Sovereign Modal Signatures

Yingfei Yan[1], Khai Hanh Tang[2](✉), Hien Chu[3,4](✉), Sherman S. M. Chow[5], San Ling[2], Huaxiong Wang[2], and Kai Zhang[6]

[1] Université Clermont Auvergne, CNRS, LIMOS, France
[2] Nanyang Technological University, 50 Nanyang Ave, Singapore
[3] TU Wien, Vienna, Austria
[4] Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
[5] Dept. of Information Engineering, Chinese University of Hong Kong, Hong Kong
[6] School of Mathematics and Statistics, Shaanxi Normal University, Xi'an, China
khaihanh.tang@ntu.edu.sg, hien.chu@tuwien.ac.at

Abstract. Signing anonymously under both hidden and public policies with per-signature accountability has only been studied piecemeal, as prior schemes address at most one dimension or restrict function classes. Here, we introduce sovereign modal signatures, the first scheme unifying both dimensions with support for hidden-circuit evaluation over bounded-size arithmetic circuits without universal circuits. Each signer holds a private attribute and a hidden policy F, certified by an authority and concealed from signatures. Resting on a public combiner P, the signing process maps the joint outputs of F and an ad hoc function G to a derived message and an opening tag governing per-signature traceability.

Modifying the effective policy at signing time requires no key reissuance, since G is chosen freshly against the fixed F. Achieving zero-knowledge evaluation of F without exposing its structure is our core challenge, resolved by treating F's PLONK description as witness rather than statement. No universal circuits are required: our reduction yields a fixed verification circuit of size linear in F's gate count; the resulting scheme is provably secure in the random oracle model.

Concretely, our generic construction needs only signatures, public-key encryption, and non-interactive zero-knowledge arguments of knowledge for arithmetic circuits. Hidden arithmetic circuits over any finite field are fully supported, subsuming the restricted Boolean and linear function classes of prior hidden-function schemes. Our post-quantum instantiation combines ZKBoo (Usenix Security '16) under the MPC-in-the-Head (MPCitH, STOC '07) paradigm with lattice-based primitives over $\mathbb{Z}_q$, achieving plausibly post-quantum security without pairings or the algebraic group model. With proof sizes quasilinear in F's gate count and linear in G and P's multiplications, the scheme also readily admits richer MPCitH instantiations.

Keywords: Accountable Anonymity · Functional Signatures · Lattices

1 Introduction

Privacy-preserving signatures serve a fundamental purpose: a signer proves she satisfies a governing policy and belongs to an authorized group, without disclosing her identity. The design space is determined by two independent axes of control. The first axis is *signability* (signing eligibility control): what messages a signer may sign, and whether the governing policy is public knowledge or a hidden credential. The second axis is *accountability*: when, and how completely, a designated authority may recover the signer's identity from a signature. Three decades of work have advanced each axis in isolation, leaving a fragmented landscape in which no single scheme addresses both.

Signability. Ring signatures (RS) [9,61] and group signatures (GS) [19,45] impose no constraint: any enrolled member may sign any message freely. Attribute-based signatures (ABS) [57] and multimodal private signatures (MPS) [58] allow signing only when the message and the signer's attribute together satisfy a function chosen publicly at signing time. Functional signatures (FS) [12] and policy-based signatures (PBS) [8] go further: each signing key encodes a fixed *hidden* function, and a signature is valid only when this function approves the message. The hidden function cannot be updated without reissuing the key, and no existing construction supports it beyond restricted circuit classes, namely, linear systems over $\mathbb{Z}_2$ [20,68], quadratic Boolean functions [65], or functions left abstract [12,37]. This stands in sharp contrast to public-function schemes, which already support all bounded-size arithmetic circuits [52,63].

Accountability. Group signatures [19] expose the full signer identity to a tracing authority upon misuse; ring signatures [61] hide the signer unconditionally among n members. Bifurcated anonymous signatures (BiAS) [49] introduced per-signature accountability control, letting the signer choose at each signing between full traceability and unconditional anonymity. MPS [58] refines the traceable branch into distinct modes, such as role-level or group-membership linkability. A spectrum of intermediate mechanisms has been proposed along this axis, from linkable and accountable ring signatures to certified limited opening [23,25,26,48,67,71]; we survey these in Section 1.2. Throughout, accountability predicates are publicly known and chosen from a fixed set at signing time. Hidden policies and flexible accountability have not been jointly designed.

The intersection gap. Existing notions advance one line at a time; even MPS [58], which touches both, fixes a finite set of disclosure modes and restricts hidden policies to narrow function classes. No known scheme simultaneously enforces a hidden arithmetic signing policy, admits an ad hoc public policy overlaid at signing time, and grants per-signature control over how much identity the authority recovers. This gap is a deployment barrier. Consider corporate compliance. An auditor signs reports under a proprietary risk-scoring function F, hidden to protect business logic, composed with a publicly auditable eligibility check G; routine reports require only role-level accountability, while reports flagging systemic risk require full identity recovery. Specifying F, G, and the accountability rule as a single coherent system is beyond any existing primitive.

Expressiveness. Unifying the two axes requires proving in zero knowledge the correct evaluation of a hidden function F over a bounded-size arithmetic circuit, without exposing F's structure or computation trace. Universal circuits [66] offer a general approach: to evaluate a circuit $\mathsf{C}(\mathbf{x})$ with input $\mathbf{x}$, one proves the evaluation of a fixed universal program $\mathcal{U}$ input the code of C and $\mathbf{x}$. Despite recent improvements [47,54,55,72], current universal circuits remain too large for practical use in privacy-preserving signatures. The best known construction [55] achieves quasilinear size using Boolean gates, still larger than a direct arithmetic-circuit proof by a $\log n_{\mathsf{gate}}$ factor of the number of gates n_{gate}.

Converting arithmetic operations to Boolean circuits is also unsatisfying. An addition over the finite field $\mathbb{F}_q$ requires $2 \log q$ bit multiplications and a single multiplication costs $\log^2 q$ bit multiplications under schoolbook arithmetic, with faster algorithms such as Karatsuba [41] and FFT reducing but not closing this gap. Arguments for bounded-size Boolean circuits exist [59] but do not extend to $\mathbb{Z}_q$ arithmetic at acceptable cost.

Prior schemes thus restrict hidden functions to narrow classes, bounded public families [58], or leave the function class open [12]. A recent MPS construction [65] supports quadratic Boolean functions and could extend to all Boolean circuits via techniques from [59], but arithmetic circuits over $\mathbb{Z}_q$ remain out of reach. This state of affairs motivates three research questions (RQs).

RQ 1. *Can a privacy-preserving signature scheme prove the correct evaluation of an arbitrary hidden bounded-size arithmetic circuit over $\mathbb{F}_q$, without universal circuits and the algebraic group model?*

RQ 2. *Is there a single signature notion that unifies hidden and public policy enforcement with flexible per-signature signability and accountability, subsuming FS, PBS, ABS, BiAS, and MPS as special cases?*

RQ 3. *Can the notion from **RQ 2** be instantiated generically from standard primitives and concretely from post-quantum lattice-based assumptions?*

1.1 Our Contributions

We answer RQ 1, RQ 2, and RQ 3 affirmatively. We first establish that hidden arithmetic-circuit evaluation is achievable (RQ 1), then define the notion that exploits this capability (RQ 2), and finally construct and instantiate it (RQ 3).

Efficient hidden arithmetic-circuit evaluation (RQ 1). The main technical challenge is proving in zero knowledge the evaluation of the function $P(F(M, \mathsf{attr}, w), G(M, \mathsf{attr}, w))$, concealing both the function F and its computation trace. Evaluating a hidden circuit requires concealing not only the inputs but also the circuit structure and all intermediate computation traces, making standard non-interactive zero-knowledge arguments of knowledge (NIZKAoK) techniques for public circuits not directly applicable. We resolve this by adapting PLONK's arithmetization [34]: PLONK encodes each arithmetic gate as a quadratic constraint via selector values and enforces wire consistency across gates via a permutation. Crucially, this encoding, which we call the *PLONK description* pd of C and which we refer to via PLONK's arithmetization throughout, is

a vector of length linear in the total gate count of C. Treating this description as part of the *witness* rather than the statement reduces hidden-circuit satisfiability to verifying a fixed public circuit $\mathcal{U}^{\mathsf{plonk}}$ of the same linear size.

Concretely, to prove F's evaluation without revealing its structure, we fix F's PLONK description as part of the witness, derive random challenges α, β from the random oracle, and view the resulting constraint system as a fixed public verification circuit $\mathcal{U}_F^{\mathsf{plonk}}$ whose size is linear in the gate count of F. A NIZKAoK on $\mathcal{U}_F^{\mathsf{plonk}}$ then certifies F's evaluation in zero knowledge, as efficiently as if F's circuit structure were public. This is the first result of this kind for bounded-size arithmetic circuits over $\mathbb{F}_q$ without universal circuits [66] and without the algebraic group model [32], and consequently yields the first efficient privacy-preserving signature scheme supporting hidden arithmetic-circuit evaluation without universal circuits. See Section 4 for a detailed discussion.

Prior attempts either require non-standard assumptions [11,22] or restrict to narrow circuit classes [51,65]; our technical contribution here is this application of a PLONKish-based reduction, not general uses of zero-knowledge proofs.

A new notion (RQ 2). We propose *Sovereign Modal Signatures* (SMS), a notion in which both axes are governed by a single unified mechanism. "*Sovereign*" names the per-signature autonomy of the signer, who governs the effective policy and accountability mode at each signing, unconstrained by a predefined menu of options. "*Modal*" names the mode-structured accountability output, generalizing the finite multimodal disclosure of MPS [58] to an open-ended spectrum governed by the signer's choice of G and the certified structure of F. "Sovereign modal" thus names a compound property: the signer exercises *sovereign* per-signature control over a *mode*-structured policy and accountability system.

In SMS, each signer holds a private attribute attr and a hidden policy F certified by a function-issuing authority FA. For each message M, the signer additionally chooses an *ad hoc* function $G \in \mathcal{G}$, freely selected at each signing without reissuing the certificate. G is *public*: any verifier can confirm that G approved the message, providing the publicly auditable policy component absent from FS and PBS. Hiding G would forfeit this auditability and double the proof cost; the flexibility of SMS lies in the per-signature choice of G, not its concealment. In practice, G's inputs or outputs may themselves carry private information, *e.g.*, as ciphertexts or commitments, even when G's structure is public. G does not override F's accountability encoding; a public combiner P composes their outputs so that both the signing policy and the accountability mode are jointly determined:

$$(\overline{M}, \mathsf{op}) = P\big(F(M, \mathsf{attr}, w),\, G(M, \mathsf{attr}, w)\big). \tag{1}$$

The output $\overline{M}$ determines what is signed, and op determines what the authority recovers. A special value $\mathsf{op} \notin \mathcal{OP}_{\mathsf{sgnbl}}$ blocks signing entirely; any $\mathsf{op} \in \mathcal{OP}_{\mathsf{sgnbl}}$ permits a signature and encodes the accountability mode, ranging from unconditional anonymity through partial disclosure to full identity recovery. The signer produces a signature on $\overline{M}$ in the manner of a functional signature [12], binding the output to the certified hidden policy, with op then encrypted for the tracing

Table 1. Comparison of SMS with prior privacy-preserving signature schemes

Scheme	Public Policy		Hidden Policy		Accountability
	Scoped	Class	Scoped	Class	
RS [9,61]	—	—	—	—	—
GS [19,45]	—	—	—	—	Identity (ID)
PBS [8]	—	—	✓	Linear [20,68]	—
FS [12]	—	—	✓	NP	—
ABS [57]	✓	Arithmetic [63]	—	—	—
BiAS [49]	—	—	—	—	ID / None [49]
MPS [58]	✓	Arithmetic [58,65]	—	—	Finite Modal
SMS	✓	Arithmetic	✓	Arithmetic	Sovereign Modal

Scoped: whether this policy class determines signing eligibility.
Arithmetic: instantiations exist for all bounded-size arithmetic circuits over $\mathbb{F}_q$.
Linear: instantiations exist for linear functions over $\mathbb{Z}_2$.
Finite Modal: selection from a predefined, bounded set of disclosure modes.
Sovereign Modal: signer-governed control over an open-ended spectrum.

authority TA [58]. Tracing is triggered by TA running Open on any valid signature, recovering exactly the information encoded in op. Throughout, attr and F remain hidden from all signatures.

Crucially, even with F hidden, any verifier can confirm that the signature was produced under a policy certified by FA and a publicly known G, with P yielding a signable output, a form of policy compliance auditability absent from FS and PBS. The effective policy can be updated at each signing via G without reissuing the key or the certified hidden F. Existing primitives arise as special cases: PBS and FS correspond to P acting as the identity on F's output; ABS and BiAS correspond to P acting as the identity on G's output; MPS corresponds to G selecting an index into a public function family fed to P. SMS thus subsumes RS, GS, ABS, BiAS, MPS, FS, and PBS as special cases, and realizes the single-signer, function-bound slice of the functional credentials framework envisioned [24].

Table 1 situates SMS among prior privacy-preserving signatures; SMS is the unique one with public arithmetic functions, hidden arithmetic functions, sovereign per-signature policy control, and modal accountability simultaneously.

