*

Consider the following experiment between PPT adversary $\mathcal{A}$ *and challenger* $\mathcal{C}$*.*

1. *Perform Steps 1 and 2 of Definition 19, with* $2 \le n \le M$*.*
2. $\mathcal{A}$ *makes polynomially many adaptive queries to the* Revoke *and* Corrupt *oracles specified in Step 3 of Definition 19.*
3. $\mathcal{A}$ *selects distinct* $i_0, i_1 \in [\![n]\!] \setminus (R \cup \mathsf{Cor})$*.* $\mathcal{C}$ *samples* $b \leftarrow_{\$} \{0,1\}$ *and returns* y_{i_b} *anonymously, without adding either challenge user to* R *or changing* L*.*
4. *Revoke and Corrupt queries may continue, and* $\mathcal{A}$ *outputs* b'*.*

An adversary $\mathcal{A}$ *is admissible if, in every execution, the final sets* R *and* Cor *satisfy* $\{i_0, i_1\} \cap (R \cup \mathsf{Cor}) = \emptyset$*. The scheme satisfies perfect pseudonym hiding if the registration-view distributions specified above are identical and, for every admissible PPT* $\mathcal{A}$ *and every security parameter* λ*,*

$$\Pr[b' = b] = \frac{1}{2},$$

where the probability is over all randomness in the experiment, including that of $\mathcal{A}$ *and* $\mathcal{C}$*.*

6.1.2 Construction and Analysis

Fix $D = \mathcal{ID} = (d_1, \ldots, d_M)$, with $\mathsf{id}_i = d_i$ and $n \le M$ registered users. Write $\mathbf{pk} = (\mathsf{pk}_i^{\mathsf{DBE}})_i$, where $\mathsf{pk}_i^{\mathsf{DBE}} = [x_i]_1$, and $\mathbf{pk}^{\mathsf{INM}} = (\mathsf{pk}_i^{\mathsf{INM}})_i$, where $\mathsf{pk}_i^{\mathsf{INM}} = [p_i(\theta, \alpha)]_1$. Both indices include identity labels and dummy entries at unregistered slots. Let $H : \{0,1\}^* \to \mathbb{G}_2$ be a random oracle independent of the compiler's masking oracle.

The relation DBE of [Pol26, GHK+25] has parameters $[1]_1$, index $\mathbf{pk}$, instance (tag, S), and witness $(\sigma, b) \in \mathbb{G}_2 \times (\mathbb{F}_p^M \setminus \{0\})$. Here $\mathsf{tag} \in \{0,1\}^*$ and S contains registered identities in D. It requires $b_i = 0$ whenever $\mathsf{id}_i \notin S$, and

$$[1]_1 \circ \sigma = \left(\sum_{i \in [\![M]\!]} b_i \, \mathsf{pk}_i^{\mathsf{DBE}} \right) \circ H(\mathsf{tag}).$$

Define rDBE by adding INM with parameters $\mathsf{pp}_{\mathsf{INM}}$, index $\mathbf{pk}^{\mathsf{INM}}$, instance L, and witness (b, y, r), together with $\sum_i b_i = 1$. The normalized NonZero gadget of Lemma 17 enforces this additional constraint. A registered-slot check below forces inactive coordinates to zero.

Construction 6 (Revocable Distributed Broadcast Encryption)**.** *Use the extended CRS of Appendix A, common setup and auxiliary generators, and normalized NonZero gadget. Instantiate* $\mathsf{INM}^{\mathsf{wt}}$ *by Construction 3 on* D*, sharing its SCS selector commitment with DBE. Use shifts* X^{N-M+1} *for all six encodings,* Y^{N-M+1} *for* U, W*, and* Y^N *for* B, V*.*

Let $E \subseteq D$ *be the identities under which users have registered. For* $A \in \{S, E\}$*, put* $Z_A(X) = \prod_{d \in A}(X - d)$*. For the shared selector* $B(X) = \sum_i b_i L_i(X)$*, supply* $[Q_A(\tau)]_2$*, where* $Q_A(X) = Z_A(X)B(X)/Z_D(X)$*, and check*

1. $[Z_S(\tau)]_1 \circ [B(\tau)]_2 = [Z_D(\tau)]_1 \circ [Q_S(\tau)]_2$*.*
2. $[Z_E(\tau)]_1 \circ [B(\tau)]_2 = [Z_D(\tau)]_1 \circ [Q_E(\tau)]_2$*.*

These Zero Checks [GHK+25, Section 5.3, Construction 5] use fixed verifier data $[Z_D(\tau)]_1$ *and the shared selector degree checks of Lemma 15. Compile the components into* WE *by Theorem 1.*

- $\texttt{Setup}(1^\lambda, M)$: *Set* $N = M$. *Sample bilinear groups* $\mathcal{BG}$ *and independent* $\theta, \alpha \leftarrow_\$ \mathbb{F}_p^\times$. *Initialize* $L = \emptyset$ *and use pseudonym space* $\mathcal{Y} = \mathbb{F}_p$. *Fix a public identity space* $\mathcal{ID}$ *of* M *distinct field elements and a public point* $x^* \notin \mathcal{ID}$. *Independently sample* $\tau \leftarrow_\$ \mathbb{F}_p \setminus \mathcal{ID}$, *set* $\mathsf{td} = (\tau, \theta, \alpha)$, *and run the common generator over* $\mathcal{BG}$:

$$(\mathsf{pp}_0, \mathsf{crs}) \leftarrow \texttt{Gen}(\lambda, N, \mathsf{td}).$$

Output $(\mathsf{pp}, \mathsf{crs})$, *where* $\mathsf{pp} = (\mathsf{pp}_0, M, \mathcal{ID}, x^*)$, *and erase the setup trapdoors.*

- $\texttt{KGen}(\mathsf{crs})$: *Independently sample* $x \leftarrow_\$ \mathbb{F}_p^\times$ *and* $y \leftarrow_\$ \mathcal{Y}$, $r \leftarrow_\$ \mathbb{F}_p$. *Set*

$$\begin{aligned} \mathsf{pk}^{\mathsf{DBE}} &= [x]_1, & \mathsf{aux}^{\mathsf{DBE}} &= ([x\tau^e]_1)_{1 \le e \le N}, \\ \mathsf{pk}^{\mathsf{INM}} &= q = [\theta - y + \alpha r]_1, & \mathsf{aux}^{\mathsf{INM}} &= ([p(\theta, \alpha)\tau^a \theta^c]_1)_{0 \le a,c < M}, \end{aligned}$$

where $p(Y, Z) = Y - y + rZ$. *Output* $(\mathsf{aux}, \mathsf{pk}, \mathsf{sk})$, *where*

$$\mathsf{aux} = (\mathsf{aux}^{\mathsf{DBE}}, \mathsf{aux}^{\mathsf{INM}}), \qquad \mathsf{pk} = (\mathsf{pk}^{\mathsf{DBE}}, \mathsf{pk}^{\mathsf{INM}}), \qquad \mathsf{sk} = (x, y, r).$$

- $\texttt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux})$: *Reject if* $\mathsf{pk}^{\mathsf{DBE}} = [0]_1$. *For every DBE entry* A_e *and non-membership entry* $R_{a,c}$, *check*

$$\begin{aligned} A_e \circ [1]_2 &= \mathsf{pk}^{\mathsf{DBE}} \circ [\tau^e]_2 \qquad (1 \le e \le N), \\ R_{a,c} \circ [1]_2 &= \mathsf{pk}^{\mathsf{INM}} \circ [\tau^a \theta^c]_2 \quad (0 \le a, c < M). \end{aligned}$$

Output 1 *if all checks pass and* 0 *otherwise.*

- $\texttt{Reg}(\mathsf{crs}, \mathsf{dd}, \mathsf{id}, \mathsf{pk}, \mathsf{aux})$: *Read records and digests from* dd. *For* $\mathsf{dd} = \perp$, *initialize all* M *slots with* $\mathsf{pk}_i^{\mathsf{DBE}} = [0]_1$, *zero DBE powers,* $p_i(Y, Z) = Y$, *and* $\mathsf{pk}_i^{\mathsf{INM}} = [\theta]_1$ *as in Remark 1, without validating these dummy entries. Reject unless* $\mathsf{id} \in \mathcal{ID}$ *is unused, fewer than* M *users are registered, and* $\texttt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux}) = 1$. *Add* $(\mathsf{id}, \mathsf{pk}, \mathsf{aux}^{\mathsf{DBE}})$ *to* records. *Update the* INM *digest, all* M *helper rows, and* C_0 *by that remark, then discard* $\mathsf{aux}^{\mathsf{INM}}$.

  Compute the IIP digest of Lemma 16 from retained powers on D, *padding inactive keys and powers by zero and using the prescribed shifts. Generate the normalization digest from* 1 *and* $[Z_E(\tau)]_1$ *from the updated registered set* E. *Assemble*

$$(\mathsf{ed}_{\mathsf{WE}}, \mathsf{dd}_{\mathsf{WE}}) \leftarrow \mathsf{AssembleDigest}\big(\mathsf{dd}_{\mathsf{DBE}}, \mathsf{dd}_{\mathsf{INM}}, \mathsf{dd}_{\mathrm{norm}}, [Z_E(\tau)]_1\big).$$

  where AssembleDigest *combines component prover and verifier data, including registered-slot Check 2 and fixed helper keys from the CRS. Output*

$$\mathsf{mpk}' = (n, \mathsf{ed}_{\mathsf{WE}}), \qquad \mathsf{dd}' = (\mathsf{records}, \mathsf{dd}_{\mathsf{WE}}).$$

- $\texttt{Enc}(\mathsf{mpk}, S, L, \mathsf{msg})$: *Write* $\mathsf{mpk} = (n, \mathsf{ed}_{\mathsf{WE}})$. *Compute* $[Z_S(\tau)]_1$ *from* S *and the CRS for Check 1. Sample* $\mathsf{tag} \leftarrow_\$ \{0,1\}^\lambda$ *and compute*

$$\mathsf{ct}_{\mathsf{WE}} \leftarrow \mathsf{WE}.\texttt{Enc}(\mathsf{ed}_{\mathsf{WE}}, (\mathsf{tag}, S, L), \mathsf{msg}).$$

  Output $\mathsf{ct} = (S, L, \mathsf{tag}, \mathsf{ct}_{\mathsf{WE}})$.

- $\texttt{Dec}(\mathsf{crs}, \mathsf{dd}, (\mathsf{id}_i, \mathsf{sk}_i)_{i \in I}, S, \mathsf{ct})$: *Read the registry and* $\mathsf{dd}_{\mathsf{WE}}$ *from* dd, *and* (S, L, tag) *from the ciphertext. Let* T *index held keys whose labels satisfy* $\mathsf{id}_i \in S$ *and* $P_L(y_i) \neq 0$. *If* $T = \emptyset$, *output* $\perp$. *Otherwise choose* $b \in \mathbb{F}_p^M$ *supported on* T*'s fixed slots with* $\sum_i b_i = 1$, *and set*

$$\sigma = \sum_{i \in T} b_i x_i H(\mathsf{tag}).$$

Set $t_i = b_i y_i$, $s_i = b_i r_i$ on selected positions and zero elsewhere, and compute

$$\pi \leftarrow \mathsf{WE.Prove}\big(\mathsf{dd}_{\mathsf{WE}}, (\mathsf{tag}, S, L), (\sigma, b, t, s)\big).$$

Output $\mathsf{WE.Dec}(\mathsf{ct}_{\mathsf{WE}}, \pi)$.