Generic construction and post-quantum instantiation (RQ 3). Based on the paradigm of Nguyen *et al.* [58], we build SMS generically from four standard primitives: a secure signature scheme $\mathcal{S}$, a chosen-plaintext-secure (CPA-secure) public-key encryption scheme $\mathcal{E}$, a strongly unforgeable one-time signature scheme $\mathcal{OTS}$, and a NIZKAoK for arithmetic circuits satisfying simulation-sound extractability [18], where a valid proof always yields an extractable witness even in the presence of simulated proofs. The hidden-circuit evaluation of NIZKAoK is handled by the PLONK-based reduction established in RQ 1: fixing F's PLONK description as part of the witness reduces the proof obligation to a fixed public verification circuit, enabling the NIZKAoK to verify F's evaluation

as efficiently as that of a public circuit of the same size. The NIZKAoK proves in zero knowledge that the signer holds a valid FA-issued certificate over (attr, F), that $P(F(\cdot,\cdot,\cdot), G(\cdot,\cdot,\cdot))$ evaluates correctly to a signable opening tag, and that the tag is honestly encrypted for TA and bound to a one-time verification key certified under the signer's attribute key. This avoids universal circuits [66], pairing-based assumptions, and idealized algebraic models.

Regarding our post-quantum instantiation, we employ ZKBoo [35] under the MPC-in-the-Head (MPCitH) paradigm [39]. ZKBoo's direct handling of arithmetic-circuit satisfiability makes it a natural fit for the NIZKAoK slot in the generic SMS construction, and this instantiation replaces the abstract NIZKAoK with ZKBoo without redesigning the signing protocol. The lattice-based primitives over $\mathbb{Z}_q$ include Lindner–Peikert (LP) public-key encryption (PKE) [50] ($\mathcal{E}_{\mathsf{lp}}$), Jeudy–Roux-Langlois–Sanders (JRS) signatures [40] ($\mathcal{S}_{\mathsf{jrs}}$), and Kawachi–Tanaka–Xagawa (KTX) commitments [43]. Since ZKBoo satisfies simulation-sound extractability with quasi-unique response [29], *i.e.*, any commitment and challenge admit at most one valid response up to negligible probability, it subsumes the anti-malleability role of $\mathcal{OTS}$ in the generic construction.

The PLONK's arithmetization induces a two-phase witness structure. Witnesses available before the random challenges α, β are committed first. Witnesses determined by α, β are committed in a second phase. This requires a targeted modification to ZKBoo's decomposition, detailed in Section 5. The resulting signature size is quasilinear in the gate count of F and linear in the number of multiplications of G and P, with a $\log q$ overhead arising from the binary decomposition required to parse field elements into the binary representation compatible with the lattice-based primitives. This framework also supports richer instantiations via advanced MPCitH techniques [3,4,7,16,17,42,62,30] and vector oblivious linear evaluation in the head (VOLEitH) techniques [6,13,27,60].

Benchmarking with concrete parameter selection is deferred to future work. See Section 5 and the full version [69] for details of these building blocks.

1.2 Related Work

Signability in prior notions. Table 1 summarizes the landscape. Here, we discuss the work most closely related to SMS in design goals. We classify MPS [58] as limited function composition: the signer evaluates a public function to obtain an index j, then selects the j-th member of a public family $\{P_i\}_{i=1}^{k}$ as the effective hidden policy. The hidden function is never directly composed with a public one; the family size is bounded ($k = 4$ in [58]); and the selection circuit evades, rather than solves, the challenge of proving hidden-circuit satisfiability. SMS replaces selection with genuine composition via the combiner P, supports an unbounded family $\mathcal{F}$ of hidden arithmetic circuits, and directly confronts hidden-circuit evaluation. Furthermore, MPS's selection-based approach costs $\mathcal{O}(k \cdot (n_{\mathsf{mlt}} + n_{\mathsf{add}}))$ constraints; the PLONK-based reduction developed here reduces this to $\mathcal{O}(n_{\mathsf{mlt}} + n_{\mathsf{add}})$, removing the multiplicative factor k. The original PBS construction [8], built on Groth–Sahai proofs [36] over structure-preserving

signatures [2], is inherently restricted to functions expressible as bilinear maps. Subsequent lattice-based constructions [20,68] extend the reach only to linear systems over $\mathbb{Z}_2$. Bobolz *et al.* [10] introduced universal anonymous signatures to balance utility and privacy, but omit hidden signing functions and provide only a high-level framework without concrete instantiations. Nguyen *et al.* [59] defined bicameral and auditably private signatures, pairing a private signing policy with a public disclosure mechanism in a fixed two-state model; that binary structure is a special case of SMS, which supports arbitrary composition of hidden and public functions and a continuously variable accountability spectrum.

Accountability mechanisms in prior notions. The accountability axis has a richer history than the binary GS model suggests. Linkable ring signatures, proposed by Liu, Wei, and Wong [56], allow linking two signatures by the same signer without revealing identity. Accountable ring signatures, introduced by Xu and Yung [67] and first realized in the standard model by Lai, Zhang, Chow, and Schröder [48], allow the signer to designate an opener who can later reveal the signer's identity, thereby combining the ad hoc group flexibility of ring signatures with the accountability of group signatures. Chow, Susilo, and Yuen [25] proposed escrowed linkability, where only a designated linking authority can decide whether two ring signatures share a signer. Traceable ring signatures, proposed by Fujisaki and Suzuki [33], publicly reveal any signer who signs two distinct messages under the same tag, without any designated opener.

Kiayias, Tsiounis, and Yung [44] introduced traceable signatures, where a tracing trapdoor tags a misbehaving user's signatures without opening others; Chow [23] improved this by enabling per-user tagging without first collecting all signatures. Abe, Chow, Haralambiev, and Ohkubo [1] introduced double-trapdoor anonymous tags for a modular construction of traceable signatures supporting authorship claiming and denial: the signer herself can selectively prove or disprove her authorship of a specific signature, a form of user-controlled accountability orthogonal to authority-initiated tracing. Kiayias and Zhou [46] proposed hidden identity-based signatures, where the opening authority needs only its secret key to reveal the signer's identity; Chow, Zhang, and Zhang [26] gave the first standard-model construction. Zhang, Wu, and Chow [71] introduced certified limited opening, delegating opening rights so that neither the master certifier nor unauthorized parties can revoke anonymity. Each of these offers a distinct tracing granularity; SMS's opening tag op can encode any of these levels as a special configuration of the combiner P; user-controlled accountability such as authorship claiming and denial points to a natural extension of the framework.

Proofs for hidden function evaluations. Several recent works construct zero-knowledge proofs for the satisfiability of hidden circuits, but each either targets a different security model or a different problem class. Ling *et al.* [51] gave zero-knowledge proofs for committed symmetric Boolean functions, covering only a narrow subclass. Boneh, Nguyen, and Ozdemir [11] commit to a hidden circuit's PLONK's description via Marlin [21], yielding proofs of arithmetic-circuit satisfiability, but their construction relies on the algebraic group model (AGM) [32], which carries documented soundness concerns [70]. Di *et al.* [28] propose MUX-

Proofs, layering Marlin over a univariate sum-check to verify RAM steps encoded as rank-1 constraint systems; Choudhuri *et al.* [22] introduced Sublonk, applying PLONK's arithmetization to arbitrary-degree polynomials for instruction-level proofs. Both also rely on the AGM.

Our construction operates in the random oracle model (ROM) for both the generic and the lattice-based instantiation, avoiding these concerns entirely. Tang, Pham, and Ngo [64] apply PLONK's arithmetization to prove correct execution of RAM programs, a problem structurally different from that of hidden-circuit evaluation for signatures; the treatment developed here, specifically treating the PLONK's description as a witness component to reduce hidden-circuit satisfiability to a fixed public verification circuit, is tailored to the NIZKAoK requirements of privacy-preserving signatures and is not addressed in that prior work. In short, no prior scheme achieves ROM-based proofs for arbitrary hidden bounded-size arithmetic circuits in a privacy-preserving signature setting.

Relationship to anonymous credentials. Anonymous credentials [5,14,15] and SMS share structural ancestry: both involve a certifying authority, a hidden user attribute, and a zero-knowledge proof of possession. The two paradigms are nonetheless orthogonal. The former focuses on selective disclosure of attribute *values* in a presentation token, where the predicate being proven is public to the verifier. SMS focuses on signing messages under a hidden function *structure* F, whose circuit topology is concealed from signatures, and adds per-signature accountability control absent from anonymous credentials, which typically target full unlinkability. Combining the two, producing an SMS signature while selectively disclosing which attributes the signer holds, is a natural direction for future work and is part of the broader functional credentials framework [24].

1.3 Notations, Conventions, and Paper Organization

Let $\emptyset$ and ϵ denote the empty set and empty string, respectively. We say a quantity p_λ (indexed by the security parameter λ) satisfies $p_\lambda \leq \mathsf{negl}(\lambda)$ if there is a negligible function $\epsilon(\lambda)$ and λ_0 such that $p_\lambda \leq \epsilon(\lambda)$ for all $\lambda \geq \lambda_0$. $x \leftarrow \mathcal{D}$ means x is drawn from distribution $\mathcal{D}$. $x \xleftarrow{\$} S$ means x is drawn uniformly from set S. Let $\mathbb{Z}$, $\mathbb{Z}_+$, and $\mathbb{N}$ denote the integers, positive integers, and natural numbers $\{0, 1, 2, \ldots\}$. For $a, b \in \mathbb{Z}$ with $a \leq b$, let $[a, b] = \{a, \ldots, b\}$. For $b \in \mathbb{Z}_+$, let $[b] = \{1, \ldots, b\}$. Bold lower/upper-case letters, *e.g.*, $\mathbf{x}/\mathbf{A}$, denote column vectors/matrices. $(\mathbf{x}\|\mathbf{y})$ denotes the concatenation of the vectors, $[\mathbf{x}^\top \mid \mathbf{y}^\top]^\top$.

Throughout, $\mathcal{M}$ denotes the *input* message space and $\overline{\mathcal{M}}$ denotes the *output* message space (the space of derived messages after applying the combiner P); elements are written $M \in \mathcal{M}$ and $\overline{M} \in \overline{\mathcal{M}}$. This distinction is maintained consistently across all definitions, constructions, and security experiments.

Paper organization. The next section defines SMS: syntax, correctness, and security. Section 3 presents the generic construction. Section 4 develops the PLONK arithmetization framework for proving hidden function evaluations, supporting the generic construction. Section 5 overviews the post-quantum lattice-based instantiation via ZKBoo and the modified decomposition. Full details of

the lattice-based instantiation, all primitives, proofs, parameter settings, and preliminaries are in the full version.

2 Definition of SMS

This section formalizes SMS by defining its syntax (Section 2.1) and security requirements (Section 2.2). The security experiments are collected in Figure 1; the supporting oracles are defined in Section 2.2 and referenced throughout. SMS subsumes other notions via the combiner P, using F and G to encode policies.

2.1 Syntax

We adopt the partial-dynamic model [45], where users may join at any time but revocation is not supported; a fully dynamic model would additionally allow membership expiry.

Parties. The *function-issuing authority* FA holds master function key mfk and enrolls users by certifying each user's attribute and hidden policy. The *tracing authority* TA holds tracing key tsk and recovers signer identity or role from an opening tag when authorized. Each *user* holds a certified attribute–policy pair (attr, F) and chooses a public ad hoc function G at each signing. *Verifiers* check signatures publicly using pp and the declared G.

Signing process. To sign a message $M \in \mathcal{M}$ with auxiliary witness $w \in \mathcal{W}$ supporting the computations of F and G, the signer computes

$$(\overline{M}, \mathsf{op}) = P\big(F(M, \mathsf{attr}, w),\, G(M, \mathsf{attr}, w)\big),$$

where P is the public combiner fixed in pp and $\overline{M} \in \overline{\mathcal{M}}$ is the derived message (equal to M when P returns the input message unchanged). If $\mathsf{op} \in \mathcal{OP}_{\mathsf{sgnbl}}$ the signer issues a signature Σ on $(\overline{M}, G)$, which also serves as a zero-knowledge proof of correct evaluation; if $\mathsf{op} \notin \mathcal{OP}_{\mathsf{sgnbl}}$ signing is blocked. The attribute attr and policy F remain hidden in Σ; only TA learns op via Open.

Definition 1 (SMS). *An SMS scheme is defined by the algorithms below.*

$\mathsf{Setup}(1^\lambda, N)$**:** *On input security parameter 1^λ and maximum user count N, run $(\mathsf{vk}, \mathsf{mfk}) \leftarrow \mathcal{S}.\mathsf{KeyGen}(1^\lambda)$ for FA and $(\mathsf{ek}, \mathsf{tsk}) \leftarrow \mathcal{E}.\mathsf{KeyGen}(1^\lambda)$ for TA, initialize $\mathsf{State} := \emptyset$, and publish the public parameter*

$$\mathsf{pp} = \big(\mathsf{gpk},\ \mathcal{M},\ \overline{\mathcal{M}},\ \mathcal{AS},\ \mathcal{W},\ \mathcal{F},\ \mathcal{G},\ P,\ \mathcal{OP},\ \mathcal{OP}_{\mathsf{sgnbl}}\big),$$

where gpk incorporates vk and ek, $\mathcal{M}$ and $\overline{\mathcal{M}}$ are the input and derived message spaces, $\mathcal{AS}$ and $\mathcal{W}$ are the attribute and witness spaces, $\mathcal{F}$ and $\mathcal{G}$ are the hidden and ad hoc function families, P is the public combiner, $\mathcal{OP}$ is the set of all opening values, and $\mathcal{OP}_{\mathsf{sgnbl}} \subsetneq \mathcal{OP}$ is the signable subset. Output $(\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk})$.

The strict inclusion ensures that $\mathcal{OP} \setminus \mathcal{OP}_{\mathsf{sgnbl}}$ *is non-empty, so the combiner* P *can block signing by outputting a value outside* $\mathcal{OP}_{\mathsf{sgnbl}}$*, encoding a policy rejection without producing a signature.*

The unified Setup *is retained here for notational convenience. In deployments where* FA *and* TA *are distinct,* Setup *can be split into two independent key generation algorithms:* $\mathsf{Setup}_{\mathsf{FA}}(1^\lambda)$ *outputting* $(\mathsf{vk}, \mathsf{mfk})$ *and* $\mathsf{Setup}_{\mathsf{TA}}(1^\lambda)$ *outputting* $(\mathsf{ek}, \mathsf{tsk})$*, with* pp *assembled from their public components, such that the two authority keys* mfk *and* tsk *are held by separate parties* FA *and* TA*.*

$\mathsf{Join}\ \langle \mathsf{U}(1^\lambda), \mathsf{FA}(\mathsf{mfk}) \rangle\ (\mathsf{pp}, \mathsf{State}, \mathsf{attr}, F)$**:** FA *enrolls user* U *with attribute* $\mathsf{attr} \in \mathcal{AS}$ *and assigns function* $F \in \mathcal{F}$*. Upon successful completion:*

1. U *obtains* $(\mathsf{sec}_{\mathsf{attr}}, \mathsf{cert}_{\mathsf{attr}})$*, where* $\mathsf{sec}_{\mathsf{attr}}$ *is the user secret key and* $\mathsf{cert}_{\mathsf{attr}}$ *is a certificate binding* (attr, F)*;*
2. FA *updates* $\mathsf{State}' \coloneqq (\mathsf{State} \| (\mathsf{attr}, F, \mathsf{trans}))$*, where* trans *is the transcript. The updated state* State' *replaces* State *in subsequent executions.*

If Join *fails,* U *receives* $\perp$*, and* State *is unchanged. By convention,* $\mathsf{cert}_{\mathsf{attr}}$ *encodes* (attr, F) *in the clear so that any party may read* (attr, F) *from* $\mathsf{cert}_{\mathsf{attr}}$*; the pair* (attr, F) *need only be concealed from signatures, not from certificates. We write* $\mathsf{sec}_{\mathsf{attr}} \leftrightharpoons_{\mathsf{pp}} \mathsf{cert}_{\mathsf{attr}}$ *to indicate that* $(\mathsf{sec}_{\mathsf{attr}}, \mathsf{cert}_{\mathsf{attr}})$ *was produced by an honest execution of* Join *under* pp*,* $\mathsf{sec}_{\mathsf{attr}} \not\leftrightharpoons_{\mathsf{pp}} \mathsf{cert}_{\mathsf{attr}}$ *otherwise.*

$\mathsf{Sign}(\mathsf{pp}, \mathsf{cert}_{\mathsf{attr}}, \mathsf{sec}_{\mathsf{attr}}, G, M, w)$**:** *Given* pp*, a certificate* $\mathsf{cert}_{\mathsf{attr}}$ *along with a secret* $\mathsf{sec}_{\mathsf{attr}}$*, a function* $G \in \mathcal{G}$*, a message* $M \in \mathcal{M}$*, and a witness* $w \in \mathcal{W}$*, output a derived message* $\overline{M} \in \overline{\mathcal{M}}$ *and a signature* Σ*. Return* $\perp$ *if it fails.*

$\mathsf{Verify}(\mathsf{pp}, G, \overline{M}, \Sigma)$**:** *Output* 1 *if* Σ *is a valid SMS signature on* $\overline{M} \in \overline{\mathcal{M}}$ *under function* $G \in \mathcal{G}$ *and parameters* pp*; output* 0 *otherwise.*

$\mathsf{Open}(\mathsf{pp}, \mathsf{tsk}, G, \overline{M}, \Sigma)$**:** *Using tracing key* tsk*, extract and return the opening value* $\mathsf{op} \in \mathcal{OP}$ *from a valid signature* Σ *on derived message* $\overline{M}$ *under function* G*. The information recoverable from* op *ranges from a role or partial identity to the full identity, depending on the configuration encoded in* P*.*

Illustration. A central bank issues digital currency (CBDC) to verified citizens. Each citizen's signing key encodes a certified hidden policy $F(M, \mathsf{attr}, w)$ that checks transaction validity against her know-your-customer compliance tier attr: whether the transaction amount is within her spending limit, the merchant category is permitted, and the note has not been spent before (via witness w). At each payment, the citizen additionally chooses a public function G encoding the transaction's public validity conditions, *e.g.*, that the amount is positive and the merchant is registered. The combiner P outputs the transaction record $\overline{M}$ and an opening tag op reflecting the transaction's risk level: routine payments yield $\mathsf{op} = \mathsf{pseudonym}$, exposing only a per-citizen unlinkable tag to the central bank; transactions flagged by F as anomalous yield $\mathsf{op} = \mathsf{id}$, exposing the citizen's full identity to the regulator. If F rejects the transaction outright, P outputs $\mathsf{op} \notin \mathcal{OP}_{\mathsf{sgnbl}}$, blocking the payment without producing a signature. The spending policy F remains hidden from merchants and the public; verifiers confirm only that G approved the transaction under some certified hidden policy.

2.2 Correctness and Security

The system maintains a public state State recording each enrolled user's attribute, assigned function, and enrollment transcript. State is *consistent* if every recorded transcript verifies under Join. A secure SMS scheme satisfies correctness, privacy, and unforgeability.

Correctness. Correctness requires that honest signatures verify and open correctly, that the system state remains consistent throughout, and that only enrolled users can produce valid signatures.

Definition 2 (Correctness). *For* $\mathsf{Setup}(1^\lambda)$ *returning* $(\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk})$*, an SMS scheme* $\mathcal{SMS}$ *is* correct *if the following conditions hold.*

1. $\mathcal{OP}_{\mathsf{sgnbl}} \subsetneq \mathcal{OP}$ *and all attributes in* State *are pairwise distinct.*
2. *If* $\mathsf{Join}\left\langle \mathsf{U}(1^\lambda), \mathsf{FA}(\mathsf{mfk}) \right\rangle (\mathsf{pp}, \mathsf{State}, \mathsf{attr}, F)$ *is executed honestly, with transcript* trans*:* $\mathsf{cert}_{\mathsf{attr}} \leftrightharpoons_{\mathsf{pp}} \mathsf{sec}_{\mathsf{attr}}$*,* trans *is valid, with* $(\mathsf{attr}, F, \mathsf{trans})$ *inserted into* State*.*
3. *For* $(\mathsf{attr}, F, \cdot) \in \mathsf{State}$ *with* $\mathsf{cert}_{\mathsf{attr}} \leftrightharpoons_{\mathsf{pp}} \mathsf{sec}_{\mathsf{attr}}$*,* $G \in \mathcal{G}$*,* $M \in \mathcal{M}$*,* $w \in \mathcal{W}$*:*

$$\Pr\left[\begin{array}{l|l} b = 1 & (\overline{M}, \Sigma) \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{cert}_{\mathsf{attr}}, \mathsf{sec}_{\mathsf{attr}}, G, M, w), \\ \wedge\ \mathsf{op} \in \mathcal{OP}_{\mathsf{sgnbl}} & (M', \mathsf{op}) = P(F(M, \mathsf{attr}, w), G(M, \mathsf{attr}, w)), \\ \wedge\ \mathsf{op} = \mathsf{op}' & b := \mathsf{Verify}(\mathsf{pp}, G, \overline{M}, \Sigma), \\ \wedge\ \overline{M} = M' & \mathsf{op}' \leftarrow \mathsf{Open}(\mathsf{pp}, \mathsf{tsk}, G, \overline{M}, \Sigma) \end{array}\right] = 1. \tag{2}$$

Adversary model. The adversary $\mathcal{A}$ interacts with the system via oracles that model realistic attack capabilities, defined below, and the security properties are parameterized by which oracles $\mathcal{A}$ may access. We write $(\cdot)^{\mathsf{condition}}$ to denote that an oracle returns ϵ when $\mathsf{condition}$ holds.

Oracles OMfk, OTsk. OMfk returns mfk to $\mathcal{A}$ and sets flag amfk to True. OTsk returns tsk and sets flag atsk to True. Both flags are initialized to False.

Oracle OCU, set $\mathcal{L}_{\mathsf{cu}}$. On query (attr, F) with $(\mathsf{attr}, \cdot) \notin \mathcal{L}_{\mathsf{cu}}$, OCU acts as an honest FA, runs Join, sets $\mathcal{L}_{\mathsf{cu}} := \mathcal{L}_{\mathsf{cu}} \cup \{(\mathsf{attr}, F)\}$, updates State, and returns $(\mathsf{sec}_{\mathsf{attr}}, \mathsf{cert}_{\mathsf{attr}})$ to $\mathcal{A}$.

Oracle OHU, set $\mathcal{L}_{\mathsf{hu}}$. On query attr with $(\mathsf{attr}, \cdot) \notin \mathcal{L}_{\mathsf{cu}}$, OHU acts as an honest FA, samples $F \in \mathcal{F}$, runs Join, and sets $\mathcal{L}_{\mathsf{hu}} := \mathcal{L}_{\mathsf{hu}} \cup \{(\mathsf{attr}, F)\}$ and updates State. The secrets of users in $\mathcal{L}_{\mathsf{hu}}$ are not given to $\mathcal{A}$. Both OCU and OHU guarantee that all attributes in State are distinct and $\mathcal{L}_{\mathsf{cu}} \cap \mathcal{L}_{\mathsf{hu}} = \emptyset$ on the attribute component.

Oracle OSign, set $\mathcal{L}_{\mathsf{sig}}$. On query (attr, G, M, w), if $\exists F$ such that $(\mathsf{attr}, F) \in \mathcal{L}_{\mathsf{hu}}$, $G \in \mathcal{G}$, $M \in \mathcal{M}$, $w \in \mathcal{W}$, and $\mathsf{op} \in \mathcal{OP}_{\mathsf{sgnbl}}$ where op is determined by P, F, G, M, attr, w, then OSign returns $(\overline{M}, \Sigma) \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{cert}_{\mathsf{attr}}, \mathsf{sec}_{\mathsf{attr}}, G, M, w)$ and inserts $(\mathsf{attr}, G, M, \Sigma)$ into $\mathcal{L}_{\mathsf{sig}}$.

Oracle OOpen. On query $(G, \overline{M}, \Sigma)$, return $\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{pp}, \mathsf{tsk}, G, \overline{M}, \Sigma)$.

Privacy. Privacy guarantees that SMS signatures reveal no information about the signer's attribute attr or hidden policy F. $\mathcal{A}$ chooses G and two candidate signers, each with a valid certificate, secret, message, and witness, and attempts to identify which signer produced the challenge signature. Both candidate signers must use the same G, since G is public and a verifier could trivially distinguish signatures produced under different public functions.

$\mathcal{A}$ may access OCU, OHU, OMfk, and OSign, but not OTsk or OOpen: access to tsk trivially breaks privacy, and allowing OOpen access requires public-key encryption to be secure against chosen-ciphertext attacks, a conceptually straightforward extension (*e.g.*, [58]). To prevent trivial wins, both candidate signers must yield the same derived message $\overline{M}$ and a signable opening value.

Definition 3 (Privacy). *Let* $\mathsf{E}_{\mathcal{A}}^{\text{priv-}b}(\lambda)$ *be defined as in Figure 1.* $\mathcal{SMS}$ *is* private *if, for any probabilistic polynomial-time (PPT)* $\mathcal{A}$*,*

$$\left|\Pr\left[\mathsf{E}_{\mathcal{A}}^{\text{priv-}0}(\lambda)=1\right]-\Pr\left[\mathsf{E}_{\mathcal{A}}^{\text{priv-}1}(\lambda)=1\right]\right|\leq \mathsf{negl}(\lambda).$$

Unforgeability. Unforgeability prevents any PPT $\mathcal{A}$ from producing a valid SMS signature without a legitimate certificate–secret pair. It encompasses three nested properties, formalized jointly in $\mathsf{E}_{\mathcal{A}}^{\mathsf{unforge}}(\lambda)$ (Figure 1).

- *Extractability.* Every valid forgery $(G,\overline{M},\Sigma)$ must extract to a list of witnesses $(\mathsf{attr}',F',M',w')$ such that $\mathsf{op}'\in\mathcal{OP}_{\mathsf{sgnbl}}$, $\mathsf{op}'=\mathsf{op}$ as opened by Open, and $\widetilde{M}=\overline{M}$. No adversary can produce a signature whose extracted witness is inconsistent with its opening value or its derived message.
- *Type-1 unforgeability (traceability).* Extractability holds, and $\mathcal{A}$ cannot forge a signature whose extracted (attr',F') lies outside the corrupted set $\mathcal{L}_{\mathsf{cu}}$.
- *Type-2 unforgeability (non-frameability).* Extractability holds, and no coalition of users, FA, and TA can forge a signature whose extracted (attr',F') belongs to $\mathcal{L}_{\mathsf{hu}}$.

The two unforgeability types are independent, as the table below shows.

	Type-1 (Traceability)	**Type-2 (Non-frameability)**
Forgery target	$(\mathsf{attr}',F')\notin\mathcal{L}_{\mathsf{cu}}$	$(\mathsf{attr}',F')\in\mathcal{L}_{\mathsf{hu}}$
Covers	$\mathcal{L}_{\mathsf{hu}}\cup(\mathcal{AS}\setminus(\mathcal{L}_{\mathsf{cu}}\cup\mathcal{L}_{\mathsf{hu}}))$	$\mathcal{L}_{\mathsf{cu}}\cup(\mathcal{AS}\setminus(\mathcal{L}_{\mathsf{cu}}\cup\mathcal{L}_{\mathsf{hu}}))$
OMfk access	Restricted	Restricted

Neither type implies the other: Type-2 covers $\mathcal{L}_{\mathsf{cu}}$ but not $\mathcal{L}_{\mathsf{hu}}$; Type-1 covers $\mathcal{L}_{\mathsf{hu}}$ but not $\mathcal{L}_{\mathsf{cu}}$. A forgery is non-trivial only if $(G,\overline{M},\Sigma)$ was not previously obtained via OSign, enforced by the $\mathcal{L}_{\mathsf{sig}}$ membership check in $\mathsf{E}^{\mathsf{unforge}}$.