- $\mathtt{Revoke}(y)$*: Publish and output y, and update $L \leftarrow L \cup \{y\}$.*

Lemma 8 (Correctness and hiding)**.** *Construction 6 satisfies Definition 18, including perfect correctness and perfect pseudonym hiding under Definition 20.*

Proof. By nondegeneracy, the validation equations uniquely fix each auxiliary entry as the prescribed multiple of its index [GHK+25, Construction 8, footnote 13]. The helper degrees and reusable-CRS bounds of Construction 3 supply all quotient and shifted encodings, including when $M = 1$.

The selector chosen by `Dec` satisfies the signature equation and normalization. Its selected openings give a valid $\mathsf{INM}^{\mathsf{wt}}$ witness, so Lemma 4 gives the non-membership proof. For $A \in \{S, E\}$, the selector vanishes on $D \setminus A$, making $Q_A = Z_A B / Z_D$ a polynomial of degree at most $|A| - 1$, or zero if $A = \emptyset$. Its encoding satisfies the support check. Component and compiler completeness give decryption. One eligible key suffices by taking $b = e_i$.

Lemma 5, independent DBE keys, and public registration give identical public views for every pseudonym vector. Undisclosed pseudonyms remain independent and uniform after adaptive pseudonym or full-key disclosures. Swapping the challenge pseudonyms and adjusting their masks preserves the view and all permitted later responses. Since decryption publishes no witness, proof, or outcome, Definition 20 follows.

A constant number of succinct gadgets gives the ciphertext bound and a master key containing constantly many group elements and n, with shared parameters and public decryption data counted separately. □

Lemma 9 (Security)**.** *Construction 6 satisfies Definition 19 in the generic bilinear-group and random-oracle models.*

Proof. The gadgets share setup, auxiliary generators, and commitment algorithms. Completeness and knowledge soundness hold under the combined public view by Appendix A and Theorem 2, so Theorem 1 applies. For either support check, extraction gives the identity $Z_A(X)B(X) = Z_D(X)Q_A(X)$. At each $d \in D \setminus A$, $Z_A(d) \neq 0$, forcing $B(d) = 0$. This is the Zero Check argument of Lemma 15, with the public factor Z_A canceled.

Give the adversary all revocation secrets. These data are known CRS combinations independent of the BLS secrets, and registration outputs are deterministic public preprocessing. The coefficient argument of [GHK+25, Theorem 1] turns a challenge-mask query into an accepting proof from source-group combinations independent of the fresh encryption variables. Adaptive corruptions reveal only preexisting scalars, preserving this argument. Extract using Lemma 18 for the framework components and Theorem 2 for indexed non-membership.

Its shared selector satisfies $\sum_i b_i = 1$, vanishes outside S^*, vanishes outside the registered set E by Check 2, and obeys $b_i = P_{L^*}(y_i)v_i$ by the INM identities. Thus every nonzero coordinate belongs to an unrevoked recipient, which admissibility requires to remain uncorrupted. For such a user, the source-$\mathbb{G}_2$ span contains no $x_i H(\mathsf{tag}^*)$. Its secret-dependent public auxiliary data occur only in $\mathbb{G}_1$, and decryption publishes no signatures. Comparing coefficients in the extracted DBE signature equation therefore forces every remaining b_i to zero, contradicting normalization. Thus the mask is queried only with negligible probability. Otherwise its fresh pad hides the message. Generic collisions and fresh-tag failures are negligible. □

Efficiency. The live digest retains $O(M^2)$ elements, including all M aggregate INM helper rows and at most $O(nM)$ IIP powers, excluding archived messages and snapshots. Individual INM submissions can be discarded without affecting later updates. Preprocessing is independent of recipient sets and revoked lists.

The registered-slot check adds one master-key element, one $\mathbb{G}_2$ proof element, and one pairing equation. Encryption computes $[Z_S(\tau)]_1$ from S and uses the stored $[Z_E(\tau)]_1$, without reading the full registry.

The CRS and each fully validated non-membership submission contain $O(M^2)$ elements. A direct digest update takes $O(M^3)$ group operations. Given the coefficients of P_L, its encoding and a single-user proof each take $O(|L|+1)$ group operations. Recipient-set polynomial preparation and encoding, context reading, and authorization are separate costs. Ciphertext size excludes explicit S, L fields.

6.2 Revocable Attribute Based Encryption

6.2.1 Definition

We follow [GHK+25, Construction 7], naming Aggregate as Reg and separating auxiliary data from keys. Security covers honest keys and a fixed registry.

Definition 21 (Revocable Attribute Based Encryption)**.** *A revocable ABE scheme Σ for attribute space $\mathcal{U}$, DNF policy space $\mathcal{P}$, message space $\mathcal{M}$, and pseudonym space $\mathcal{Y}$ consists of the following seven PPT algorithms. Indices i enumerate users, with pseudonyms $y_i \in \mathcal{Y}$ and the revocation authority as in Definition 18.*

1. $\mathtt{Setup}(1^\lambda, M) \to (\mathsf{pp}, \mathsf{crs})$*: Generate public parameters and a CRS for security parameter λ and capacity $M > 0$.*
2. $\mathtt{KGen}(\mathsf{crs}) \to (\mathsf{aux}, \mathsf{pk}, \mathsf{sk})$*: Generate public auxiliary data, a public key, and a secret key.*
3. $\mathtt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux}) \to \{0,1\}$*: Deterministically output 1 if the key and auxiliary data are well formed, and 0 otherwise.*
4. $\mathtt{Reg}(\mathsf{crs}, (\mathsf{pk}_i, \mathsf{aux}_i, U_i)_{i\in[\![n]\!]}) \to (\mathsf{mpk}, (\mathsf{hk}_i)_i)$ *or $\perp$: Register keys with attributes $U_i \subseteq \mathcal{U}$. Check every key/auxiliary pair with* IsValid *and deterministically output the master public key and public helpers, or $\perp$ if validation fails or $n > M$.*
5. $\mathtt{Enc}(\mathsf{mpk}, P, L, \mathsf{msg}) \to \mathsf{ct}$*: Encrypt $\mathsf{msg} \in \mathcal{M}$ to users satisfying $P \in \mathcal{P}$ whose pseudonyms lie outside $L \subseteq \mathcal{Y}$, where $|L| \le n$ and n is this key's registry size. Include P, L in the ciphertext.*
6. $\mathtt{Dec}(\mathsf{sk}_i, \mathsf{hk}_i, \mathsf{ct}) \to \mathsf{msg}$ *or $\perp$: Decrypt locally using the registered key and public helper. Output the message or $\perp$.*
7. $\mathtt{Revoke}(y) \to y$*: Publish and output y, and update $L \leftarrow L \cup \{y\}$.*

The scheme satisfies the following properties.

1. ***Key validation.*** *For honest $(\mathsf{pp}, \mathsf{crs})$, a pair $(\mathsf{pk}, \mathsf{aux})$ passes $\mathtt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux})$ exactly when some $(\mathsf{aux}, \mathsf{pk}, \mathsf{sk})$ lies in the support of $\mathtt{KGen}(\mathsf{crs})$. For $n \le M$, registration accepts exactly when every pair passes.*
2. ***Perfect correctness.*** *For every honest setup and registry with attributes $(U_i)_i$, policy $P \in \mathcal{P}$, revoked set L, message msg, and registered user i satisfying $P(U_i) = 1$ and $y_i \notin L$,*

$$\mathtt{Dec}\big(\mathsf{sk}_i, \mathsf{hk}_i, \mathtt{Enc}(\mathsf{mpk}, P, L, \mathsf{msg})\big) = \mathsf{msg}$$

   with probability one over the honest algorithms' randomness. The master key and helpers correspond to those keys and attributes.
3. ***Perfect pseudonym hiding.*** *The scheme satisfies Definition 23.*
4. ***Ciphertext compactness.*** *For P with k conjunctions, ciphertext size is at most $k|\mathsf{msg}| + \mathrm{poly}(\lambda, \log n, |P|)$, excluding the explicit policy and revoked list. The registry, shared parameters, and public helpers are counted separately.*

Definition 22 (Adaptive IND-CPA Security for Revocable ABE)**.** *Consider the following experiment between PPT $\mathcal{A}$ and challenger $\mathcal{C}$, using the algorithms of Definition 21.*

1. *$\mathcal{A}$ chooses a polynomially bounded capacity M and $1 \le n \le M$. $\mathcal{C}$ samples $(\mathsf{pp}, \mathsf{crs}) \leftarrow \mathtt{Setup}(1^\lambda, M)$ and independently samples $(\mathsf{aux}_i, \mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathtt{KGen}(\mathsf{crs})$ for each $i \in [\![n]\!]$. It sends $(\mathsf{pp}, \mathsf{crs}, (\mathsf{pk}_i, \mathsf{aux}_i)_{i\in[\![n]\!]})$ to $\mathcal{A}$. Let y_i denote user i's pseudonym, and initialize $L = R = \mathsf{Cor} = \emptyset$.*

2. *$\mathcal{A}$ may make adaptive $\mathsf{Revoke}(i)$ and $\mathsf{Corrupt}(i)$ queries for $i \in [\![n]\!]$. On $\mathsf{Revoke}(i)$, $\mathcal{C}$ returns $\mathtt{Revoke}(y_i)$, updates $L \leftarrow L \cup \{y_i\}$, and sets $R \leftarrow R \cup \{i\}$. On $\mathsf{Corrupt}(i)$, $\mathcal{C}$ returns sk_i and sets $\mathsf{Cor} \leftarrow \mathsf{Cor} \cup \{i\}$. $\mathcal{A}$ then chooses attribute sets $U_i \subseteq \mathcal{U}$ for all $i \in [\![n]\!]$. $\mathcal{C}$ computes*

$$(\mathsf{mpk}, (\mathsf{hk}_i)_i) \leftarrow \mathsf{Reg}(\mathsf{crs}, (\mathsf{pk}_i, \mathsf{aux}_i, U_i)_{i\in[\![n]\!]})$$

   and publishes the attributes, mpk, and all helpers. The registry, attributes, and registration outputs remain fixed thereafter.

3. *After registration, $\mathcal{A}$ may make further adaptive Revoke and Corrupt queries. The total number of queries is polynomial.*

4. *$\mathcal{A}$ submits a policy $P^* \in \mathcal{P}$, a set $L^* \subseteq \mathcal{Y}$ with $|L^*| \le n$, and equal-length messages $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathcal{M}$. The chosen L^* need not equal the current revoked set. $\mathcal{C}$ samples $b \leftarrow_\$ \{0,1\}$ and sends*

$$\mathsf{ct}^* \leftarrow \mathtt{Enc}(\mathsf{mpk}, P^*, L^*, \mathsf{msg}_b).$$

5. *Queries may continue before $\mathcal{A}$ outputs b'. The challenge's registry, attributes, P^*, and L^* remain fixed, including after subsequent revocations.*

An adversary is admissible if, in every execution, each i in the final Cor satisfies $P^*(U_i) = 0$ or $y_i \in L^*$. Adaptive IND-CPA security requires $|\Pr[b' = b] - 1/2|$ to be negligible in λ for every admissible PPT $\mathcal{A}$, over the randomness of $\mathcal{A}$ and $\mathcal{C}$.

Definition 23 (Perfect Pseudonym Hiding for Revocable ABE)**.** *Fix honest public parameters and public attributes. Honest key generation conditioned on any two possible pseudonym vectors in $\mathcal{Y}^n$ gives identically distributed public views, comprising the CRS, all public keys and auxiliary data, attributes, the master key, and every helper.*

In addition, consider the following experiment.