Extractability permits OMfk access in $\mathsf{E}^{\mathsf{unforge}}$ because it only requires consistency of extracted components with the system prescription, independent of membership. Conversely, for Type-1 and Type-2 unforgeability, OMfk access is restricted, because possessing mfk would allow $\mathcal{A}$ to certify attributes outside both $\mathcal{L}_{\mathsf{cu}}$ and $\mathcal{L}_{\mathsf{hu}}$, trivially breaking the membership-based winning conditions;

Experiment $\mathsf{E}^{\mathsf{priv}\text{-}b}_{\mathcal{A}}(\lambda)$**:**
$(\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
$\mathsf{amfk} \coloneqq \text{False}$, $\mathcal{L}_{\mathsf{cu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{hu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{sig}} \coloneqq \emptyset$.
$(\mathsf{aux}, G, (\mathsf{cert}_{\mathsf{attr}_i}, \mathsf{sec}_{\mathsf{attr}_i}, M_i, w_i)_{i\in\{0,1\}}) \leftarrow \mathcal{A}^{\mathsf{OCU},\mathsf{OHU},\mathsf{OMfk},\mathsf{OSign}}(\mathsf{pp})$.
Obtain (attr_i, F_i) from $\mathsf{cert}_{\mathsf{attr}_i}$ for $i \in \{0,1\}$.
If $G \notin \mathcal{G}$ or $(\exists i \in \{0,1\}\colon M_i \notin \mathcal{M} \vee \mathsf{attr}_i \notin \mathcal{AS} \vee F_i \notin \mathcal{F} \vee \mathsf{sec}_{\mathsf{attr}_i} \not\leftrightharpoons_{\mathsf{pp}} \mathsf{cert}_{\mathsf{attr}_i})$, return 0.
$(\overline{M}_i, \mathsf{op}_i) \coloneqq P(F_i(M_i, \mathsf{attr}_i, w_i), G(M_i, \mathsf{attr}_i, w_i))$ for $i \in \{0,1\}$.
If $\overline{M}_0 \neq \overline{M}_1 \vee \mathsf{op}_0 \notin \mathcal{OP}_{\mathsf{sgnbl}} \vee \mathsf{op}_1 \notin \mathcal{OP}_{\mathsf{sgnbl}}$, return 0.
Set $(\overline{M}, \Sigma) \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{cert}_{\mathsf{attr}_b}, \mathsf{sec}_{\mathsf{attr}_b}, G, M_b, w_b)$.
$b' \leftarrow \mathcal{A}^{\mathsf{OCU},\mathsf{OHU},\mathsf{OMfk},\mathsf{OSign}}(\mathsf{pp}, \overline{M}, \Sigma, \mathsf{aux})$.
Return b'.

Experiment $\mathsf{E}^{\mathsf{real}}_{\mathcal{A}}(\lambda)$**:**
$(\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
$\mathsf{amfk} \coloneqq \text{False}$, $\mathsf{atsk} \coloneqq \text{False}$, $\mathcal{L}_{\mathsf{cu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{hu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{sig}} \coloneqq \emptyset$.
$b \leftarrow \mathcal{A}^{\mathsf{OCU},\mathsf{OHU},\mathsf{OMfk},\mathsf{OTsk},\mathsf{OSign},\mathsf{OOpen}}(\mathsf{pp})$.
Return b.

Experiment $\mathsf{E}^{\mathsf{sim}}_{\mathcal{A}}(\lambda)$**:**
$(\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk}, \tau_{\mathsf{ext}}) \leftarrow \mathsf{SimSetup}(1^\lambda)$. //Change $\mathsf{Setup}(1^\lambda)$ in $\mathsf{E}^{\mathsf{real}}_{\mathcal{A}}(\lambda)$ to $\mathsf{SimSetup}(1^\lambda)$
$\mathsf{amfk} \coloneqq \text{False}$, $\mathsf{atsk} \coloneqq \text{False}$, $\mathcal{L}_{\mathsf{cu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{hu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{sig}} \coloneqq \emptyset$.
$b \leftarrow \mathcal{A}^{\mathsf{OCU},\mathsf{OHU},\mathsf{OMfk},\mathsf{OTsk},\mathsf{OSign},\mathsf{OOpen}}(\mathsf{pp})$.
Return b.

Experiment $\mathsf{E}^{\mathsf{unforge}}_{\mathcal{A}}(\lambda)$**:**
$(\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk}, \tau_{\mathsf{ext}}) \leftarrow \mathsf{SimSetup}(1^\lambda)$.
$\mathsf{amfk} \coloneqq \text{False}$, $\mathsf{atsk} \coloneqq \text{False}$, $\mathcal{L}_{\mathsf{cu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{hu}} \coloneqq \emptyset$, $\mathcal{L}_{\mathsf{sig}} \coloneqq \emptyset$.
$(G, \overline{M}, \Sigma) \leftarrow \mathcal{A}^{\mathsf{OCU},\mathsf{OHU},\mathsf{OMfk},\mathsf{OTsk},\mathsf{OSign},\mathsf{OOpen}}(\mathsf{pp})$.
If $G \notin \mathcal{G} \vee \overline{M} \notin \overline{\mathcal{M}} \vee \mathsf{Verify}(\mathsf{pp}, G, \overline{M}, \Sigma) = 0$, return 0.
$(\mathsf{attr}', F', M', w') \leftarrow \mathsf{Extract}(\mathsf{pp}, \tau_{\mathsf{ext}}, G, \overline{M}, \Sigma)$.
If $\mathsf{attr}' \notin \mathcal{AS} \vee F' \notin \mathcal{F} \vee M' \notin \mathcal{M} \vee w' \notin \mathcal{W} \vee (\mathsf{attr}', G, M', \Sigma) \in \mathcal{L}_{\mathsf{sig}}$, return 0.
$\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{pp}, \mathsf{tsk}, G, \overline{M}, \Sigma)$.
$(\widetilde{M}, \mathsf{op}') \coloneqq P(F'(M', \mathsf{attr}', w'), G(M', \mathsf{attr}', w'))$.
Return 1 if (3) holds; otherwise return 0.

$$
\begin{aligned}
&(\mathsf{op}' \notin \mathcal{OP}_{\mathsf{sgnbl}} \vee \mathsf{op} \neq \mathsf{op}' \vee \overline{M} \neq \widetilde{M}) && \text{//extractability} \\
&\vee (\mathsf{amfk} = \text{False} \wedge (\mathsf{attr}', F') \notin \mathcal{L}_{\mathsf{cu}}) && \text{//type-1 unforgeability} \\
&\vee (\mathsf{amfk} = \text{False} \wedge (\mathsf{attr}', F') \in \mathcal{L}_{\mathsf{hu}}) && \text{//type-2 unforgeability}
\end{aligned} \tag{3}
$$

Fig. 1. Experiments for security of SMS

in particular, $\mathcal{A}$ could certify the same attribute as an honest user under a different hidden function (*e.g.*, $(\mathsf{attr}, F) \in \mathcal{L}_{\mathsf{hu}}$ but $(\mathsf{attr}, F') \notin \mathcal{L}_{\mathsf{hu}}$ where $F' \neq F$), producing a valid signature outside the honest-user record.

Simulated setup for extraction. We devise algorithms $\mathsf{SimSetup}$ and $\mathsf{Extract}$ to enable simulation-sound extractability [18]: a simulated setup indistinguishable from the real one, with a trapdoor for extracting the underlying witness.

Definition 4 (Simulated setup and extraction).

$\mathsf{SimSetup}(1^\lambda)$**:** *Identical to* Setup *but additionally outputs extraction trapdoor* τ_{ext}*. Output:* $(\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk}, \tau_{\mathsf{ext}})$.

$\mathsf{Extract}(\mathsf{pp}, \tau_{\mathsf{ext}}, G, \overline{M}, \Sigma)$**:** *Given* pp*, trapdoor* τ_{ext}*,* $G \in \mathcal{G}$*,* $\overline{M} \in \overline{\mathcal{M}}$*, and signature* Σ*, return* $(\mathsf{attr}, F, M, w) \in \mathcal{AS} \times \mathcal{F} \times \mathcal{M} \times \mathcal{W}$.

The correctness of SMS under $\mathsf{SimSetup}$ is defined identically to Definition 2 with Setup replaced by $\mathsf{SimSetup}$; additionally, the extracted components must match those used to create the signature. Setup indistinguishability implies this correctness against PPT adversaries, while correctness of the simulated setup holds unconditionally.

Definition 5 (Setup indistinguishability). $\mathcal{SMS}$ *satisfies* setup indistinguishability *if, for any PPT* $\mathcal{A}$*,*

$$\left|\Pr\left[\mathsf{E}^{\mathsf{real}}_{\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{E}^{\mathsf{sim}}_{\mathcal{A}}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{E}^{\mathsf{real}}_{\mathcal{A}}(\lambda)$ *and* $\mathsf{E}^{\mathsf{sim}}_{\mathcal{A}}(\lambda)$ *are defined in Figure 1.*

Definition 6 (Unforgeability). $\mathcal{SMS}$ *is* unforgeable *if, for any PPT* $\mathcal{A}$*,*

$$\Pr\left[\mathsf{E}^{\mathsf{unforge}}_{\mathcal{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{E}^{\mathsf{unforge}}_{\mathcal{A}}(\lambda)$ *is defined in Figure 1.*

3 Generic Construction of SMS

With the SMS notion established, we build a generic scheme from standard primitives, treating the hidden-circuit NIZKAoK as an abstract component instantiated in Section 4. Following Nguyen *et al.* [58], each enrolled user holds an FA-issued certificate $\mathsf{cert}_{\mathsf{attr}} = (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}, \sigma)$ binding their attribute, hidden policy, and a per-user verification key, where $\sigma \leftarrow \mathcal{S}.\mathsf{Sign}(\mathsf{mfk}, (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}))$ and the global verification key vk is embedded in gpk.

At signing, the user encrypts the opening tag op under TA's public key ek, producing a ciphertext $\mathsf{ct}_{\mathsf{op}}$ that TA can later decrypt to recover op. A NIZKAoK proof $\Pi_{\mathsf{g\text{-}SMS}}$ certifies correct evaluation of $P(F(\cdot), G(\cdot))$, validity of the membership certificate, and correct encryption of op, without revealing attr, F, or any intermediate computation trace. How the NIZKAoK handles the hidden evaluation of F via PLONK's arithmetization is developed in Section 4; here, we

treat the NIZKAoK as an abstract component for arithmetic circuit satisfiability. In the SMS instance for the CBDC example, F checks the spending limit, G validates the merchant, and P sets op to the pseudonym or identity tag.

Without further binding, an adversary could modify $(\mathsf{ct}_{\mathsf{op}}, \Pi_{\text{g-SMS}})$ to produce a new valid SMS signature. The user generates a one-time key pair $(\mathsf{ovk}, \mathsf{osk})$ at the time of signing, certifies ovk under $\mathsf{sk}_{\mathsf{attr}}$, and computes a one-time signature $\sigma_{\text{one-time}}$ over all signature components. The NIZKAoK then proves validity of $\sigma_{\text{one-time}}$ under $\mathsf{vk}_{\mathsf{attr}}$ rather than proving the secret key $\mathsf{sk}_{\mathsf{attr}}$ directly, binding all components together against post hoc modification.

3.1 Construction

Suppose $\mathcal{S} = (\mathcal{S}.\mathsf{KeyGen}, \mathcal{S}.\mathsf{Sign}, \mathcal{S}.\mathsf{Verify})$ is a secure signature scheme, $\mathcal{OTS} = (\mathcal{OTS}.\mathsf{KeyGen}, \mathcal{OTS}.\mathsf{Sign}, \mathcal{OTS}.\mathsf{Verify})$ is a strongly unforgeable one-time signature scheme, $\mathcal{E} = (\mathcal{E}.\mathsf{KeyGen}, \mathcal{E}.\mathsf{Enc}, \mathcal{E}.\mathsf{Dec})$ is a CPA-secure PKE scheme with randomness space $\mathsf{R}_{\mathsf{encrypt}}$. We also use a NIZKAoK system $\mathcal{NIZKAoK}$ for arithmetic circuit satisfiability. The message space $\mathcal{M}$, derived message space $\overline{\mathcal{M}} = \mathbb{F}^{\ell}_{\mathsf{msg}}$, attribute space $\mathcal{AS}$, witness space $\mathcal{W}$, function families $\mathcal{F}$ and $\mathcal{G}$, combiner P, opening set $\mathcal{OP}$, and signable subset $\mathcal{OP}_{\mathsf{sgnbl}} \subsetneq \mathcal{OP}$ are determined as part of Setup.

All computations are over $\mathbb{F}$ (instantiated as $\mathbb{Z}_q$ for a prime q). Each $F \in \mathcal{F}$ and $G \in \mathcal{G}$ share output length n_{out}, so the combiner has type $P\colon \mathbb{F}^{2n_{\mathsf{out}}} \to \overline{\mathcal{M}} \times \mathcal{OP}$. The witness decomposes as $\mathbf{w} = (\mathbf{w}_F \| \mathbf{w}_G)$, where $\mathbf{w}_F$ and $\mathbf{w}_G$ support the computations of F and G, respectively.

$\mathsf{Setup}(1^\lambda, N) \to (\mathsf{pp}, \mathsf{State}, \mathsf{mfk}, \mathsf{tsk})$**:** Sample $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathcal{E}.\mathsf{KeyGen}(1^\lambda)$ for TA, $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{KeyGen}(1^\lambda)$ for FA, and lastly $\mathsf{pp}_{\mathsf{nizk}}$ as the public parameter of $\mathcal{NIZKAoK}$. Set $\mathsf{gpk} \coloneqq (\mathsf{ek}, \mathsf{vk}, \mathsf{pp}_{\mathsf{nizk}})$, $\mathsf{mfk} \coloneqq \mathsf{sk}$, $\mathsf{tsk} \coloneqq \mathsf{dk}$, and $\mathsf{State} \coloneqq \emptyset$. Determine $\mathcal{M}$, $\overline{\mathcal{M}}$, $\mathcal{AS}$, $\mathcal{W}$, $\mathcal{F}$, $\mathcal{G}$, P, $\mathcal{OP}$, $\mathcal{OP}_{\mathsf{sgnbl}}$. Return

$$\mathsf{pp} \coloneqq (\mathsf{gpk}, \mathcal{M}, \overline{\mathcal{M}}, \mathcal{AS}, \mathcal{W}, \mathcal{F}, \mathcal{G}, P, \mathcal{OP}, \mathcal{OP}_{\mathsf{sgnbl}}),\ \mathsf{State},\ \mathsf{mfk},\ \mathsf{tsk}.$$

$\mathsf{Join}\left\langle \mathsf{U}(1^\lambda), \mathsf{FA}(\mathsf{mfk}) \right\rangle (\mathsf{pp}, \mathsf{State}, \mathsf{attr}, F) \to (\mathsf{State}', (\mathsf{cert}_{\mathsf{attr}}, \mathsf{sec}_{\mathsf{attr}}))$**:** For user enrollment, user U and function-issuing authority FA proceed as follows.

1. U computes $\sigma_{\mathsf{enrl}} \leftarrow \mathcal{S}.\mathsf{Sign}(\mathsf{sk}_{\mathsf{attr}}, (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}))$, where $(\mathsf{vk}_{\mathsf{attr}}, \mathsf{sk}_{\mathsf{attr}}) \leftarrow \mathcal{S}.\mathsf{KeyGen}(1^\lambda)$, and sends $(\mathsf{vk}_{\mathsf{attr}}, \sigma_{\mathsf{enrl}})$ to FA.
2. FA aborts if $\mathcal{S}.\mathsf{Verify}(\mathsf{vk}_{\mathsf{attr}}, (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}), \sigma_{\mathsf{enrl}}) = 0$ or attr is already in State. Otherwise, FA computes $\sigma \leftarrow \mathcal{S}.\mathsf{Sign}(\mathsf{mfk}, (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}))$, sends σ to U, and updates $\mathsf{State}' \coloneqq (\mathsf{State} \| (\mathsf{attr}, F, \mathsf{trans} \coloneqq (\mathsf{vk}_{\mathsf{attr}}, \sigma_{\mathsf{enrl}}, \sigma)))$. The updated state State' replaces State in subsequent executions.
3. U proceeds if $\mathcal{S}.\mathsf{Verify}(\mathsf{vk}, (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}), \sigma) = 1$. Otherwise, U aborts.
4. U sets $\mathsf{cert}_{\mathsf{attr}} \coloneqq (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}, \sigma)$ and $\mathsf{sec}_{\mathsf{attr}} \coloneqq \mathsf{sk}_{\mathsf{attr}}$.

$\mathsf{Sign}(\mathsf{pp}, \mathsf{cert}_{\mathsf{attr}}, \mathsf{sec}_{\mathsf{attr}}, G, M, w) \to (\overline{M}, \Sigma)$**:** To sign a message M,

1. Parse $\mathsf{gpk} = (\mathsf{ek}, \mathsf{vk}, \mathsf{pp}_{\mathsf{nizk}})$, $\mathsf{cert}_{\mathsf{attr}} = (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}, \sigma)$, $\mathsf{sec}_{\mathsf{attr}} = \mathsf{sk}_{\mathsf{attr}}$.
2. Compute $(\overline{M}, \mathsf{op}) \coloneqq P(F(M, \mathsf{attr}, w), G(M, \mathsf{attr}, w))$.
3. Abort if $\mathsf{op} \notin \mathcal{OP}_{\mathsf{sgnbl}}$. Set $\mathsf{ct}_{\mathsf{op}} \coloneqq \mathcal{E}.\mathsf{Enc}(\mathsf{ek}, \mathsf{op}; r_{\mathsf{op}})$ with $r_{\mathsf{op}} \leftarrow \mathsf{R}_{\mathsf{encrypt}}$.

4. Generate $(\mathsf{ovk}, \mathsf{osk}) \leftarrow \mathcal{OTS}.\mathsf{KeyGen}(1^\lambda)$ and $\sigma_{\mathsf{ovk}} \leftarrow \mathcal{S}.\mathsf{Sign}(\mathsf{sk}_{\mathsf{attr}}, \mathsf{ovk})$.
5. Execute the prover of $\mathcal{NIZK}\mathcal{A}o\mathcal{K}$ to generate $\Pi_{\text{g-SMS}}$, for $\mathrm{R}_{\text{g-SMS}}$ (as in (4)) with the statement $(\mathsf{pp}, G, \overline{M}, \mathsf{ct}_{\mathsf{op}}, \mathsf{ovk})$ and witness $(\mathsf{attr}, \mathsf{vk}_{\mathsf{attr}}, F, \sigma, M, w, \mathsf{op}, r_{\mathsf{op}}, \sigma_{\mathsf{ovk}})$.
6. Compute $\sigma_{\text{one-time}} \leftarrow \mathcal{OTS}.\mathsf{Sign}(\mathsf{osk}, (G, \overline{M}, \mathsf{ct}_{\mathsf{op}}, \Pi_{\text{g-SMS}}))$.
7. Return $\big(\overline{M},\ \Sigma_{\text{g-SMS}}\big) = \big(\overline{M},\ (\mathsf{ct}_{\mathsf{op}}, \Pi_{\text{g-SMS}}, \mathsf{ovk}, \sigma_{\text{one-time}})\big)$.

The relation $\mathrm{R}_{\text{g-SMS}}$ certified by $\Pi_{\text{g-SMS}}$ is:

$$(\mathsf{pp}, G, \overline{M}, \mathsf{ct}_{\mathsf{op}}, \mathsf{ovk};\ \mathsf{attr}, \mathsf{vk}_{\mathsf{attr}}, F, \sigma, M, w, \mathsf{op}, r_{\mathsf{op}}, \sigma_{\mathsf{ovk}}) \in \mathrm{R}_{\text{g-SMS}} \tag{4}$$

$$\iff \begin{cases} \mathcal{S}.\mathsf{Verify}(\mathsf{vk}, (\mathsf{attr}, F, \mathsf{vk}_{\mathsf{attr}}), \sigma) = 1, \\ (\overline{M}, \mathsf{op}) = P(F(M, \mathsf{attr}, w), G(M, \mathsf{attr}, w)) \ \wedge\ \mathsf{op} \in \mathcal{OP}_{\mathsf{sgnbl}}, \\ \mathsf{ct}_{\mathsf{op}} = \mathcal{E}.\mathsf{Enc}(\mathsf{ek}, \mathsf{op};\ r_{\mathsf{op}}) \ \wedge\ \mathcal{S}.\mathsf{Verify}(\mathsf{vk}_{\mathsf{attr}}, \mathsf{ovk}, \sigma_{\mathsf{ovk}}) = 1. \end{cases}$$

The relation captures three obligations simultaneously: the signer holds a valid FA-issued certificate over (attr, F); the combiner evaluation is correct and the resulting tag is signable; and op is honestly encrypted for TA under ek.

$\mathsf{Verify}(\mathsf{pp}, G, \overline{M}, \Sigma_{\text{g-SMS}}) \to \{0,1\}$**:** To verify $\Sigma_{\text{g-SMS}}$:

1. Parse $\mathsf{gpk} = (\mathsf{ek}, \mathsf{vk}, \mathsf{pp}_{\mathsf{nizk}})$ and $\Sigma_{\text{g-SMS}} = (\mathsf{ct}_{\mathsf{op}}, \Pi_{\text{g-SMS}}, \mathsf{ovk}, \sigma_{\text{one-time}})$.
2. Return 1 if and only if $\mathcal{OTS}.\mathsf{Verify}(\mathsf{ovk}, (G, \overline{M}, \mathsf{ct}_{\mathsf{op}}, \Pi_{\text{g-SMS}}), \sigma_{\text{one-time}}) = 1$ and $\Pi_{\text{g-SMS}}$ is valid for $\mathrm{R}_{\text{g-SMS}}$ with public parameter $\mathsf{pp}_{\mathsf{nizk}}$ and statement $(\mathsf{ek}, \mathsf{vk}, G, \overline{M}, \mathsf{ct}_{\mathsf{op}}, \mathsf{ovk})$.

$\mathsf{Open}(\mathsf{pp}, \mathsf{tsk}, G, \overline{M}, \Sigma_{\text{g-SMS}}) \to \mathcal{OP}$**:** Parse $\Sigma_{\text{g-SMS}} = (\mathsf{ct}_{\mathsf{op}}, \Pi_{\text{g-SMS}}, \mathsf{ovk}, \sigma_{\text{one-time}})$ and $\mathsf{tsk} = \mathsf{dk}$. Return $\mathsf{op}' := \mathcal{E}.\mathsf{Dec}(\mathsf{dk}, \mathsf{ct}_{\mathsf{op}})$.

3.2 Analysis

Theorem 1 (Security). *$\mathcal{SMS}_{\text{generic}}$ satisfies correctness, privacy, setup indistinguishability, and unforgeability (Definitions 2, 3, 5, and 6), assuming that $\mathcal{NIZK}\mathcal{A}o\mathcal{K}$ is a simulation-sound extractable argument of knowledge, $\mathcal{S}$ is existentially unforgeable, $\mathcal{OTS}$ is strongly unforgeable, and $\mathcal{E}$ is CPA-secure.*

Proof (Sketch). Correctness follows directly from the correctness of the underlying primitives. We order the properties by logical dependency.

Privacy. The only signature-dependent information visible to the adversary is $\mathsf{ct}_{\mathsf{op}}$ and $\Pi_{\text{g-SMS}}$. Since the two candidate signers produce the same $\overline{M}$ and the same op (enforced by the privacy experiment), $\mathsf{ct}_{\mathsf{op}}$ is an encryption of the same value in both cases; CPA security of $\mathcal{E}$ ensures $\mathsf{ct}_{\mathsf{op}}$ is indistinguishable. The zero-knowledge property of the $\mathcal{NIZK}\mathcal{A}o\mathcal{K}$ ensures $\Pi_{\text{g-SMS}}$ reveals no information about the witness, including attr, F, and the computation trace. Together, the two components are jointly indistinguishable across the two signers.

Setup indistinguishability. SimSetup is identical to Setup except that it additionally generates an extraction trapdoor τ_{ext} (deferred to the full version). All other components, namely, $\mathcal{S}$, $\mathcal{OTS}$, $\mathcal{E}$, are executed identically in the real and

simulated setups. Setup indistinguishability follows from the simulation indistinguishability of $\mathcal{NIZKAoK}$: any adversary distinguishing the two setups yields a distinguisher for $\mathcal{NIZKAoK}$'s simulation indistinguishability.

Extractability. Given a valid forgery $(G, \overline{M}, \Sigma_{\mathsf{g\text{-}SMS}})$, $\mathsf{Extract}$ runs $\mathcal{NIZKAoK}$'s extractor on $\Pi_{\mathsf{g\text{-}SMS}}$ using trapdoor τ_{ext} to recover $(\mathsf{attr}', F', M', w', \mathsf{op}', r'_{\mathsf{op}}, \sigma'_{\mathsf{ovk}})$. The simulation-sound extractability [18] guarantees that the extracted witness satisfies $\mathrm{R}_{\mathsf{g\text{-}SMS}}$, so $\mathsf{op}' \in \mathcal{OP}_{\mathsf{sgnbl}}$, $\mathsf{op}' = \mathsf{op}$, and $\widetilde{M} = \overline{M}$ hold.

Type-1 unforgeability (traceability). Suppose $\mathcal{A}$ forges a signature with extracted $(\mathsf{attr}', F') \notin \mathcal{L}_{\mathsf{cu}}$. By extractability, the extracted σ' verifies, meaning $\mathcal{S}.\mathsf{Verify}(\mathsf{vk}, (\mathsf{attr}', F', \mathsf{vk}'_{\mathsf{attr}}), \sigma') = 1$. Since (attr', F') was never enrolled by FA (not in $\mathcal{L}_{\mathsf{cu}}$), no honest Join execution produced σ', and $\mathcal{A}$ did not receive mfk (as OMfk is restricted). This yields a forgery against $\mathcal{S}$ under verification key vk.

Type-2 unforgeability (non-frameability). Suppose $\mathcal{A}$ forges a signature with extracted $(\mathsf{attr}', F') \in \mathcal{L}_{\mathsf{hu}}$. By extractability, $\mathcal{S}.\mathsf{Verify}(\mathsf{vk}'_{\mathsf{attr}}, \mathsf{ovk}', \sigma'_{\mathsf{ovk}}) = 1$, where $\mathsf{vk}'_{\mathsf{attr}}$ is the honest user's verification key recorded in State. The one-time signature $\sigma_{\mathsf{one\text{-}time}}$ on $(G, \overline{M}, \mathsf{ct}_{\mathsf{op}}, \Pi_{\mathsf{g\text{-}SMS}})$ verifies under ovk', which was never signed by the honest user's $\mathsf{sk}'_{\mathsf{attr}}$ (since the forgery is fresh relative to $\mathcal{L}_{\mathsf{sig}}$). This yields a forgery against the existential unforgeability of $\mathcal{S}$ under $\mathsf{vk}'_{\mathsf{attr}}$, the honest user's per-user verification key.

The role of $\mathcal{OTS}$ is distinct: strong unforgeability of $\mathcal{OTS}$ prevents an adversary from taking a legitimately obtained signature $(\mathsf{ct}_{\mathsf{op}}, \Pi_{\mathsf{g\text{-}SMS}}, \mathsf{ovk}, \sigma_{\mathsf{one\text{-}time}})$ and modifying any component to produce a new valid signature, since $\sigma_{\mathsf{one\text{-}time}}$ binds all components under ovk, which is itself certified under $\mathsf{vk}_{\mathsf{attr}}$.