1. *Perform Steps 1 and 2 of Definition 22, with $2 \le n \le M$. Attributes may instead be fixed publicly at the outset.*

2. *$\mathcal{A}$ may make further adaptive queries to the Revoke and $\mathsf{Corrupt}$ oracles specified in Step 2 of Definition 22. The total number of queries is polynomial.*

3. *$\mathcal{A}$ selects distinct $i_0, i_1 \in [\![n]\!] \setminus (R \cup \mathsf{Cor})$. $\mathcal{C}$ samples $b \leftarrow_\$ \{0,1\}$ and returns y_{i_b} anonymously, without adding either challenge user to R or changing L.*

4. *Revoke and Corrupt queries may continue, and $\mathcal{A}$ outputs b'.*

Admissibility requires i_0, i_1 to remain outside the final $R \cup \mathsf{Cor}$ in every execution. No decryption witness, proof, or outcome is supplied. Perfect pseudonym hiding requires $\Pr[b' = b] = 1/2$ for every admissible PPT $\mathcal{A}$, over all randomness of $\mathcal{A}$ and $\mathcal{C}$.

6.2.2 Construction and Analysis

Construction 7 (Revocable Attribute Based Encryption)**.** *Use the policy decomposition, public helpers, and Appendix B.1 normalization of [GHK+25, Construction 7] with our Appendix A gadgets. Write* $\mathsf{pk}_i^{\mathsf{DBE}} = [x_i]_1$*,* $x_i \in \mathbb{F}_p^\times$*. For* n *users, take* $D = (d_1, \dots, d_n)$ *from the first* n *points of the setup's public enumeration, its* $x^* \notin D$*, and CRS bound* $N = M$*.*

For a literal a*, let* $J(a)$ *contain the users that do not satisfy it.* $J(u) = \{j : u \notin U_j\}$ *and* $J(\neg u) = \{j : u \in U_j\}$*. For a conjunction* ℓ*, the normalized relation* AND_1 *has witness* $b \in \mathbb{F}_p^n$ *and requires*

$$\sum_{j=1}^{n} b_j \mathsf{pk}_j^{\mathsf{DBE}} = [1]_1, \qquad b_j = 0 \quad \textit{for every } j \in \bigcup_{a \in \ell} J(a).$$

This replaces the separate NonZero and DLOG checks in [GHK+25, Construction 7].

Implement the first condition with Lemma 16, fixing its output to $v = [1]_1$*. In particular, its index digest is*

$$\overline{C} = \left[\sum_{j=1}^{n} L_j(x^*)^{-1} x_j L_j(\tau) \right]_1 .$$

Implement each literal condition with Lemma 15 for $S = J(a)$*, using* $Z_{J(a)}(X) = \prod_{j \in J(a)} (X - d_j)$*. All these gadgets share* $\overline{B} = [\sum_j b_j L_j(\tau)]_2$ *and its* X*- and* Y*-shift checks with exponents* $N - n + 1$ *and* N*. For* $n \geq 2$*, the IIP quotient and output checks use shifts* $N - n + 2$ *and* N*, respectively. For* $n = 1$*, use the direct pairing check of that lemma.*

Encrypt each conjunction using witness encryption for

$$\mathsf{rAND} = (\mathsf{AND}_1 \times \mathsf{INM}^{\mathsf{wt}})_{\mathcal{C}},$$

sharing the selector commitment as in Section 6. Lemma 19 gives the compiled scheme. In Construction 3, use X*-shift* $N - n + 1$ *for all six polynomials,* Y*-shift* N *for* B, V*, and* Y*-shift* $N - n + 1$ *for* U, W*. Generate the shifted encodings from the CRS when forming* $\mathsf{hk}_i^{\mathsf{INM}}$*.*

- $\mathtt{Setup}(1^\lambda, M)$*: Use Construction 6.*
- $\mathtt{KGen}(\mathsf{crs})$*: Use Construction 6.*
- $\mathtt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux})$*: Use Construction 6.*
- $\mathtt{Reg}(\mathsf{crs}, (\mathsf{pk}_i, \mathsf{aux}_i, U_i)_{i \in [\![n]\!]})$*: Reject if* $n \notin [\![M]\!]$ *or any key/auxiliary pair fails* $\mathtt{IsValid}$*. Set* $U_{\mathrm{eff}} = \bigcup_i U_i$ *and form the literal dictionary of [GHK+25, Construction 7] using the component digests above on* D*. For each user* i*, precompute the IIP proof for* $(e_i, \mathsf{pk}_i^{\mathsf{DBE}})$*, the selector commitment* $[L_i(\tau)]_2$*, and its two shifts. Include* $[L_i(\tau)/Z_{J(a)}(\tau)]_2$ *for each satisfied literal* a*, and* $\perp$ *otherwise. For literals outside* U_{eff}*, use* $J(a) = [\![n]\!]$ *for positive literals and* $J(a) = \emptyset$ *for negative literals. Run* $\mathsf{INM}.\mathtt{Digest}$ *on the batch and combine its verifier data with the authorization data to form* mpk*. Append* $(\widehat{\mathsf{aux}}_{i,c})_{c<n}$ *and* $\mathsf{hk}_i^{\mathsf{INM}}$ *to each user's helper. Output* $(\mathsf{mpk}, (\mathsf{hk}_i)_i)$ *with the public attribute metadata.*
- $\mathtt{Enc}(\mathsf{mpk}, P, L, \mathsf{msg})$*: For each conjunction* ℓ *of* P*, encrypt* msg *with the compiled gadgets for instance* (ℓ, L) *of* rAND*. Output these component ciphertexts together with* P, L*, recorded once.*
- $\mathtt{Dec}(\mathsf{sk}_i, \mathsf{hk}_i, \mathsf{ct})$*: Read* P, L *from the ciphertext. Return* $\perp$ *if* $y_i \in L$ *or no conjunction of* P *is satisfied by* U_i*. Otherwise choose a satisfied conjunction. Multiply the helper's selector commitment, selector shifts, IIP proof elements, and selected Zero quotients by* x_i^{-1}*. Set* $b = x_i^{-1} e_i$*. Set* $t_i = b_i y_i$*,* $s_i = b_i r_i$*, with zero entries elsewhere, and invoke the indexed non-membership prover on* (b, t, s) *using the retained row and helper. Combine this proof with the scaled authorization proof, using their common selector commitment and sharing repeated degree checks, and decrypt the chosen component. Output the resulting message or* $\perp$*.*

- `Revoke`(y)*: Publish and output* y*, and update* $L \leftarrow L \cup \{y\}$.

Decryption witnesses and proofs remain private.

Lemma 10 (Correctness, security, and hiding)**.** *Under honest setup and registration and the input conditions of Lemma 18, Construction 7 satisfies Definition 21. It satisfies Definition 22 in the generic bilinear-group and random-oracle models, using Theorem 1 and Lemma 20.*

Proof. Key validation follows from Lemma 8. A satisfied literal has $i \notin J(a)$, making $L_i(X)/Z_{J(a)}(X)$ a polynomial. For $J(a) = \emptyset$, its encoding is the selector commitment. An eligible user i scales its unit-selector proofs by x_i^{-1} to obtain $b = x_i^{-1} e_i$, IIP output $[1]_1$, and the literal checks. This works because the equations are homogeneous before fixing the output. Its opening and $P_L(y_i) \neq 0$ give an indexed non-membership witness with the same b. Component and compiler completeness give perfect decryption.

For security, replace the nonzero key scalars x_i by uniform scalars and abort if any $[x_i]_1 = [0]_1$. This changes the experiment by at most $n/|\mathbb{F}_p|$. In every hybrid, abort with a random output bit if the challenge is inconsistent with earlier corruptions or before answering a corruption of an eligible challenge user. Admissibility leaves both real endpoints unchanged. A polynomial hybrid over conjunctions reduces distinguishing to one component, while the reduction generates the others. Distinguishing requires querying that component's random-oracle mask, except with negligible probability.

Apply the coefficient argument of Lemma 9 to the full transcript, treating its fresh coins as formal variables. Record all generic operations and later Revoke and Corrupt replies. These disclose only preexisting key scalars, which are substituted as in Section A.1, preserving the source-group bounds. The other ciphertext headers use known coins and are public scalar combinations of preexisting source-group inputs. Comparing coefficients therefore recovers an accepting proof from source-group combinations independent of the selected component's fresh coins, even with post-challenge corruptions. Lemma 18 and Theorem 2 then extract a common selector b satisfying

$$\sum_{i=1}^{n} x_i b_i = 1, \qquad b_i \neq 0 \implies P^*(U_i) = 1 \text{ and } y_i \notin L^*.$$

In particular, $b \neq 0$, and every index in its support is outside the adversary's final corrupted set.

This extraction breaks adaptive IIP.`AuxGen` hardness in Definition 15, established for the extended CRS and full public view by Lemma 20. Obtain IIP key/auxiliary pairs from the Auxgen oracle, sample (y_i, r_i) independently, and compute revocation data and registration outputs publicly. Answer Revoke queries with sampled pseudonyms and Corrupt queries through the inversion oracle, before and after the challenge. After extraction, choose j with $b_j \neq 0$. Since x_j was never requested, query every other unknown x_i and output

$$\left(j,\ b_j^{-1}\Big(1 - \sum_{i \neq j} x_i b_i\Big) \right).$$

The displayed scalar equals x_j, while j remains outside the inversion oracle's queried set. Queries are allowed until final output, so j need not be guessed in advance. A nonnegligible distinguishing advantage would therefore contradict Lemma 20, after only the polynomial conjunction-hybrid loss and negligible zero-key, generic-collision, and extraction errors.

The hiding argument of Lemma 8 applies for each fixed attribute assignment. Adaptively choosing attributes from the public view and previous disclosures, then publicly computing registration outputs, preserves the conditional independence and uniformity of undisclosed pseudonyms. The same swap therefore proves Definition 23. The ABE scheme of [GHK+25, Construction 7] encrypts once per conjunction with constant-size gadgets per literal. INM adds constant ciphertext overhead per conjunction, proving ciphertext compactness. □

Efficiency. Beyond the shared $O(M^2)$ CRS, $n \leq M$ users retain $O(n^2)$ public non-membership helpers. Key generation and validation use an $O(M^2)$ grid per user, totaling $O(nM^2)$ submitted elements discarded after forming the batch digest. Given P_L and its encoding, revocation adds constant ciphertext and encryption group-operation overhead per conjunction and $O(|L| + 1)$ group operations per eligible user's proof. List-polynomial preparation, explicit P, L context, and underlying ABE costs [GHK+25, Construction 7] are separate. Registration preprocessing is independent of the policy and revoked list. This ABE aggregation provides no incremental update rule.

6.3 Revocable Threshold Encryption

We extend DBE using [GHK+25, Constructions 8–9] so that decryption requires public contributions from at least t unrevoked committee members. For a unit selector and $L = \{y\}$, an unmasked proof satisfies $(y_i - y)\overline{V} = \overline{B}$, which lets observers test later-published pseudonyms against it. We mask the proofs while preserving aggregation.

6.3.1 Definition

We extend [GHK+25, Definition 19] with revocation and public partial decryptions, using DBE's notation and honestly generated keys.