Details (*e.g.*, $\mathsf{SimSetup}$ and $\mathsf{Extract}$) are deferred to the full version. □

4 Proving Hidden Function Evaluations via PLONK

The key observation is that PLONKish arithmetization encodes any arithmetic circuit C via a description vector pd of length linear in the gate count of C, and crucially, this encoding has the same algebraic structure for any two circuits of the same gate count, independent of circuit topology. Treating pd as part of the *witness* rather than the statement therefore reduces hidden-circuit satisfiability to verifying a single fixed public circuit of the same linear size, without universal circuits and without non-standard assumptions.

Section 4.1 introduces the PLONK constraint system and reduces it to a purely quadratic form compatible with standard NIZKAoK proof systems. Section 4.2 applies this framework to the SMS evaluation structure, merging the hidden-circuit proof for F with the public-circuit proofs for G and P into a single merged circuit $\mathsf{C}_{\mathsf{merge}}$. Throughout this section, let C be an arithmetic circuit over $\mathbb{F}$ with n_{in} inputs, n_{gate} gates, and $n_{\mathsf{w}} \coloneqq n_{\mathsf{in}} + 3n_{\mathsf{gate}}$.

In the CBDC example, $\mathsf{C}_{\mathsf{merge}}$ coalesces F's limit checks, G's merchant validation, and P's risk scoring into a single verifiable statement.

4.1 Transforming PLONK's Constraints to Quadratic Constraints

PLONK's constraint system. PLONK's arithmetization [34] encodes C via selectors $\{s_i^{\mathsf{lf}}, s_i^{\mathsf{rg}}, s_i^{\mathsf{mlt}}, s_i^{\mathsf{cnst}}\}_{i=1}^{n_{\mathsf{gate}}}$ and a permutation $\varphi\colon [n_{\mathsf{w}}] \to [n_{\mathsf{w}}]$ encoding wire-sharing. Let $\ell_{\mathsf{plonk}} \coloneqq 4n_{\mathsf{gate}} + n_{\mathsf{w}}$. Following the formulation of Tang, Pham, and Ngo [64], these are collected into the *PLONK description*:

$$\mathsf{pd} = \left((s_i^{\mathsf{lf}})_{i=1}^{n_{\mathsf{gate}}} \parallel (s_i^{\mathsf{rg}})_{i=1}^{n_{\mathsf{gate}}} \parallel (s_i^{\mathsf{mlt}})_{i=1}^{n_{\mathsf{gate}}} \parallel (s_i^{\mathsf{cnst}})_{i=1}^{n_{\mathsf{gate}}} \parallel (\varphi(i))_{i=1}^{n_{\mathsf{w}}}\right) \in \mathbb{F}^{\ell_{\mathsf{plonk}}}, \tag{5}$$

This vector fully encodes the circuit's gate structure and wiring.

For any input $\mathbf{x} \in \mathbb{F}_{\mathsf{in}}^n$, evaluating C yields gate-value sequences $\mathbf{a} = (a_i)_{i=1}^{n_{\mathsf{gate}}}$, $\mathbf{b} = (b_i)_{i=1}^{n_{\mathsf{gate}}}$, $\mathbf{c} = (c_i)_{i=1}^{n_{\mathsf{gate}}}$, and output $\mathbf{c}_{\mathsf{out}} = (c_i)_{i=n_{\mathsf{gate}}-n_{\mathsf{out}}+1}^{n_{\mathsf{gate}}}$ for some $n_{\mathsf{out}} \le n_{\mathsf{gate}}$. The *extended witness* is:

$$\widetilde{\mathbf{w}} = (\mathbf{x} \parallel \mathbf{a} \parallel \mathbf{b} \parallel \mathbf{c}) = (w_i)_{i=1}^{n_{\mathsf{w}}} \in \mathbb{F}^{n_{\mathsf{w}}}. \tag{6}$$

Together with pd, $\widetilde{\mathbf{w}}$ carries all information needed to verify C's evaluation.

For random challenges $\alpha, \beta \in \mathbb{F}$, the output $\mathbf{c}_{\mathsf{out}}$ is correct if and only if:

$$\begin{cases} s_i^{\mathsf{lf}} \cdot w_{n_{\mathsf{in}}+i} + s_i^{\mathsf{rg}} \cdot w_{n_{\mathsf{in}}+n_{\mathsf{gate}}+i} \\ \quad + s_i^{\mathsf{mlt}} \cdot (w_{n_{\mathsf{in}}+i} \cdot w_{n_{\mathsf{in}}+n_{\mathsf{gate}}+i}) & \forall i \in [n_{\mathsf{gate}}], \\ \quad + s_i^{\mathsf{cnst}} - w_{n_{\mathsf{in}}+2n_{\mathsf{gate}}+i} = 0 & \\ \prod_{i=1}^{n_{\mathsf{w}}} (\alpha + \beta \cdot i + w_i) = \prod_{i=1}^{n_{\mathsf{w}}} (\alpha + \beta \cdot \varphi(i) + w_i), \quad \alpha, \beta \xleftarrow{\$} \mathbb{F}. \end{cases} \tag{7}$$

The first equation encodes gate constraints; the second encodes wire consistency via a grand product check. This system can be viewed as a circuit $\mathcal{U}^{\mathsf{plonk}}$ taking $(\mathsf{pd}, \widetilde{\mathbf{w}}, \alpha, \beta)$ as inputs and verifying whether $\mathsf{C}(\mathbf{x}) = \mathbf{c}_{\mathsf{out}}$. The soundness error is at most $n_{\mathsf{w}}/|\mathbb{F}|$.

Reduction to quadratic constraints. The gate constraint in (7) contains the cubic term $s_i^{\mathsf{mlt}} \cdot (w_{n_{\mathsf{in}}+i} \cdot w_{n_{\mathsf{in}}+n_{\mathsf{gate}}+i})$. To reduce to purely quadratic constraints, introduce auxiliary variables: $\widehat{w}_i \coloneqq w_{n_{\mathsf{in}}+i} \cdot w_{n_{\mathsf{in}}+n_{\mathsf{gate}}+i}$ for each $i \in [n_{\mathsf{gate}}]$. For the grand product, define running products:

$$\begin{aligned} \widetilde{w}_1 &\coloneqq \alpha + \beta + w_1, & w'_1 &\coloneqq \alpha + \beta \cdot \varphi(1) + w_1, \\ \widetilde{w}_i &\coloneqq \widetilde{w}_{i-1} \cdot (\alpha + \beta \cdot i + w_i) & w'_i &\coloneqq w'_{i-1} \cdot (\alpha + \beta \cdot \varphi(i) + w_i) \end{aligned}$$

for all $i \in [2, n_{\mathsf{w}}]$. The grand product holds if and only if $\widetilde{w}_{n_{\mathsf{w}}} = w'_{n_{\mathsf{w}}}$. System (7) is therefore equivalent to the following purely quadratic system:

$$\begin{cases} \widehat{w}_i - w_{n_{\mathsf{in}}+i} \cdot w_{n_{\mathsf{in}}+n_{\mathsf{gate}}+i} = 0 & \forall i \in [n_{\mathsf{gate}}], \\ s_i^{\mathsf{lf}} \cdot w_{n_{\mathsf{in}}+i} + s_i^{\mathsf{rg}} \cdot w_{n_{\mathsf{in}}+n_{\mathsf{gate}}+i} + s_i^{\mathsf{mlt}} \cdot \widehat{w}_i + s_i^{\mathsf{cnst}} - w_{n_{\mathsf{in}}+2n_{\mathsf{gate}}+i} = 0 & \forall i \in [n_{\mathsf{gate}}], \\ \alpha + \beta + w_1 - \widetilde{w}_1 = 0, \alpha + \beta \cdot \varphi(1) + w_1 - w'_1 = 0, \widetilde{w}_{n_{\mathsf{w}}} = w'_{n_{\mathsf{w}}}, & \\ \widetilde{w}_{i-1} \cdot (\alpha + \beta \cdot i + w_i) - \widetilde{w}_i = 0 & \forall i \in [2, n_{\mathsf{w}}], \\ w'_{i-1} \cdot (\alpha + \beta \cdot \varphi(i) + w_i) - w'_i = 0 & \forall i \in [2, n_{\mathsf{w}}]. \end{cases} \tag{8}$$

System (8) has the same structure for any arithmetic circuit of the same gate count, independent of the specific circuit. This uniformity is what enables hidden-circuit satisfiability proofs: the constraints do not expose the circuit's topology.

All witnesses supporting verification of (8) are collected in:

$$\mathsf{wit} = \big(\mathsf{pd} \parallel \mathbf{x} \parallel \mathbf{a} \parallel \mathbf{b} \parallel \mathbf{c} \parallel (\widehat{w}_i)_{i=1}^{n_{\mathsf{gate}}} \parallel (\widetilde{w}_i)_{i=1}^{n_{\mathsf{w}}} \parallel (w'_i)_{i=1}^{n_{\mathsf{w}}}\big). \tag{9}$$

This vector is fully determined by α, β, and $(w_i)_{i=1}^{n_{\mathsf{w}}}$.

Two-phase witness structure. The vector wit splits into two parts: $\mathsf{wit} = (\mathsf{wit}_{\mathsf{pre}} \parallel \mathsf{wit}_{\mathsf{suf}})$, where

$$\mathsf{wit}_{\mathsf{pre}} = \big(\mathsf{pd} \parallel \mathbf{x} \parallel \mathbf{a} \parallel \mathbf{b} \parallel \mathbf{c} \parallel (\widehat{w}_i)_{i=1}^{n_{\mathsf{gate}}}\big) \quad \text{is determined without knowing } \alpha, \beta;$$
$$\mathsf{wit}_{\mathsf{suf}} = \big((\widetilde{w}_i)_{i=1}^{n_{\mathsf{w}}} \parallel (w'_i)_{i=1}^{n_{\mathsf{w}}}\big) \quad \text{is determined only after } \alpha, \beta \text{ are known.}$$

This split is critical for the ZKBoo-based instantiation of Section 5: the prover first commits to $\mathsf{wit}_{\mathsf{pre}}$, derives α, β from a random oracle, then computes and commits to $\mathsf{wit}_{\mathsf{suf}}$.

In summary, any hidden arithmetic circuit F can now be verified via a fixed public system of quadratic constraints of size $\mathcal{O}(n_{\mathsf{in}} + n_{\mathsf{add}} + n_{\mathsf{mlt}})$, with the circuit's structure encoded entirely in the witness wit.

Sizes. Since $n_{\mathsf{w}} = \mathcal{O}(n_{\mathsf{in}} + n_{\mathsf{gate}}) = \mathcal{O}(n_{\mathsf{in}} + n_{\mathsf{add}} + n_{\mathsf{mlt}})$, the witness vector wit has $\mathcal{O}(n_{\mathsf{in}} + n_{\mathsf{add}} + n_{\mathsf{mlt}})$ field elements, and system (8) has $\mathcal{O}(n_{\mathsf{in}} + n_{\mathsf{add}} + n_{\mathsf{mlt}})$ multiplications and additions over $\mathbb{F}$.

4.2 Proving SMS Function Evaluations

This subsection applies the PLONK framework of Section 4.1 to the SMS evaluation structure, producing a single merged circuit $\mathsf{C}_{\mathsf{merge}}$ whose satisfiability the NIZKAoK in Sign proves.

Proving the hidden evaluation of F. Let F be the signer's hidden policy with $n_{\mathsf{in}}^{(F)}$ inputs, $n_{\mathsf{gate}}^{(F)}$ gates, and $n_{\mathsf{w}}^{(F)} := n_{\mathsf{in}}^{(F)} + 3n_{\mathsf{gate}}^{(F)}$. The signer evaluates F on $\mathbf{x} = (M \| \mathsf{attr} \| w)$ and retains the full computation trace. Let pd_F denote F's PLONK description of the form (5). The signer then samples $\alpha, \beta \xleftarrow{\$} \mathbb{F}$ from a random oracle and assembles the witness vector:

$$\mathsf{wit}_F = \big(\mathsf{pd}_F \parallel \mathbf{x} \parallel \mathbf{a} \parallel \mathbf{b} \parallel \mathbf{c} \parallel (\widehat{w}_i)_{i=1}^{n_{\mathsf{gate}}^{(F)}} \parallel (\widetilde{w}_i)_{i=1}^{n_{\mathsf{w}}^{(F)}} \parallel (w'_i)_{i=1}^{n_{\mathsf{w}}^{(F)}}\big), \tag{10}$$

where $\mathbf{x} = (M \| \mathsf{attr} \| w)$ and all remaining components follow the definitions of Section 4.1. Viewing system (8) instantiated with pd_F as a single verification circuit $\mathcal{U}_F^{\mathsf{plonk}}$, the signer must prove that wit_F satisfies $\mathcal{U}_F^{\mathsf{plonk}}$ with public output out_F. Since $\mathcal{U}_F^{\mathsf{plonk}}$ has $\mathcal{O}(n_{\mathsf{in}}^{(F)} + n_{\mathsf{gate}}^{(F)})$ multiplications, this proof is as efficient as proving the satisfiability of a public circuit of the same size.

For the NIZKAoK to be sound, α and β must be derived from a random oracle after the prover commits to $\mathsf{wit}_{\mathsf{pre}}$; allowing the prover to choose α, β

in advance would permit a cheating prover to satisfy the grand product trivially. This commitment-then-challenge structure is precisely the two-phase split $(\mathsf{wit}_{\mathsf{pre}}, \mathsf{wit}_{\mathsf{suf}})$ of Section 4.1, and it drives the modification to ZKBoo in Section 5.