Definition 24 (Revocable Threshold Encryption)**.** *A revocable threshold-encryption scheme has message space $\mathcal{M}$, identity space $\mathcal{ID}$, pseudonym space $\mathcal{Y}$, and nine PPT algorithms. Each user samples a pseudonym y_i from $\mathcal{Y}$. The revocation authority maintains the user-to-pseudonym association and the initially empty public revoked set $L \subseteq \mathcal{Y}$. Public parameters* pp *are implicit inputs.*

1. $\mathtt{Setup}(1^\lambda, M) \to (\mathsf{pp}, \mathsf{crs})$*: Generate public parameters and a CRS for capacity $M > 0$.*
2. $\mathtt{KGen}(\mathsf{crs}) \to (\mathsf{aux}, \mathsf{pk}, \mathsf{sk})$*: Generate a public key, secret key, and public auxiliary data.*
3. $\mathtt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux}) \to \{0, 1\}$*: Check that the key and auxiliary data are well formed.*
4. $\mathtt{Reg}(\mathsf{crs}, \mathsf{dd}, \mathsf{id}, \mathsf{pk}, \mathsf{aux}) \to (\mathsf{mpk}', \mathsf{dd}')$ *or $\perp$: Validate and register the key under* id*, updating the master public key and public decryption digest. Reject invalid keys, identities outside $\mathcal{ID}$, repeated identities, and registrations exceeding capacity. Initially* $\mathsf{dd} = \perp$.
5. $\mathtt{Enc}(\mathsf{mpk}, S, t, L, \mathsf{msg}) \to \mathsf{ct}$*: Encrypt* $\mathsf{msg} \in \mathcal{M}$ *for at least t unrevoked members of a registered committee S, where $1 \leq t \leq |S|$ and $L \subseteq \mathcal{Y}$ has size at most the registry size m. The ciphertext context fixes the registry, S, t, L, and a fresh public tag.*
6. $\mathsf{PartDec}(\mathsf{dd}, \mathsf{id}, \mathsf{sk}, \mathsf{ct}) \to \delta$ *or $\perp$: Produce a labelled partial decryption, or $\perp$ if ineligible.*
7. $\mathsf{PartVerify}(\mathsf{dd}, \mathsf{ct}, \delta) \to \{0, 1\}$*: Accept exactly when the partial's verification checks hold.*
8. $\mathsf{DecAggr}(\mathsf{crs}, \mathsf{dd}, S, (\delta_{\mathsf{id}})_{\mathsf{id} \in T}, \mathsf{ct}) \to \mathsf{msg}$ *or $\perp$: Combine verified partials from at least t distinct committee members to recover the message, or output $\perp$.*
9. $\mathtt{Revoke}(y) \to y$*: Publish and output y, and update $L \leftarrow L \cup \{y\}$.*

The public digest retains registered identities, keys, and reusable prover data. Ciphertexts and partials use their original registration outputs. The scheme satisfies:

1. ***Key validation.*** *For every honest* $(\mathsf{pp}, \mathsf{crs})$*,* $\mathtt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux}) = 1$ *exactly when some* $(\mathsf{aux}, \mathsf{pk}, \mathsf{sk})$ *lies in the support of* $\mathtt{KGen}(\mathsf{crs})$*. Every such pair can be registered under an unused identity in $\mathcal{ID}$ when capacity remains.*

2. ***Perfect correctness.*** *For every honest setup and registry, committee S of registered identities, threshold $1 \le t \le |S|$, revoked set $L \subseteq \mathcal{Y}$ with $|L| \le m$, where m is the registry size, and message* msg*, let* $\mathsf{ct} \leftarrow \mathtt{Enc}(\mathsf{mpk}, S, t, L, \mathsf{msg})$*. Every eligible identity* $\mathsf{id}_i \in S$ *with* $y_i \notin L$ *produces a nonfailure partial* $\delta_{\mathsf{id}_i} \leftarrow \mathsf{PartDec}(\mathsf{dd}, \mathsf{id}_i, \mathsf{sk}_i, \mathsf{ct})$ *satisfying* $\mathsf{PartVerify}(\mathsf{dd}, \mathsf{ct}, \delta_{\mathsf{id}_i}) = 1$*. For every set T of at least t distinct eligible identities and their honestly generated partials,*

$$\mathsf{DecAggr}\big(\mathsf{crs}, \mathsf{dd}, S, (\delta_{\mathsf{id}})_{\mathsf{id}\in T}, \mathsf{ct}\big) = \mathsf{msg}.$$

Both requirements hold with probability one over honest randomness.

3. ***Historical pseudonym hiding.*** *The scheme satisfies Definition 26 below, which includes public partial decryptions.*

4. ***Compactness.*** *The master public key has size* $\mathrm{poly}(\lambda, \log M)$ *beyond shared public parameters. Ciphertext size is at most* $|\mathsf{msg}| + \mathrm{poly}(\lambda)$*, excluding the explicit committee and revoked list, and each partial decryption has size* $\mathrm{poly}(\lambda)$*. Identity labels, the registry, auxiliary arrays, and reusable public decryption digest are counted separately.*

Definition 25 (Adaptive IND-CPA Security)**.** *For a scheme as in Definition 24, define the experiment between PPT $\mathcal{A}$ and challenger $\mathcal{C}$.*

1. *$\mathcal{A}$ chooses a positive polynomially bounded capacity n. $\mathcal{C}$ samples* $(\mathsf{pp}, \mathsf{crs}) \leftarrow \mathtt{Setup}(1^\lambda, n)$ *and sends them to $\mathcal{A}$, which chooses distinct identities* $\mathsf{id}_1, \ldots, \mathsf{id}_n \in \mathcal{ID}$*. Starting with* $\mathsf{dd}_0 = \perp$*, for $i = 1, \ldots, n$, $\mathcal{C}$ runs* $(\mathsf{aux}_i, \mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathtt{KGen}(\mathsf{crs})$ *and* $(\mathsf{mpk}_i, \mathsf{dd}_i) \leftarrow \mathtt{Reg}(\mathsf{crs}, \mathsf{dd}_{i-1}, \mathsf{id}_i, \mathsf{pk}_i, \mathsf{aux}_i)$*. It sends every* $(\mathsf{aux}_i, \mathsf{pk}_i, \mathsf{mpk}_i, \mathsf{dd}_i)$ *to $\mathcal{A}$ and retains the secret keys and associated pseudonyms y_i. Set $E = \{\mathsf{id}_1, \ldots, \mathsf{id}_n\}$ and fix* $(\mathsf{mpk}, \mathsf{dd}) = (\mathsf{mpk}_n, \mathsf{dd}_n)$ *for the rest of the experiment. Initialize the revoked set $L = \emptyset$, the revoked-identity and corrupted-identity sets $R = U = \emptyset$, and the partial-query history $\mathcal{H} = \emptyset$.*

2. *$\mathcal{A}$ makes polynomially many adaptive queries. On* $\mathsf{Revoke}(\mathsf{id}_i)$*, $\mathcal{C}$ returns* $\mathtt{Revoke}(y_i)$ *and sets $R \leftarrow R \cup \{\mathsf{id}_i\}$. On* $\mathsf{Corrupt}(\mathsf{id}_i)$*, it returns* sk_i *and sets $U \leftarrow U \cup \{\mathsf{id}_i\}$. On* $\mathsf{Partial}(\mathsf{id}_i, \mathsf{ct})$*, it samples $\delta \leftarrow$* $\mathsf{PartDec}(\mathsf{dd}, \mathsf{id}_i, \mathsf{sk}_i, \mathsf{ct})$ *with fresh randomness and returns δ. If $\delta \neq \perp$, it adds* $(\mathsf{id}_i, \mathsf{ctx})$ *to $\mathcal{H}$, where* ctx *is the context in* ct*. Queries with an identity outside E return $\perp$ without changing any records.*

3. *$\mathcal{A}$ supplies $S^* \subseteq E$, an integer $1 \le t^* \le |S^*|$, a set $L^* \subseteq \mathcal{Y}$ with $|L^*| \le n$, and equal-length messages* $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathcal{M}$*. The chosen L^* need not equal L. $\mathcal{C}$ samples $b \leftarrow_\$ \{0,1\}$ and generates* $\mathsf{ct}^* \leftarrow \mathtt{Enc}(\mathsf{mpk}, S^*, t^*, L^*, \mathsf{msg}_b)$ *with fresh context* ctx^**. Set* $J^* = \{\mathsf{id} \in E : (\mathsf{id}, \mathsf{ctx}^*) \in \mathcal{H}\}$ *and* $\mathsf{Elig}^* = \{\mathsf{id}_i \in S^* : y_i \notin L^*\}$*. Require*

$$|\mathsf{Elig}^* \cap (U \cup J^*)| < t^*. \tag{7}$$

If this fails, abort and set b' to an independent fair bit. Otherwise send ct^* *to $\mathcal{A}$.*

4. *$\mathcal{A}$ continues querying the oracles of step 2. Refuse any query that would violate (7) before answering or updating records. After each permitted query, update U, $\mathcal{H}$, and J^*. Every successful partial for* ctx^* *counts toward J^*, including earlier releases and queries with different encrypted components. Count each identity once. The registry and S^*, t^*, L^* remain fixed after later revocations.*

5. *$\mathcal{A}$ outputs b'.*

Security requires $|\Pr[b' = b] - 1/2|$ *to be negligible in λ for every PPT $\mathcal{A}$, including aborted executions, over all randomness of $\mathcal{A}$ and $\mathcal{C}$.*

Definition 26 (Historical Pseudonym Hiding)**.** *For a scheme as in Definition 24, consider the following experiment between PPT $\mathcal{A}$ and challenger $\mathcal{C}$.*

1. *Run step 1 of Definition 25 with $n \geq 2$ and initialize an empty table $\mathcal{T}$ of ciphertexts.*
2. *$\mathcal{A}$ makes polynomially many adaptive queries. Answer Revoke and Corrupt queries as in step 2 of Definition 25. On $\mathsf{NewContext}(S, t, \mathsf{msg})$ with $S \subseteq E$, integer $1 \leq t \leq |S|$, and $\mathsf{msg} \in \mathcal{M}$, set $L = \{y_i : \mathsf{id}_i \in R\}$ and sample $\mathsf{ct} \leftarrow \mathtt{Enc}(\mathsf{mpk}, S, t, L, \mathsf{msg})$. Store $\mathcal{T}[c] = \mathsf{ct}$ under a fresh identifier c and return (c, ct). On $\mathsf{Partial}(\mathsf{id}_i, c)$ with $\mathsf{id}_i \in E$ and c in the table, sample $\delta \leftarrow \mathsf{PartDec}(\mathsf{dd}, \mathsf{id}_i, \mathsf{sk}_i, \mathcal{T}[c])$ with fresh randomness and return δ, including $\perp$. Other queries return $\perp$ without changing any records.*
3. *$\mathcal{A}$ chooses distinct $\mathsf{id}_{i_0}, \mathsf{id}_{i_1} \in E \setminus (R \cup U)$. $\mathcal{C}$ samples $b \leftarrow_{\$} \{0,1\}$ and returns $\mathtt{Revoke}(y_{i_b})$, without adding the challenge identity to R. Freeze $\mathcal{T}$ at this point.*
4. *Revoke, Corrupt, and Partial queries continue as in step 2, using the unchanged ciphertexts in $\mathcal{T}$ and their original lists. No further NewContext queries are answered. Public computations remain unrestricted. Corruption reveals the full secret key but no past partial-generation coins.*
5. *$\mathcal{A}$ outputs b'.*

Admissibility requires both challenge identities to remain outside the final $R \cup U$ in every execution. Historical hiding requires $|\Pr[b' = b] - 1/2|$ to be negligible in λ for every admissible PPT $\mathcal{A}$, over all randomness of $\mathcal{C}$ and $\mathcal{A}$.

6.3.2 Construction and Analysis

Use Construction 6's fixed domain $D = \mathcal{ID}$ of size $n = M$, with $m \leq M$ registered users and $x^*, \tau \notin D$. All degree bounds use n, and the security experiments use a full registry.

Definition 27 (Extended Public Indexed Non-membership Relation)**.** *Let $\widetilde{\mathsf{INM}}$ have the same parameters, index, and instance as INM. A witness (b, y, r, t_*, s_*) is valid exactly when (b, y, r) is valid for INM and $t_*, s_* \in \mathbb{F}_p$. Define $\widetilde{\mathsf{INM}}^{\mathsf{wt}}$ by*

$$(b, y, r, t_*, s_*) \longmapsto (b, \widetilde{t}, \widetilde{s}), \qquad \widetilde{t} = ((b_i y_i)_{i \in [\![n]\!]}, t_*), \quad \widetilde{s} = ((b_i r_i)_{i \in [\![n]\!]}, s_*).$$

The last coordinates are indexed by x^*. Recover (y_i, r_i) from $(b_i, \widetilde{t}_i, \widetilde{s}_i)$ as in Definition 14, retaining t_*, s_*.

Construction 8 (Masked indexed non-membership)**.** *Use degree bound $N = n + 1$. Write $p_i(Y, Z) = Y - y_i + r_i Z$ and $\tilde{p}(X, Y, Z) = \sum_i p_i(Y, Z) L_i(X)$. Let $B(X)$ interpolate b on D and $\widetilde{T}(X), \widetilde{S}(X)$ interpolate $\widetilde{t}, \widetilde{s}$ on $D \cup \{x^*\}$. Commit componentwise using deterministic SCS commitments at τ.*

$$\mathtt{Commit}_{\widetilde{\mathsf{INM}}}(\mathsf{crs}, b, \widetilde{t}, \widetilde{s}) = \big([B(\tau)]_2, [\widetilde{T}(\tau)]_2, [\widetilde{S}(\tau)]_2\big).$$

Denote this commitment scheme by $\widetilde{\mathsf{C}}_{\mathsf{INM}}$. The committed relation is $[\widetilde{\mathsf{INM}}^{\mathsf{wt}}]_{\widetilde{\mathsf{C}}_{\mathsf{INM}}}$.

Use Construction 3's $\mathtt{Gen}$, $\mathtt{AuxGen}$, and $\mathtt{Digest}$ with bound N. Include $C_0 = [\tilde{p}(\tau, \theta, \alpha)]_1$ and the shifted helpers in Pk, and the CRS encodings needed for checks 3–6 in Vk.

- $\mathtt{Prove}\big(\mathsf{Pk}, ((\overline{B}, \overline{\widetilde{T}}, \overline{\widetilde{S}}), L), (b, \widetilde{t}, \widetilde{s})\big)$*: Recover the selected openings as in Definition 27. Put $T(X) = \sum_i b_i y_i L_i(X)$ and $S(X) = \sum_i b_i r_i L_i(X)$, and compute*

$$\eta = \frac{t_* - T(x^*)}{Z_D(x^*)}, \qquad \zeta = \frac{s_* - S(x^*)}{Z_D(x^*)}.$$

Thus the committed polynomials are

$$\begin{aligned} \widetilde{T}(X) &= T(X) + \eta Z_D(X), \\ \widetilde{S}(X) &= S(X) + \zeta Z_D(X). \end{aligned} \tag{8}$$

Compute the ordinary proof polynomials from Construction 3. Independently sample $A, K, \beta \leftarrow_{\$} \mathbb{F}_p$ and set

$$\begin{aligned} \widetilde{U}(X,Y) &= U(X,Y) + Z_D(X)A, \\ \widetilde{V}(X) &= V(X) + Z_D(X)\beta, \\ \widetilde{W}(X,Y) &= W(X,Y) + Z_D(X)K. \end{aligned} \tag{9}$$

Using the ordinary quotients from (3), compute

$$\begin{aligned} \overline{\widetilde{Q}}_{\mathrm{nm}} &= \overline{Q}_{\mathrm{nm}} + AC_0 + \beta[P_L(\theta)]_1 - K[\alpha]_1, \\ \overline{\widetilde{Q}}_{\mathrm{open}} &= \overline{Q}_{\mathrm{open}} + \eta[1]_1 - \zeta[\alpha]_1, \end{aligned} \tag{10}$$

where $\overline{Q}_{\mathrm{nm}} = [Q_{\mathrm{nm}}(\tau,\theta,\alpha)]_1$ and $\overline{Q}_{\mathrm{open}} = [Q_{\mathrm{open}}(\tau,\theta,\alpha)]_1$.

For $\mathcal{F} = \{B, \widetilde{T}, \widetilde{S}, \widetilde{U}, \widetilde{V}, \widetilde{W}\}$, set $\overline{F} = [F(\tau,\theta)]_2$ and form the shifts specified by checks 3–6. Output

$$\pi = \big(\overline{\widetilde{U}}, \overline{\widetilde{V}}, \overline{\widetilde{W}}, \overline{\widetilde{Q}}_{\mathrm{nm}}, \overline{\widetilde{Q}}_{\mathrm{open}}, (\overline{F}_X, \overline{F}_Y)_{F\in\mathcal{F}}\big).$$

- $\mathtt{Verify}\big(\mathsf{Vk}, ((\overline{B}, \overline{\widetilde{T}}, \overline{\widetilde{S}}), L), \pi\big)$*: Reject unless $L \subseteq \mathbb{F}_p$ and $|L| \le n$. Compute $[P_L(\theta)]_1$ from L and the CRS, parse π, and check*

  1. $[\tilde{p}(\tau,\theta,\alpha)]_1 \circ \overline{\widetilde{U}} - [\alpha]_1 \circ \overline{\widetilde{W}} + [P_L(\theta)]_1 \circ \overline{\widetilde{V}} - [1]_1 \circ \overline{B} = \overline{\widetilde{Q}}_{\mathrm{nm}} \circ [Z_D(\tau)]_2$.
  2. $\big([\tilde{p}(\tau,\theta,\alpha)]_1 - [\theta]_1\big) \circ \overline{B} + [1]_1 \circ \overline{\widetilde{T}} - [\alpha]_1 \circ \overline{\widetilde{S}} = \overline{\widetilde{Q}}_{\mathrm{open}} \circ [Z_D(\tau)]_2$.
  3. $[\tau^{N-n+1}]_1 \circ \overline{B} = [1]_1 \circ \overline{B}_X$.
  4. $[\tau^{N-n}]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_X$ *for* $F \in \{\widetilde{T}, \widetilde{S}, \widetilde{U}, \widetilde{V}, \widetilde{W}\}$.
  5. $[\theta^N]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_Y$ *for* $F \in \{B, \widetilde{T}, \widetilde{S}, \widetilde{V}\}$.
  6. $[\theta^{N-n+1}]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_Y$ *for* $F \in \{\widetilde{U}, \widetilde{W}\}$.

  Accept exactly when all checks hold.

Lemma 11 (Masked indexed non-membership)**.** *Under honest setup and auxiliary generation, the source-$\mathbb{G}_2$ conditions of Theorem 2, and $N \ge n+1$, Construction 8 is perfectly complete and admits polynomial-time straight-line extraction for $[\widetilde{\mathsf{INM}}^{\mathsf{wt}}]_{\widetilde{\mathsf{C}}_{\mathsf{INM}}}$. Recovering the openings gives a witness (b, y, r, t_*, s_*) for Definition 27. Each selected opening equals the auxiliary-generation input and satisfies $b_i = P_L(y_i)\widetilde{V}(d_i)$.*

Proof. By Theorem 2's coefficient argument, the positive shifts in checks 3–6 extract six polynomials. Their X-degrees are at most $n-1$ for B and n otherwise. The Y-degree is zero for $B, \widetilde{T}, \widetilde{S}, \widetilde{V}$ and at most $n-1$ for $\widetilde{U}, \widetilde{W}$. The separate checks for $\widetilde{T}, \widetilde{S}$ are needed because the opening identity alone permits $Z_D(X)$ times a polynomial in Y. All shifts fit the CRS for $N \ge n+1$ and exclude additional independent source-$\mathbb{G}_2$ terms.

Write (y_i^0, r_i^0) for the auxiliary-generation inputs and $p_i^0(Y,Z) = Y - y_i^0 + r_i^0 Z$. Outside negligible generic-group collisions, checks 1 and 2 give formal identities. At $X = d_i$, their quotient terms vanish, giving

$$(-y_i^0 + r_i^0 Z)B(d_i) + \widetilde{T}(d_i) - Z\widetilde{S}(d_i) = 0.$$

Comparing its constant and Z coefficients yields

$$\widetilde{T}(d_i) = B(d_i)y_i^0, \qquad \widetilde{S}(d_i) = B(d_i)r_i^0.$$

Both weighted coordinates vanish when $b_i := B(d_i) = 0$. Otherwise, division by b_i recovers the index opening $y_i = y_i^0$ and $r_i = r_i^0$. The non-membership identity in $\mathbb{F}_p[Y, Z]$ gives

$$p_i^0(Y,Z)\widetilde{U}(d_i,Y) - Z\widetilde{W}(d_i,Y) + P_L(Y)\widetilde{V}(d_i) = B(d_i).$$

Evaluating at $(Y, Z) = (y_i^0, 0)$ gives $b_i = P_L(y_i^0)\widetilde{V}(d_i)$, so every selected y_i is outside L.

Evaluate B on D and $\widetilde{T}, \widetilde{S}$ on $D \cup \{x^*\}$ to recover the committed witness, with free coordinates $t_* := \widetilde{T}(x^*)$ and $s_* := \widetilde{S}(x^*)$. Set unselected openings to zero. The degree bounds give the original commitments and their computational binding under τ.

For completeness, fix a valid extended witness. The degree bounds and agreement on D make $\widetilde{T} - T$ and $\widetilde{S} - S$ scalar multiples of Z_D. Evaluation at x^* gives η, ζ in (8). By Lemma 4, the ordinary prover satisfies (4)–(5) at $Z = \alpha$, including for alternative valid openings.

Adding the scalar masks gives

$$\widetilde{Q}_{\mathrm{open}}(X,Y,Z) = Q_{\mathrm{open}}(X,Y,Z) + \eta - Z\zeta,$$
$$\widetilde{Q}_{\mathrm{nm}}(X,Y,Z) = Q_{\mathrm{nm}}(X,Y,Z) + \tilde{p}(X,Y,Z)A + P_L(Y)\beta - ZK.$$

Evaluation at (τ, θ, α) proves checks 1 and 2. The ordinary quotients use retained helper rows, and (10) uses only C_0 and the CRS for corrections. The shifts fit the CRS, proving perfect completeness. All fourteen checks are linear in the commitments and proof. □

Contexts. Fix an efficiently computable injective canonical encoding enc of $\mathsf{ctx} = (\mathsf{registryID}, S, t, L, \mathsf{tag})$, with 2λ-bit registry identifiers and λ-bit tags. It uses $O(\lambda + (|S| + |L|)\log p + \log M)$ bits. Parsers reject malformed or noncanonical encodings, repeated set elements, and trailing data. Contexts match exactly when their encodings match. A context is valid for a digest with registered set E when its registry identifier matches exactly, $S \subseteq E$, $1 \le t \le |S|$, $L \subseteq \mathcal{Y}$, $|L| \le |E|$, and its tag has the prescribed length. All decryption algorithms reject invalid contexts.

Definition 28 (Revocable Threshold Signature Relation)**.** *Use DBE's random oracle $H : \{0,1\}^* \to \mathbb{G}_2$. For instance (S, t, L, ξ), require $\xi = \mathsf{enc}(\mathsf{registryID}, S, t, L, \mathsf{tag})$ with the canonical encoding above, the public registry identifier* registryID *and* tag $\in \{0,1\}^\lambda$*. Route (σ^*, b) to* DBE *with instance (ξ, S) and (b, y, r, t_*, s_*) to* $\widetilde{\mathsf{INM}}$ *with instance L, sharing b. Apply the registered-slot check 2 to $B(X)$ for the registered set E. Replace DBE's sum-one normalization by $\deg B(X) \le M - t$ and $B(x^*) = 1$, where $B(X)$ interpolates b on D.*

Construction 9 (Revocable Threshold Encryption)**.** *Enforce Definition 28's normalization by the IIP check with public index $([L_i(x^*)]_1)_{i \in [\![n]\!]}$, output $[1]_1$, and CRS-derived auxiliary powers. Enforce $\deg B \le M - t$ using Appendix A. Compile* $\widetilde{\mathsf{INM}}^{\mathsf{wt}}$ *with commitment $\widetilde{\mathsf{C}}_{\mathsf{INM}}$ and the authorization gadgets, sharing the $B(X)$ commitment, to obtain* WE.