Treating G and P as public circuits. Since G and P are public, PLONK's arithmetization is not needed for them. They are treated directly as arithmetic circuits C_G and C_P, and their evaluations are proved as standard public-circuit satisfiability.

Merging into a single circuit. Given α, β, wit_F, M, attr, w, and intermediate outputs out_F and out_G, we merge $\mathcal{U}_F^{\mathsf{plonk}}$, C_G, and C_P into a single circuit $\mathsf{C}_{\mathsf{merge}}$ capturing: α, β, wit_F, and out_F satisfy $\mathcal{U}_F^{\mathsf{plonk}}$; and $G(M, \mathsf{attr}, w) = \mathsf{out}_G$ and $P(\mathsf{out}_F, \mathsf{out}_G) = (\overline{M}, \mathsf{op})$. A single NIZKAoK applied to $\mathsf{C}_{\mathsf{merge}}$ then certifies the full evaluation $P(F(M, \mathsf{attr}, w), G(M, \mathsf{attr}, w))$ with public inputs α, β, $(\overline{M}, \mathsf{op})$ and private witnesses M, attr, w, wit_F, out_F, out_G, without leaking F's circuit structure or computation trace.

After the Fiat–Shamir transform, α and β are derived from the random oracle and are publicly recomputable by the verifier; the remaining witnesses M, attr, w, wit_F, out_F, out_G are known only to the signer and hidden by the NIZKAoK.

Sizes. Let $n_{\mathsf{add}}^{(T)}$ and $n_{\mathsf{mlt}}^{(T)}$ denote the addition and multiplication gate counts of function $T \in \{F, G, P\}$. Circuit $\mathsf{C}_{\mathsf{merge}}$ has:

$$\begin{aligned} \text{additions:} \quad & \mathcal{O}\big(n_{\mathsf{in}}^{(F)} + n_{\mathsf{add}}^{(F)} + n_{\mathsf{mlt}}^{(F)}\big) + n_{\mathsf{add}}^{(G)} + n_{\mathsf{add}}^{(P)}; \\ \text{multiplications:} \quad & \mathcal{O}\big(n_{\mathsf{in}}^{(F)} + n_{\mathsf{add}}^{(F)} + n_{\mathsf{mlt}}^{(F)}\big) + n_{\mathsf{mlt}}^{(G)} + n_{\mathsf{mlt}}^{(P)}. \end{aligned}$$

For NIZKAoK systems whose proof size depends on the multiplication count, the dominant cost is $\mathcal{O}(n_{\mathsf{in}}^{(F)} + n_{\mathsf{add}}^{(F)} + n_{\mathsf{mlt}}^{(F)}) + n_{\mathsf{mlt}}^{(G)} + n_{\mathsf{mlt}}^{(P)}$, where $n_{\mathsf{in}}^{(F)} = |M| + |\mathsf{attr}| + |w|$, so the dominant multiplication count is $\mathcal{O}(|M| + |\mathsf{attr}| + |w| + n_{\mathsf{add}}^{(F)} + n_{\mathsf{mlt}}^{(F)}) + n_{\mathsf{mlt}}^{(G)} + n_{\mathsf{mlt}}^{(P)}$, which is quasilinear in the gate count of F and linear in the multiplications of G and P. This confirms the proof-size claim of Section 1.1. The ZKBoo-based instantiation exploits the two-phase split $(\mathsf{wit}_{\mathsf{pre}}, \mathsf{wit}_{\mathsf{suf}})$ from Section 4.1 to handle the random challenges α, β; details are in Section 5.

5 Lattice-Based Instantiation of SMS

This section instantiates the generic SMS construction of Section 3 in the post-quantum setting, combining lattice-based primitives over $\mathbb{Z}_q$ with ZKBoo [35], a zero-knowledge argument of knowledge following MPC-in-the-Head [39]. Two design choices drive the instantiation. First, ZKBoo's simulation-sound extractability with quasi-unique response [29] subsumes the anti-malleability role of the one-time signature $\mathcal{OTS}$ in the generic construction, simplifying the lattice scheme. Second, proving the hidden evaluation of F via PLONK's arithmetization (Section 4) requires random challenges α, β derived after committing to $\mathsf{wit}_{\mathsf{pre}}$; applying ZKBoo directly would allow a cheating prover to choose α, β in advance

and satisfy the grand product check trivially. We therefore modify ZKBoo to interleave the pre-challenge and post-challenge commitment phases.

Section 5.1 describes the instantiation and the relation $\mathrm{R}_{\mathsf{l\text{-}SMS}}$. Section 5.2 presents the modified ZKBoo protocol. Section 5.3 analyzes the signature size. Full details of all primitives, proofs, and parameter settings are in the full version.

5.1 Instantiation Overview

Primitive choices. The generic construction is instantiated with LP PKE [50] ($\mathcal{E}_{\mathsf{lp}}$) as $\mathcal{E}$, whose simple structure based on the learning-with-errors assumption is compatible with ZKBoo's arithmetic circuit proofs; JRS signatures [40] ($\mathcal{S}_{\mathsf{jrs}}$) as $\mathcal{S}$, whose lattice-based signing keys support efficient zero-knowledge membership proofs; and ZKBoo [35] as the NIZKAoK, with all computations over $\mathbb{Z}_q$ for a prime q. KTX commitments [43] replace the per-user key pair.

OTS elimination. Because ZKBoo satisfies simulation-sound extractability with quasi-unique response [29], it subsumes the anti-malleability role of the one-time signature $\mathcal{OTS}$ required in the generic construction: any adversary modifying a ZKBoo proof cannot produce a second valid proof for a different statement without breaking the quasi-unique response property. Retaining $\mathcal{OTS}$ in the lattice setting would require a fresh key pair and an additional signature over all components per signing, increasing both signature size and signing cost; ZKBoo's quasi-unique response gives the same guarantee at no extra cost. Consequently, $(\mathsf{vk}_{\mathsf{attr}}, \mathsf{sk}_{\mathsf{attr}})$ no longer serves as a one-time signing key pair for ovk, but is still required for membership proofs.

Membership key via KTX. Following the KTX commitment paradigm [43], we enforce the relationship $\mathsf{vk}_{\mathsf{attr}} = \mathsf{bin}(\mathbf{H} \cdot \mathsf{sk}_{\mathsf{attr}})$, where $\mathbf{H} \in \mathbb{Z}_q^{n \times m}$ is public and random, $\mathsf{bin}(\cdot)$ denotes binary decomposition, and $\mathsf{sk}_{\mathsf{attr}}$ is a uniformly random binary vector chosen by the enrolling user. By the security of the KTX commitment, $\mathsf{vk}_{\mathsf{attr}}$ computationally hides $\mathsf{sk}_{\mathsf{attr}}$; by the hardness of the short integer solution problem, no adversary seeing $\mathsf{vk}_{\mathsf{attr}}$ can recover a binary preimage.

Relation $\mathrm{R}_{\mathsf{l\text{-}SMS}}$. The NIZKAoK proves membership in a relation $\mathrm{R}_{\mathsf{l\text{-}SMS}}$ containing two classes of constraints.

1. *Lattice-based constraints.* These enforce binary-vector structure, small ℓ_∞-norm bounds, and linear relationships over $\mathbb{Z}_q$. Binary-vector structure is captured by $\langle \mathbf{v}, \mathbf{1} - \mathbf{v} \rangle = \mathbf{0}$. Small-norm bounds are handled via the following binary decomposition lemma.

   Lemma 1 (Binary decomposition, [53], [38, Appendix A.1]). *Let $v \in \mathbb{N}$ and $k = \lfloor \log_2 v \rfloor + 1$. Define the basis $\mathcal{B}_v = \{B_i\}_{i=1}^{k} = \{2^i\}_{i=0}^{k-1} \cup \{v - (2^{k-1} - 1)\}$. Then $x \in [0, v]$ if and only if there exist bits $b_1, \ldots, b_k$ satisfying $x = \sum_{i=1}^{k} b_i B_i$.*

   Using Lemma 1, each bound $\|\mathbf{a}\|_\infty \leq u$ is reduced to binary-vector and linear constraints, compatible with ZKBoo's arithmetic circuit satisfiability. The proof of Lemma 1 is deferred to the full version.

2. *Circuit satisfiability constraints.* These enforce $P(F(\cdot), G(\cdot))$ and any auxiliary public circuits are correctly evaluated, including the circuit checking $\mathsf{op} \in \mathcal{OP}_{\mathsf{sgnbl}}$. The $P(\cdot,\cdot)$ part is handled by the PLONK reduction of Section 4.

The full relation $\mathrm{R}_{\mathsf{l\text{-}SMS}}$ is also deferred to the full version.

ZKBoo as the NIZKAoK. ZKBoo proves arithmetic circuit satisfiability and is plausibly post-quantum. Proving $\mathrm{R}_{\mathsf{l\text{-}SMS}}$ requires the hidden-circuit strategy of Section 4, which introduces additional challenges $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_q$ that must be generated from the random oracle after the prover commits to $\mathsf{wit}_{\mathsf{pre}}$. The Fiat-Shamir transform [31] makes the protocol non-interactive; however, the standard ZKBoo decomposition must be modified to accommodate the two-phase witness structure $(\mathsf{wit}_{\mathsf{pre}}, \mathsf{wit}_{\mathsf{suf}})$. Section 5.2 describes this modification. The result is a plausibly post-quantum SMS scheme enforcing hidden arithmetic policies with per-signature modal accountability.

5.2 Modifying the ZKBoo Protocol

In the original ZKBoo protocol [35], the computation trace of a circuit C, treated as a witness W, is decomposed via a randomized function $\mathsf{decompose}(W)$, into three views $\{\mathsf{view}_j\}_{j=1}^3$, such that any two views reveal no information of W, then committed by the prover as $\{\mathsf{cview}_j\}_{j=1}^3$. The verifier challenges the prover to open two of the three commitments; consistency of the two opened views certifies correct evaluation. See the full version for more details.

Modified decomposition. For proving $\mathrm{R}_{\mathsf{l\text{-}SMS}}$, the witness decomposes as W_{pre} and W_{suf} (the pre- and post-challenge components of Section 4.1): W_{pre} is determined before the challenges α, β are known, and W_{suf} is determined only after.

Specifically, we run $\mathsf{decompose}(W_{\mathsf{pre}})$ to obtain *pre-views* $\{\mathsf{preview}_j\}_{j=1}^3$ and treat each $\mathsf{preview}_j$ as the prefix of the full view: $\mathsf{view}_j = (\mathsf{preview}_j \| \mathsf{sufview}_j)$. The suffix views $\{\mathsf{sufview}_j\}_{j=1}^3$ are obtained by running $\mathsf{decompose}(\cdot)$ on the combined process of (i) computing all components of W_{suf} using entries of $\{\mathsf{preview}_j\}_{j=1}^3$ and (ii) checking system (8). The resulting full views $\{\mathsf{view}_j\}_{j=1}^3$ are thus a valid ZKBoo decomposition of the extended witness $(W_{\mathsf{pre}} \| W_{\mathsf{suf}})$.

Modified protocol. The modified protocol proceeds in three phases.

1. The prover commits to $\{\mathsf{preview}_j\}_{j=1}^3$, sending commitments $\{\mathsf{cpreview}_j\}_{j=1}^3$ to the verifier.
2. The verifier sends $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_q$ (derived from the random oracle in the non-interactive version). The prover computes $\{\mathsf{sufview}_j\}_{j=1}^3$ and sends commitments $\{\mathsf{csufview}_j\}_{j=1}^3$.
3. The commitment to view_j is the pair $(\mathsf{cpreview}_j, \mathsf{csufview}_j)$. The prover and verifier execute the original ZKBoo consistency check on the requested views.

This structure ensures that α, β are unknown to the prover when it commits to $\mathsf{wit}_{\mathsf{pre}}$, preventing the cheating-prover attack identified in Section 4.2. See the full version for the full protocol description and security analysis.

5.3 Signature Size

The final SMS signature is $\Sigma_{\mathsf{l\text{-}SMS}} = (\mathsf{ct}_{\mathsf{op}}, \Pi_{\mathsf{l\text{-}SMS}})$, where $\mathsf{ct}_{\mathsf{op}}$ encrypts op and $\Pi_{\mathsf{l\text{-}SMS}}$ is the non-interactive version of the modified ZKBoo for $\mathrm{R}_{\mathsf{l\text{-}SMS}}$. Thus $|\Sigma_{\mathsf{l\text{-}SMS}}| = |\mathsf{ct}_{\mathsf{op}}| + |\Pi_{\mathsf{l\text{-}SMS}}|$. Since $|\Pi_{\mathsf{l\text{-}SMS}}|$ dominates $|\mathsf{ct}_{\mathsf{op}}|$, we focus on the former. The full lattice parameter settings, including $|\mathsf{ct}_{\mathsf{op}}|$, are deferred to the full version.

Bounding $|\Pi_{\mathsf{l\text{-}SMS}}|$. The proof $\Pi_{\mathsf{l\text{-}SMS}}$ consists of κ repetitions of the modified ZKBoo, with a soundness error $2/3$ of each repetition, where $\kappa = \lceil \lambda \cdot (\log 3 - 1)^{-1} \rceil$ achieves soundness error $2^{-\lambda}$. Each repetition contributes commitments $\mathsf{cpreview}_j$ and $\mathsf{csufview}_j$ (compressed via hashing) and a response rsp_j. Thus $|\Pi_{\mathsf{l\text{-}SMS}}| = \mathcal{O}(\kappa \cdot (|\mathsf{cpreview}| + |\mathsf{csufview}| + |\mathsf{rsp}|))$. The dominant term is $|\mathsf{rsp}|$. Addition-gate outputs in the views can be recomputed for free [17,42], so only multiplication-gate outputs need to be transmitted.