- $\mathtt{Setup}(1^\lambda, M)$*: Run DBE setup with degree bound $N = M + 1$ and $|\mathcal{ID}| = M$. Set $\mathcal{Y} = \mathbb{F}_p$, initialize $L = \emptyset$, and output* $(\mathsf{pp}, \mathsf{crs})$.
- $\mathtt{KGen}(\mathsf{crs})$*: Run DBE key generation with these M, N, retaining IIP powers for $1 \le e \le N$ and the non-membership grid for $0 \le a, c < M$. Output its tuple.*
- $\mathtt{IsValid}(\mathsf{crs}, \mathsf{pk}, \mathsf{aux})$*: Return DBE validation with the same M, N.*
- $\mathtt{Reg}(\mathsf{crs}, \mathsf{dd}, \mathsf{id}, \mathsf{pk}, \mathsf{aux})$*: Run DBE registration, retaining C_0 from the non-membership digest. Assemble the threshold digests with the normalization and degree checks above and the registered-slot check 2 using $[Z_E(\tau)]_1$. After successful validation and assembly, independently sample* registryID $\leftarrow_\$ \{0,1\}^{2\lambda}$*. Output* $\mathsf{mpk}' = (m, \mathsf{registryID}, \mathsf{ed}_{\mathsf{WE}})$ *and a public* dd' *containing the same identifier, active records, retained IIP powers, and aggregate prover data including C_0. The identifier names this registry snapshot. Later registrations assign fresh identifiers, while ciphertexts and partials retain their original outputs. Identities keep their fixed slots in D.*

- $\texttt{Enc}(\mathsf{mpk}, S, t, L, \mathsf{msg})$*: Read* $(m, \mathsf{registryID}, \mathsf{ed}_{\mathsf{WE}})$ *from* mpk*, with* $|L| \le m$ *and* S *containing only registered identities. Sample* $\mathsf{tag} \leftarrow_{\$} \{0,1\}^{\lambda}$ *and set*

$$\mathsf{ctx} \leftarrow (\mathsf{registryID}, S, t, L, \mathsf{tag}), \qquad \xi \leftarrow \mathsf{enc}(\mathsf{ctx}),$$

Output

$$\mathsf{ct} \leftarrow (\mathsf{ctx}, \mathsf{WE}.\texttt{Enc}(\mathsf{ed}_{\mathsf{WE}}, (S, t, L, \xi), \mathsf{msg})).$$

- $\mathsf{PartDec}(\mathsf{dd}, \mathsf{id}, \mathsf{sk}, \mathsf{ct})$*: Read* ctx *from* ct*. Return* $\perp$ *unless* $\mathsf{id} = \mathsf{id}_i \in S$ *and* $y_i \notin L$*. Set* $\xi = \mathsf{enc}(\mathsf{ctx})$ *and form the single-key DBE witness with* $b = e_i$ *and* $\sigma_i = x_i H(\xi)$*. Use opening* (y_i, r_i) *at* i *and zero elsewhere. Independently sample* $\eta, \zeta \leftarrow_{\$} \mathbb{F}_p$*, set* $t_* = \widetilde{T}(x^*)$ *and* $s_* = \widetilde{S}(x^*)$ *by (8), and run Construction 8's prover on* $(b, \widetilde{t}, \widetilde{s})$ *using* dd*. Output* $\delta_{\mathsf{id}_i} = (\mathsf{id}_i, \sigma_i, \rho_i)$*, where* ρ_i *contains the masked commitments and proof with the public selector and its two shifts omitted. Partials depend only on the context, not the encrypted component.*

- $\mathsf{PartVerify}(\mathsf{dd}, \mathsf{ct}, \delta)$*: Parse* $\delta = (\mathsf{id}_i, \sigma_i, \rho_i)$ *and the ciphertext context* $(\mathsf{registryID}, S, t, L, \mathsf{tag})$*. Check*

  1. *The context is valid for* dd *and* $\mathsf{id}_i \in S$*.*
  2. *Look up the registered key and public data for* id_i*. Set* $\xi = \mathsf{enc}(\mathsf{ctx})$ *and* $h = H(\xi)$*, and check*

$$[1]_1 \circ \sigma_i = \mathsf{pk}_i^{\mathsf{DBE}} \circ h.$$

  3. *Restore* $\overline{B} = [L_i(\tau)]_2$ *and its two shifts in* ρ_i*. Run Construction 8's verifier with this registry's key and list* L*, omitting the two restored selector-shift checks.*

  Output 1 *exactly when all checks pass, and* 0 *otherwise.*

- $\mathsf{DecAggr}(\mathsf{crs}, \mathsf{dd}, S, (\delta_{\mathsf{id}})_{\mathsf{id} \in T}, \mathsf{ct})$*: Read and validate the context in* ct*. Retain verified partials from* S*, at most one per identity. For their identity set* T*, output* $\perp$ *if* $|T| < t$*. Otherwise set*

$$B_T(X) = \prod_{d \in D \setminus T} \frac{X - d}{x^* - d}, \quad b_i = B_T(d_i), \quad \sigma^* = \sum_{i:\mathsf{id}_i \in T} b_i \sigma_i.$$

  Aggregate every coordinate as $\rho^* = \sum_{i:\mathsf{id}_i \in T} b_i \rho_i$*. For* $\xi = \mathsf{enc}(\mathsf{ctx})$*, form the remaining public routed proofs for* (S, t, L, ξ)*, including* $[\tau^{t+1} B_T(\tau)]_2$ *and* $[Q_E(\tau)]_2$*, where* $Q_E(X) = Z_E(X) B_T(X) / Z_D(X)$ *for* $T \subseteq E$*. Decrypt using the resulting compiler proof.*

- $\texttt{Revoke}(y)$*: Publish and output* y*, and update* $L \leftarrow L \cup \{y\}$*.*

Theorem 4. *In the generic bilinear-group and random-oracle models, Construction 9 satisfies Definition 24 and provides adaptive IND-CPA security and historical pseudonym hiding under Definitions 25 and 26, respectively.*

Proof. Validation follows from DBE. By Lemma 11, every eligible honest partial passes the masked-gadget checks, and its signature verifies. Aggregation preserves the verification identities and degree bounds. Its weighted openings are $b_i y_i, b_i r_i$ on D, with unrestricted values at x^*, while $B_T(x^*) = 1$ and $\deg B_T = M - |T| \le M - t$. The weighted relation preserves the shared selector. Lemma 19 and Theorem 1 therefore give correct decryption from at least t distinct eligible partials.

For IND-CPA security, reveal all revocation secrets and honest masking coins. Apart from BLS signatures, partials are then known CRS combinations. Since partials ignore the compiler header, Lemma 9's coefficient argument recovers an accepting proof independent of the fresh encryption coins. Hash values and issued signatures satisfy Appendix A.1's input bounds, which adaptive scalar disclosures preserve. Apply Lemmas 18 and 11.

Extraction, recipient support, and the registered-slot check 2 restrict the selector to eligible committee members. For an uncorrupted user without a challenge-context partial, $x_i H(\mathsf{enc}(\mathsf{ctx}^*))$ is absent from the

source-$\mathbb{G}_2$ inputs. Other contexts use distinct formal hash variables, and pairings output only in $\mathbb{G}_T$. Comparing coefficients in the signature equation forces $b_i = 0$. Its support lies in $\mathsf{Elig}^* \cap (U \cup J^*)$, but normalization and the degree bound require at least t^* nonzero coordinates, contradicting (7). The fresh-pad and collision argument is unchanged.

For hiding, Lemma 5 makes the public registration view independent of the pseudonyms. Since $Z_D(\tau) \neq 0$, the independent uniform scalars η, ζ, A, K, β make the five masked base encodings independent and uniform for each fixed honest setup, public view, and successful partial. These encodings and the trapdoors determine all shifts. Checks 1 and 2 uniquely determine the quotients because pairing with $[Z_D(\tau)]_2$ is injective. The non-membership proof is therefore witness independent. The selector and BLS signature are independent of the revocation opening.

Conditioning, on distinct pseudonyms, swapping the challenge pseudonyms and adjusting their blindings preserves the registry and permitted disclosures. Fresh independent masks preserve partial distributions, including repeated responses and those preceding later corruptions, which do not reveal past mask coins. Frozen contexts and their original lists preserve success and failure responses after publication, so the conditional advantage is zero. A union bound makes pseudonym collisions negligible, so the hiding advantage is negligible. Finally compactness follows from the constant number of proof elements. □

At threshold one, private partials recover DBE's perfect hiding. Public partials retain the historical guarantee above. Registry identifiers add 2λ bits per output and context. Their collision probability over q registrations is at most $q(q-1)/2^{2\lambda+1}$. Coupling these independent public identifiers identically in the fixed-registry games leaves the hiding bound unchanged.

Efficiency. The masked gadget uses eighteen $\mathbb{G}_2$ elements, including three commitments, and two $\mathbb{G}_1$ quotients. Verification uses fourteen pairing equations. A partial replaces the selector and its two shifts by σ_i, giving sixteen $\mathbb{G}_2$ and two $\mathbb{G}_1$ elements plus its label. Replacing two selector checks by signature verification leaves thirteen equations. The registered-slot proof is computed during aggregation.

Live registry and prover data use $O(M^2)$ group elements including helper rows, C_0, and $O(mM)$ IIP powers, excluding archived submissions and snapshots. Registration discards each $O(M^2)$ raw INM grid after updating the aggregates with $O(M^3)$ group operations. After list and context preprocessing, encryption takes constantly many group operations. Given P_L's coefficients and helper rows, a partial's non-membership component takes $O(|L|+1)$ group operations. List-polynomial construction, context parsing and hashing, and authorization are separate costs. Combination is polynomial. Constant ciphertext and partial sizes exclude context and identity labels.

AI usage disclosure

All research ideas and results are due to the authors. AI tools were used to proofread and polish the writing. The authors take full responsibility for the content of the paper.

References

[AAH+25] Kyoichi Asano, Nuttapong Attrapadung, Keisuke Hara, Keitaro Hashimoto, and Yohei Watanabe. Key revocation in registered attribute-based encryption. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part III*, volume 15676 of *LNCS*, pages 102–133. Springer, Cham, May 2025. 4

[AI09] Nuttapong Attrapadung and Hideki Imai. Attribute-based encryption supporting direct/indirect revocation modes. In Matthew G. Parker, editor, *12th IMA International Conference on Cryptography and Coding*, volume 5921 of *LNCS*, pages 278–300. Springer, Berlin, Heidelberg, December 2009. 4

[ALO98] William Aiello, Sachin Lodha, and Rafail Ostrovsky. Fast digital identity revocation (extended abstract). In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 137–152. Springer, Berlin, Heidelberg, August 1998. 3, 4

[ATSM09] Man Ho Au, Patrick P. Tsang, Willy Susilo, and Yi Mu. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In Marc Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 295–308. Springer, Berlin, Heidelberg, April 2009. 5, 9

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005. 5, 11

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Cham, May 2019. 11

[BDTW01] Dan Boneh, Xuhua Ding, Gene Tsudik, and Chi-Ming Wong. A method for fast revocation of public key certificates and security capabilities. In Dan S. Wallach, editor, *USENIX Security 2001*. USENIX Association, August 2001. 4

[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. 3, 4

[BGK08] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008*, pages 417–426. ACM Press, October 2008. 4

[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. 3, 4

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004. 8

[BM24] Olivier Blazy and Sayantan Mukherjee. A public key identity-based revocation scheme: - fully attribute-hiding and function private. In Elisabeth Oswald, editor, *CT-RSA 2024*, volume 14643 of *LNCS*, pages 3–24. Springer, Cham, May 2024. 4

[BMN+21] Olivier Blazy, Sayantan Mukherjee, Huyen Nguyen, Duong Hieu Phan, and Damien Stehlé. An anonymous trace-and-revoke broadcast encryption scheme. In Joonsang Baek and Sushmita Ruj, editors, *ACISP 21*, volume 13083 of *LNCS*, pages 214–233. Springer, Cham, December 2021. 4, 8

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007. 3

[BW06] Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 211–220. ACM Press, October / November 2006. 4, 8

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Berlin, Heidelberg, August 2014. 3, 6, 7

[CHM+20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 738–768. Springer, Cham, May 2020. 12

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 769–793. Springer, Cham, May 2020. 12

[Des88] Yvo Desmedt. Society and group oriented cryptography: A new concept. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 120–127. Springer, Berlin, Heidelberg, August 1988. 3

[DF90] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 307–315. Springer, New York, August 1990. 3

[DT08] Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. Cryptology ePrint Archive, Report 2008/538, 2008. 5, 9

[FKdP23] Dario Fiore, Dimitris Kolonelos, and Paola de Perthuis. Cuckoo commitments: Registration-based encryption and key-value map commitments for large spaces. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 166–200. Springer, Singapore, December 2023. 4

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, Berlin, Heidelberg, August 1994. 3

[Gen03] Craig Gentry. Certificate-based encryption and the certificate revocation problem. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 272–293. Springer, Berlin, Heidelberg, May 2003. 4

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013. 5, 7, 13

[GHK+25] Sanjam Garg, Mohammad Hajiabadi, Dimitris Kolonelos, Abhiram Kothapalli, and Guru-Vamsi Policharla. A framework for witness encryption from linearly verifiable SNARKs and applications. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 504–539. Springer, Cham, August 2025. 2, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 20, 21, 26, 28, 30, 31, 33, 34, 35

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run Turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Berlin, Heidelberg, August 2013. 5, 7, 13

[GKPW24] Sanjam Garg, Dimitris Kolonelos, Guru-Vamsi Policharla, and Mingyuan Wang. Threshold encryption with silent setup. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 352–386. Springer, Cham, August 2024. 5

[GMS12] Juan Manuel González Nieto, Mark Manulis, and Dongdong Sun. Fully private revocable predicate encryption. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP 12*, volume 7372 of *LNCS*, pages 350–363. Springer, Berlin, Heidelberg, July 2012. 4

[HLWW23] Susan Hohenberger, George Lu, Brent Waters, and David J. Wu. Registered attribute-based encryption. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 511–542. Springer, Cham, April 2023. 6, 7

[KMW23] Dimitris Kolonelos, Giulio Malavolta, and Hoeteck Wee. Distributed broadcast encryption from bilinear groups. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 407–441. Springer, Singapore, December 2023. 4

[KS12] Aggelos Kiayias and Katerina Samari. Lower bounds for private broadcast encryption. In *International workshop on information hiding*, pages 176–190. Springer, 2012. 4

[Lee25] Kwangsu Lee. Anonymous revocable identity-based encryption supporting anonymous revocation. *Mathematics*, 13(17):2854, 2025. 4

[LSW10] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *2010 IEEE Symposium on Security and Privacy*, pages 273–285. IEEE Computer Society Press, May 2010. 4

[Luo24] Ji Luo. Ad hoc broadcast, trace, and revoke: Plus time-space trade-offs for attribute-based encryption. *CiC*, 1(2):15, 2024. 4

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography. In Nigel P. Smart, editor, *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005. 11

[Mic96] Silvio Micali. Efficient certificate revocation. Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, Laboratory for Computer Science, March 1996. 22 March 1996. 3, 4

[ML19] Xuecheng Ma and Dongdai Lin. A generic construction of revocable identity-based encryption. Cryptology ePrint Archive, Report 2019/299, 2019. 4

[NN98] Kobbi Nissim and Moni Naor. Certificate revocation and certificate update. In Aviel D. Rubin, editor, *USENIX Security 98*. USENIX Association, January 1998. 3, 4

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, Berlin, Heidelberg, August 2001. 4, 8

[NP01] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *FC 2000*, volume 1962 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, February 2001. 4, 8

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991. 9, 17

[Pol26] Guru Vamsi Policharla. *Practical Witness Encryption Schemes and Applications*. PhD thesis, EECS Department, University of California, Berkeley, August 2026. 11, 28, 47, 48

[PTMW06] Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure attribute-based systems. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 99–112. ACM Press, October / November 2006. 4

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, February 1978. 3, 7

[RZ21] Carla Ràfols and Arantxa Zapico. An algebraic framework for universal and updatable SNARKs. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 774–804, Virtual Event, August 2021. Springer, Cham. 11

[SE13] Jae Hong Seo and Keita Emura. Revocable identity-based encryption revisited: Security model and construction. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 216–234. Springer, Berlin, Heidelberg, February / March 2013. 4

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997. 11

[SSW12] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 199–217. Springer, Berlin, Heidelberg, August 2012. 4

[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. 3

[WQZD10] Qianhong Wu, Bo Qin, Lei Zhang, and Josep Domingo-Ferrer. Ad hoc broadcast encryption (poster presentation). In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 741–743. ACM Press, October 2010. 3, 6, 7

A Framework under Our Extended CRS

A.1 Setup and Public Inputs

Work in the generic group model of Section 3.2 with honest setup and auxiliary generation. Fix $D = \{d_1, \ldots, d_n\} \subseteq \mathbb{F}_p$, $N \geq n \geq 1$, and $x^* \in \mathbb{F}_p \setminus D$. Write $L_i(X) = L_i^D(X)$ and $Z_D(X) = \prod_i (X - d_i)$. Setup samples independent τ, θ, α, with $\alpha \neq 0$ and the construction's specified exclusions, and publishes

$$\mathsf{crs} = \left([\tau^a \theta^b]_1, [\alpha \tau^a \theta^b]_1, [\tau^a \theta^b]_2\right)_{0 \leq a,b \leq N}.$$

There are no α-scaled encodings in $\mathbb{G}_2$. Formal variables X, Y, Z represent τ, θ, α.

The full public input space. Use common formal symbols $\boldsymbol{\zeta}$ for additional secrets and source/hash inputs, independent of X, Y, Z. This preserves correlations such as $s_i, H_j, s_i H_j$. Let $\mathcal{H} \subseteq \mathbb{F}_p[\boldsymbol{\zeta}]$ be the finite span of the nonconstant monomials in the permitted $\mathbb{G}_2$ inputs. They need not be algebraically independent. Known constants belong to the CRS grid, and no extra term is supplied multiplied by positive powers of X, Y, Z. Thus every source-$\mathbb{G}_2$ expression has the form

$$f(X, Y) + e(\boldsymbol{\zeta}), \qquad e(\boldsymbol{\zeta}) \in \mathcal{H}, \qquad \deg_X f(X, Y), \deg_Y f(X, Y) \leq N, \tag{11}$$

with coefficients recorded in the transcript and the grid part tracked from the public inputs' symbolic descriptions.

Every source-$\mathbb{G}_1$ expression lies in

$$\mathcal{A}[X]_{\leq N}, \tag{12}$$

where $\mathcal{A} = \mathbb{F}_p[Y, Z, \boldsymbol{\zeta}]$ is independent of X. Inverses of nonconstant polynomials in X are excluded. IIP index exponents lie in $\mathcal{A}$. Their secret coefficients are used only in the analysis, not recovered by extraction.

These bounds cover the full auxiliary arrays. IIP supplies $[s_i\tau^e]_1$ for $0 \le e \le N$. At capacity $n \le M \le N$, indexed non-membership supplies $[p_i(\theta,\alpha)\tau^a\theta^b]_1$ for $0 \le a, b < M$, where $p_i(Y,Z) = Y - y_i + r_iZ$. Public consistency checks fix these encodings, whose X-degree is at most N. The inequality $b+1 \le M \le N$ puts the required monomials in the CRS. Preprocessing and threshold masks use $a, b < n$.

The helper polynomials Q_i of Construction 3 have X-degree at most $n-2$ for $n \ge 2$ and vanish for $n = 1$. Their coefficients are linear in the p_j, so the retained arrays encode $[\theta^c Q_i(\tau,\theta,\alpha)]_1$ for $c < n$. Source-$\mathbb{G}_2$ helpers and shifts are public grid combinations within the degree bounds. The normalization indices $[1]_1, [L_i(x^*)]_1$ and their auxiliary powers are also CRS combinations. Thus preprocessing preserves both input spaces.

Generic source operations preserve these spaces, and pairings produce only target elements. Use the common symbolic execution of Section 3.2, with recorded adaptive scalar coefficients and polynomially bounded degrees. Except for negligible accidental equalities or substitution events, verification equations are formal identities. On each permitted scalar disclosure, substitute its value and absorb constants into the grid. The bounds must still hold, the setup trapdoors remain secret, and additional source inputs require rechecking the full spaces. Collision bounds account for the setup-domain exclusions.

These hypotheses concern the complete extraction computation, not setup alone. In each application, coefficient comparison first recovers a proof from source combinations independent of the fresh encryption coins. The application checks their input bounds, including all further public responses, before applying extraction. In threshold confidentiality, revealing the revocation secrets and honest partial-generation masks makes partials' polynomial coefficients known. That information is not assumed in the historical-hiding experiment.

A.2 Gadget Extraction

The results in this subsection use the full input conditions of Section A.1.

Lemma 12 (Degree shifts under the extended CRS)**.** *Suppose the full source inputs satisfy* (11)–(12)*. For $0 \le d_X, d_Y \le N$ with at least one strict inequality, let $\overline{F} \in \mathbb{G}_2$ pass the applicable $\mathbb{G}_2$ degree checks below.*

1. *If $d_X < N$, $[\tau^{N-d_X}]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_X$.*
2. *If $d_Y < N$, $[\theta^{N-d_Y}]_1 \circ \overline{F} = [1]_1 \circ \overline{F}_Y$.*

Then its transcript yields a polynomial $F(X,Y) \in \mathbb{F}_p[X,Y]$ with known coefficients, $\deg_X F(X,Y) \le d_X$, $\deg_Y F(X,Y) \le d_Y$, and $\overline{F} = [F(\tau,\theta)]_2$. For $\overline{Q}, \overline{Q}' \in \mathbb{G}_1$ and $1 \le h \le N$, use the following $\mathbb{G}_1$ degree check.

1. *$\overline{Q} \circ [\tau^h]_2 = \overline{Q}' \circ [1]_2$.*

Passing this check forces the formal exponent of $\overline{Q}$ to have X-degree at most $N-h$ in $\mathcal{A}[X]$.

Proof. Write the exponent as $f(X,Y) + e(\boldsymbol{\zeta})$. A positive X-shift produces shifted $\boldsymbol{\zeta}$ monomials absent from (11), forcing $e = 0$. A positive Y-shift does the same. Monomial multiplication has no cancellation, so the shifted grid bounds give the stated degrees. Unchecked variables already have bound N, and the transcript gives the coefficients of f. For $\mathbb{G}_1$, the check gives $X^hQ = Q'$ over the integral domain $\mathcal{A}$. Since $\deg_X Q' \le N$, either $Q = 0$ or $\deg_X Q \le N - h$. □

Scalar commitments. The usual selector commitment $\overline{B} = [\sum_i b_i L_i(\tau)]_2$ uses the CRS's Y-degree-zero slice. Enforce its opening type with

1. $[\tau^{N-n+1}]_1 \circ \overline{B} = [1]_1 \circ \overline{B}_X$.
2. $[\theta^N]_1 \circ \overline{B} = [1]_1 \circ \overline{B}_Y$.

Both shifts are positive. Lemma 12 therefore extracts a known $B(X) \in \mathbb{F}_p[X]_{<n}$ and its opening $b_i = B(d_i)$. Distinct openings would give distinct bounded-degree interpolants agreeing at τ, an excluded collision.

Lemma 13 (Maximum-degree gadget)**.** *The degree-shift checks give a succinct linearly verifiable argument of knowledge for length-n scalar vectors whose interpolants have degree at most d, where $0 \le d < n$, committed in $\mathbb{G}_2$, with polynomial-time straight-line extraction.*

Proof. Apply Lemma 12 with $(d_X, d_Y) = (d, 0)$ and return $(F(d_i))_i$. Interpolation recovers F since $d < n$. Honest witnesses give both shifts within the CRS. □

Lemma 14 (Signature gadget)**.** *The signature gadget with identity-committed (pk, σ) is a succinct linearly verifiable argument of knowledge with polynomial-time straight-line extraction.*

Proof. Verify:

1. $\mathsf{pk} \circ \mathsf{tg} = [1]_1 \circ \sigma$.

Return the two identity-committed group elements (pk, σ). The single bilinear check is the relation itself, proving completeness and extraction. □

A.2.1 Zero Check

Use [Pol26, Construction 3.4 and Lemma 3.3], swapping source groups and adding the Y-degree check. For $S \subseteq [\![n]\!]$, put $Z_S(X) = \prod_{i \in S}(X - d_i)$. The prover supplies $\overline{Q} = [Q(\tau)]_2$ and both selector shifts. Verify:

1. Apply selector Check 1.
2. Apply selector Check 2.
3. $[1]_1 \circ \overline{B} = [Z_S(\tau)]_1 \circ \overline{Q}$.

Lemma 15 (Zero-check gadget)**.** *This construction is a succinct linearly verifiable argument of knowledge that the committed vector vanishes on S, with polynomial-time straight-line extraction.*

Proof. Extract $B(X) \in \mathbb{F}_p[X]_{<n}$ from the selector checks. Writing the exponent of $\overline{Q}$ as $q(X, Y) + e(\boldsymbol{\zeta})$, the quotient check gives

$$B(X) = Z_S(X)\big(q(X, Y) + e(\boldsymbol{\zeta})\big).$$

Since $Z_S \neq 0$, coefficient comparison forces $e = 0$ and excludes positive Y-degree. Thus $B = Z_S Q$ for $Q(X) = q(X, 0)$, and $b_i = B(d_i)$ opens $\overline{B}$ and vanishes on S, including $S = \emptyset$ or $S = [\![n]\!]$. Conversely, a valid witness gives $Q = B/Z_S$ of degree at most $n - 1 - |S|$, or $Q = 0$ when $B = 0$. The CRS supplies all encodings. □

A.2.2 Indexed Inner Product

We use [Pol26, Construction 3.7 and Lemma 3.6], including its selector-degree check and individual Lagrange weights. Write the index entries as $u_i = [a_i]_1$, with $a_i \in \mathcal{A}$, and define

$$C(X) = \sum_i L_i(x^*)^{-1} a_i L_i(X), \qquad \overline{C} = [C(\tau)]_1.$$

Every $L_i(x^*) \neq 0$. For $n \ge 2$, commit to (b, v) using $\overline{B} = [B(\tau)]_2$ and the identity commitment to $v \in \mathbb{G}_1$. The proof supplies $\overline{Q}_X, \overline{Q}_Z, \widehat{Q}_X, \widehat{v} \in \mathbb{G}_1$ and both selector shifts. Verify:

1. Apply selector Check 1.
2. Apply selector Check 2.
3. $\overline{C} \circ \overline{B} = v \circ [1]_2 + \overline{Q}_X \circ [\tau - x^*]_2 + \overline{Q}_Z \circ [Z_D(\tau)]_2$.
4. $\overline{Q}_X \circ [\tau^{N-n+2}]_2 = \widehat{Q}_X \circ [1]_2$.

5. $v \circ [\tau^N]_2 = \widehat{v} \circ [1]_2$.

Lemma 16 (Indexed inner-product gadget)**.** *Under Section A.1's hypotheses, this construction, including the $n = 1$ case below, is a succinct linearly verifiable argument of knowledge for $v = \sum_{i=1}^{n} b_i u_i$. The vector $b \in \mathbb{F}_p^n$ uses the structured commitment $\overline{B}$, and $v \in \mathbb{G}_1$ uses the identity commitment. Extraction is polynomial-time and straight-line.*

Proof. For $n \geq 2$, the shift checks give a known $B(X) \in \mathbb{F}_p[X]_{<n}$, a quotient exponent $Q_X \in \mathcal{A}[X]_{\leq n-2}$, and an output exponent $a_v \in \mathcal{A}$. Check 3 becomes

$$C(X)B(X) = a_v + (X - x^*)Q_X(X) + Z_D(X)Q_Z(X).$$

Apply the $\mathcal{A}$-linear map $\Lambda(F) = \sum_i L_i(x^*)F(d_i)$. The left side becomes $\sum_i a_i B(d_i)$ and the $Z_D Q_Z$ term vanishes. Since $(X - x^*)Q_X$ has degree at most $n - 1$, its image is its value at x^*, also zero. As $\Lambda(a_v) = a_v$, returning (b, v) with $b_i = B(d_i)$ gives $v = \sum_i b_i u_i$. This permits correlated, Y, Z-dependent index coefficients without recovering their discrete logarithms.

For completeness, interpolation gives $Q_X(X), Q_Z(X)$ of degree at most $n - 2$, with coefficients linear in the a_i and known witness-dependent scalars. The IIP arrays supply their encodings, the shift of $Q_X(X)$, and $[\tau^N a_v]_1$. The CRS supplies both shifts of $B(X)$. For $n = 1$, the following verifier enforces bounds $(0, 0)$ on $\overline{B}$.

1. Apply selector Check 1 with $n = 1$.
2. Apply selector Check 2 with $n = 1$.
3. $u_1 \circ \overline{B} = v \circ [1]_2$.

Extract the constant $b_1 = B(d_1)$, with $v = b_1 u_1$. No quotient or negative degree bound is needed. □

A.3 Normalization and Composition

DBE requires $\sum_i b_i = 1$, and threshold encryption requires $B(x^*) = 1$. Both are $\sum_i a_i b_i = 1$, with $a_i = 1$ and $a_i = L_i(x^*)$, respectively. These weights are nonzero since $x^* \notin D$.

Lemma 17 (NonZero through normalization)**.** *Fix a public vector $a \in \mathbb{F}_p^n \setminus \{0\}$ and define the normalized relation*

$$\mathsf{NonZero}_a = \left\{ (\bot, \bot, \bot, b) : b \in \mathbb{F}_p^n,\ \sum_i a_i b_i = 1 \right\}.$$

Under Section A.1's hypotheses, its structured-committed version has a succinct linearly verifiable argument of knowledge with straight-line extraction. Every extracted selector is nonzero.

Proof. Apply Lemma 16 with index $([a_i]_1)_i$, output $[1]_1$, and selector bounds $(n - 1, 0)$. The auxiliary powers are CRS combinations. Extraction gives $[\sum_i a_i b_i]_1 = [1]_1$, hence $\sum_i a_i b_i = 1$ and $b \neq 0$. Completeness and efficiency follow from IIP, including its $n = 1$ case. □

Normalization [Pol26, p. 54, footnote 7] restricts nonzeroness. For example, $(1, -1)$ is nonzero but has sum zero. Rescaling works only when $a \cdot b \neq 0$, all other routed constraints are homogeneous, and all corresponding witness coordinates are rescaled together. Both applications choose normalized selectors directly in their constructions.

Lemma 18 (Framework gadgets under the extended CRS)**.** *Under Section A.1's hypotheses, the five gadgets above are succinct linearly verifiable arguments of knowledge with polynomial-time straight-line extractors.*

Proof. Apply Lemmas 14, 13, 15, 16, and 17. □

Lemma 19 (Gadget composition)**.** *Consider succinct linearly verifiable arguments complete and knowledge sound under the full input spaces above. Suppose shared parameter, index, and witness coordinates use the same setup, auxiliary, and commitment algorithms, including commitment-key generation. Shared setup coordinates have the same trapdoor spaces, and each gadget's space is their routed product. Then Theorem 1 applies to the routed relation.*

Proof. Include each coordinate's algorithms once in the common tuples $\mathtt{Gen}', \mathtt{AuxGen}', \mathsf{CGen}', \mathsf{Commit}'$. Restricting these tuples gives each gadget's algorithms, with $\mathcal{T}_i = \prod_{j \in P_i} \mathcal{T}'_j$, as required by Definition 10. Generate shared setup and auxiliary data once from common inputs, and commit each shared witness coordinate once under its common key.

Binding makes extracted scalar openings agree, and identity-committed group coordinates agree directly. The component witnesses therefore satisfy the routed product. The compiler theorem now applies. □

A.4 IIP AuxGen Inversion Hardness

Lemma 20 (IIP auxiliary generation under the extended public view)**.** *For the extended CRS of Section A.1, the algorithm*

$$\mathsf{IIP}.\mathtt{AuxGen}((\mathsf{pp}, \mathsf{crs}), x) = \big([x]_1, ([x\tau^e]_1)_{1 \le e \le N}\big)$$

has hard-to-invert auxiliary generation under Definition 15 against generic bilinear-group adversaries, including Construction 7's full public view. This view includes adaptive Revoke and Corrupt queries after the challenge, with Corrupt answered by the inversion oracle. The claim also holds for independent uniform inputs in $\mathbb{F}_p^\times$.

Proof. Reveal every independently sampled revocation pair (y_i, r_i) at creation. The non-membership inputs are then known CRS combinations by Section A.1. For each Auxgen query introduce an independent symbol S_i and supply $([S_i X^e]_1)_{0 \le e \le N}$. Public generic operations produce all registration data, including unit-selector IIP proofs and non-membership helpers. Satisfied-literal quotients $L_i(X)/Z_{J(a)}(X)$ have public coefficients and require no unknown scalar. The x_i^{-1}-scaled proofs remain private, and ABE publishes no signatures.

Assign handles to formal exponents and test formal equality, allowing scalar coefficients to depend on the preceding transcript. At the first inversion query for i, sample and return uniform s_i, then substitute $S_i = s_i$ throughout. Repeat queries return s_i, while all other S_j remain formal. Public encryption uses fresh independent coins and consistent random-oracle answers on formal handles. Process all later public computations and Revoke/Corrupt replies this way, covering the complete transcript without revealing further IIP scalars.

Source degrees before encryption are at most $2N + 1$ in $\mathbb{G}_1$ and $2N$ in $\mathbb{G}_2$. Compiler headers are fresh scalar combinations of verification coefficients, so formal encryption coins raise each bound by at most one. Linear source operations and source-only pairings bound target degrees by $4N + 3$. The real and symbolic experiments couple unless distinct expressions evaluate equally or become identical after an adaptive substitution. Each event involves a nonzero bounded-degree polynomial in independent remaining scalars. The collision bound, unioned over polynomially many operations and substitutions, is negligible. Setup exclusions replace the field size by the permitted domain size.

In the symbolic experiment, the transcript is independent of unrevealed scalars. For output (j, s^*) with j never inverted, the unsampled S_j is uniform independently of (j, s^*). Inversion requires $[s^*]_1 = [S_j]_1$, hence $s^* = S_j$, with probability $1/|\mathbb{F}_p|$. Add the negligible coupling error. For $\mathbb{F}_p^\times$ inputs, sample disclosures there and use $1/(|\mathbb{F}_p| - 1)$. This proves scalar-recovery hardness without witness extraction. □