Response size. The $\log q$ factor in the response arises because all lattice-based primitives operate over binary representations: every $\mathbb{Z}_q$ field element must be binary-decomposed before lattice arithmetic applies, multiplying each element's contribution to $|\mathsf{rsp}|$ by $\log q$. With this decomposition, $|\mathsf{rsp}|$ is bounded by

$$\mathcal{O}\Big(\log q \cdot \big(|\mathsf{lparams}| + |M| + |\mathsf{attr}| + |w| + n_{\mathsf{add}}^{(F)} + n_{\mathsf{mlt}}^{(F)} + |\mathsf{op}| \cdot \log q\big) + |\mathsf{lparams}'| + n_{\mathsf{mlt}}^{(G)} + n_{\mathsf{mlt}}^{(P)} + n_{\mathsf{mlt}}^{(\mathsf{op})} \Big) \tag{11}$$

$\mathbb{Z}_q$-elements, where $|\mathsf{lparams}|$ and $|\mathsf{lparams}'|$ denote the parameter sizes of the PKE and signature primitives, instantiated as LP and JRS respectively; their exact values depend on choices deferred to the full version, while $n_{\mathsf{mlt}}^{(\mathsf{op})}$ counts the multiplication for the circuit checking of $\mathsf{op} \in \mathcal{OP}_{\mathsf{sgnbl}}$.

Asymptotic summary. From (11), $|\Pi_{\mathsf{l\text{-}SMS}}|$ is quasilinear in the total gate count of F (due to the $\log q$ factor from binary decomposition) and linear in the multiplication counts of G and P. This matches the proof-size claim of Section 1.1 and confirms that the $\log q$ overhead is an unavoidable cost of working with lattice-based primitives, not an artifact of the construction. This framework admits richer instantiations via advanced MPCitH [17,42,7,62,30,16,3,4] and VOLEitH techniques [6,27,60,13].

6 Conclusion

Sovereign modal signatures (SMS) unify hidden and public policy enforcement with per-signature modal accountability, addressing a decade-old open problem in privacy-preserving signatures by treating a circuit's PLONK description as a witness rather than a statement, achieving the first efficient hidden arithmetic-circuit evaluation without universal circuits or idealized algebraic models.

Multi-stakeholder deployments, *e.g.*, audit systems, regulatory compliance, and credential delegation, are the natural home for SMS: each participant operates under a certifying authority's hidden policy, overlays a publicly verifiable

check at signing time, and exposes exactly as much identity as the configuration prescribes, realizing the functional credentials vision of prior work [24].

Concrete directions for future work include threshold and distributed issuance, revocation within the partial-dynamic model, and measured efficiency at the 128-bit post-quantum security level; more broadly, the PLONK-based hidden-circuit reduction developed here is a general tool applicable to any privacy primitive using hidden arithmetic-circuit proofs in the random oracle model.

Acknowledgments

The first four authors utilized AI tools for linguistic and stylistic refinement.

Research at Nanyang Technological University is supported by Singapore Ministry of Education Academic Research Fund Tier 2 Grant MOE-T2EP20223-0016, the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

Yingfei Yan is supported by the European Astral Project ERASMUS-EDU 2024:101184483. She thanks Xidian University for its support and Nanyang Technological University (NTU) and The Chinese University of Hong Kong (CUHK) for their hospitality during her visits.

Sherman Chow is supported in part by the Research Grants Council of Hong Kong under the Collaborative Research Fund (C5097-25GF) and the General Research Fund (14210825).

Kai Zhang is supported in part by the Natural Science Foundation of Shaanxi Province (No. 2025JC-YBMS-748).

References

1. Abe, M., Chow, S.S.M., Haralambiev, K., Ohkubo, M.: Double-trapdoor anonymous tags for traceable signatures. International Journal of Information Security **12**(1), 19–31 (2013). https://doi.org/10.1007/s10207-012-0184-3
2. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: CRYPTO. LNCS, vol. 6223, pp. 209–236 (2010). https://doi.org/10.1007/978-3-642-14623-7_12
3. Aguilar-Melchor, C., Gama, N., Howe, J., Hülsing, A., Joseph, D., Yue, D.: The return of the SDitH. In: EUROCRYPT Part V. LNCS, vol. 14008, pp. 564–596 (2023). https://doi.org/10.1007/978-3-031-30589-4_20
4. Aguilar-Melchor, C., Hülsing, A., Joseph, D., Majenz, C., Ronen, E., Yue, D.: SDitH in the QROM. In: ASIACRYPT Part VII. LNCS, vol. 14444, pp. 317–350 (2023). https://doi.org/10.1007/978-981-99-8739-9_11
5. Au, M.H., Susilo, W., Mu, Y., Chow, S.S.M.: Constant-size dynamic k-times anonymous authentication. IEEE Syst. J. **7**(2), 249–261 (2013). https://doi.org/10.1109/JSYST.2012.2221931

6. Baum, C., Braun, L., de Saint Guilhem, C.D., Klooß, M., Orsini, E., Roy, L., Scholl, P.: Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head. In: CRYPTO Part V. LNCS, vol. 14085, pp. 581–615 (2023). https://doi.org/10.1007/978-3-031-38554-4_19
7. Baum, C., Nof, A.: Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In: PKC Part I. LNCS, vol. 12110, pp. 495–526 (2020). https://doi.org/10.1007/978-3-030-45374-9_17
8. Bellare, M., Fuchsbauer, G.: Policy-based signatures. In: PKC. LNCS, vol. 8383, pp. 520–537 (2014). https://doi.org/10.1007/978-3-642-54631-0_30
9. Bender, A., Katz, J., Morselli, R.: Ring signatures: stronger definitions, and constructions without random oracles. In: TCC. LNCS, vol. 3876, pp. 60–79 (2006). https://doi.org/10.1007/11681878_4
10. Bobolz, J., Diaz, J., Kohlweiss, M.: Foundations of anonymous signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: FC. LNCS, vol. 14745, pp. 121–139 (2025). https://doi.org/10.1007/978-3-031-78679-2_7
11. Boneh, D., Nguyen, W., Ozdemir, A.: Efficient functional commitments: how to commit to a private function. Cryptology ePrint Archive, Paper 2021/1342 (2021), https://ia.cr/2021/1342
12. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: PKC. LNCS, vol. 8383, pp. 501–519 (2014). https://doi.org/10.1007/978-3-642-54631-0_29
13. Bui, D.: Efficient multi-instance vector commitment and application to post-quantum signatures. In: ACISP Part II. LNCS, vol. 15659, pp. 23–41 (2025). https://doi.org/10.1007/978-981-96-9098-5_2
14. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: EUROCRYPT. LNCS, vol. 2045, pp. 93–118 (2001). https://doi.org/10.1007/3-540-44987-6_7
15. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: CRYPTO. LNCS, vol. 3152, pp. 56–72 (2004). https://doi.org/10.1007/978-3-540-28628-8_4
16. Carozza, E., Couteau, G., Joux, A.: Short signatures from regular syndrome decoding in the head. In: EUROCRYPT Part V. LNCS, vol. 14008, pp. 532–563 (2023). https://doi.org/10.1007/978-3-031-30589-4_19
17. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Zaverucha, G.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: CCS. pp. 1825–1842 (2017). https://doi.org/10.1145/3133956.3133997
18. Chase, M., Lysyanskaya, A.: On signatures of knowledge. In: CRYPTO. LNCS, vol. 4117, pp. 78–96 (2006). https://doi.org/10.1007/11818175_5
19. Chaum, D., van Heyst, E.: Group signatures. In: EUROCRYPT. LNCS, vol. 547, pp. 257–265 (1991). https://doi.org/10.1007/3-540-46416-6_22
20. Cheng, S., Nguyen, K., Wang, H.: Policy-based signature scheme from lattices. Des. Codes Cryptogr. **81**(1), 43–74 (2016). https://doi.org/10.1007/S10623-015-0126-Y
21. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, N., Ward, N.: Marlin: preprocessing zkSNARKs with universal and updatable SRS. In: EUROCRYPT Part I. LNCS, vol. 12105, pp. 738–768 (2020). https://doi.org/10.1007/978-3-030-45721-1_26

22. Choudhuri, A.R., Garg, S., Goel, A., Sekar, S., Sinha, R.: SublonK: sublinear prover PlonK. In: PETS. vol. 2024, pp. 314–335 (2024). https://doi.org/10.56553/POPETS-2024-0080
23. Chow, S.S.M.: Real traceable signatures. In: SAC. pp. 92–107. LNCS (2009). https://doi.org/10.1007/978-3-642-05445-7_6
24. Chow, S.S.M.: Functional credentials for Internet of Things. In: IoTPTS. p. 1 (2016). https://doi.org/10.1145/2899007.2899014
25. Chow, S.S.M., Susilo, W., Yuen, T.H.: Escrowed linkability of ring signatures and its applications. In: VIETCRYPT. pp. 175–192. LNCS (2006). https://doi.org/10.1007/11958239_12
26. Chow, S.S.M., Zhang, H., Zhang, T.: Real hidden identity-based signatures. In: FC. pp. 21–38. LNCS (2017). https://doi.org/10.1007/978-3-319-70972-7_2
27. Cui, H., Liu, H., Yan, D., Yang, K., Yu, Y., Zhang, K.: ReSolveD: shorter signatures from regular syndrome decoding and VOLE-in-the-head. In: PKC Part I. LNCS, vol. 14601, pp. 229–258 (2024). https://doi.org/10.1007/978-3-031-57718-5_8
28. Di, Z., Xia, L., Nguyen, W., Tyagi, N.: MuxProofs: succinct arguments for machine computation from vector lookups. In: ASIACRYPT Part V. LNCS, vol. 15488, pp. 236–265 (2025). https://doi.org/10.1007/978-981-96-0935-2_8
29. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the Fiat-Shamir transform. In: INDOCRYPT. LNCS, vol. 7668, pp. 60–79 (2012). https://doi.org/10.1007/978-3-642-34931-7_5
30. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: shorter signatures from zero-knowledge proofs. In: CRYPTO Part II. LNCS, vol. 13508, pp. 541–572 (2022). https://doi.org/10.1007/978-3-031-15979-4_19
31. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: CRYPTO. LNCS, vol. 263, pp. 186–194 (1987). https://doi.org/10.1007/3-540-47721-7_12
32. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: CRYPTO Part II. LNCS, vol. 10992, pp. 33–62 (2018). https://doi.org/10.1007/978-3-319-96881-0_2
33. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: PKC. LNCS, vol. 4450, pp. 181–200 (2007). https://doi.org/10.1007/978-3-540-71677-8_13
34. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Paper 2019/953 (2019), https://ia.cr/2019/953
35. Giacomelli, I., Madsen, J., Orlandi, C.: ZKBoo: faster zero-knowledge for boolean circuits. In: USENIX Security. pp. 1069–1083 (2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/giacomelli
36. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: EUROCRYPT. LNCS, vol. 4965, pp. 415–432 (2008). https://doi.org/10.1007/978-3-540-78967-3_24
37. Guo, Q., Huang, Q., Ma, S., Xiao, M., Yang, G., Susilo, W.: Functional signatures: new definition and constructions. Sci. China Inf. Sci. **64**(12) (2021). https://doi.org/10.1007/S11432-019-2855-3
38. He, Y., Ling, S., Tang, K.H., Wang, H.: Everlasting fully dynamic group signatures. In: ACNS Part III. LNCS, vol. 15827, pp. 3–28 (2025). https://doi.org/10.1007/978-3-031-95767-3_1, ePrint Archive https://ia.cr/2025/627
39. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: STOC. pp. 21–30 (2007). https://doi.org/10.1145/1250790.1250794

40. Jeudy, C., Roux-Langlois, A., Sanders, O.: Lattice signature with efficient protocols, application to anonymous credentials. In: CRYPTO Part II. LNCS, vol. 14082, pp. 351–383 (2023). https://doi.org/10.1007/978-3-031-38545-2_12
41. Karatsuba, A.A., Ofman, Y.P.: Multiplication of many-digital numbers by automatic computers. In: Doklady Akademii Nauk. vol. 145, pp. 293–294 (1962)
42. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: CCS. pp. 525–537 (2018). https://doi.org/10.1145/3243734.3243805
43. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: ASIACRYPT. LNCS, vol. 5350, pp. 372–389 (2008). https://doi.org/10.1007/978-3-540-89255-7_23
44. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: EUROCRYPT. pp. 571–589. LNCS (2004). https://doi.org/10.1007/978-3-540-24676-3_34
45. Kiayias, A., Yung, M.: Secure scalable group signature with dynamic joins and separable authorities. Int. J. Secur. Networks **1**(1/2), 24–45 (2006). https://doi.org/10.1504/IJSN.2006.010821
46. Kiayias, A., Zhou, H.S.: Hidden identity-based signatures. In: FC. LNCS, vol. 4886, pp. 134–147 (2007). https://doi.org/10.1007/978-3-540-77366-5_14
47. Kiss, Á., Schneider, T.: Valiant's universal circuit is practical. In: EUROCRYPT Part III. LNCS, vol. 9665, pp. 699–728 (2016). https://doi.org/10.1007/978-3-662-49890-3_27
48. Lai, R.W.F., Zhang, T., Chow, S.S.M., Schröder, D.: Efficient sanitizable signatures without random oracles. In: ESORICS Part I. LNCS, vol. 9878, pp. 363–380 (2016). https://doi.org/10.1007/978-3-319-45744-4_18
49. Libert, B., Nguyen, K., Peters, T., Yung, M.: Bifurcated signatures: folding the accountability vs. anonymity dilemma into a single private signing scheme. In: EUROCRYPT Part III. LNCS, vol. 12698, pp. 521–552 (2021). https://doi.org/10.1007/978-3-030-77883-5_18
50. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: CT-RSA. LNCS, vol. 6558, pp. 319–339 (2011). https://doi.org/10.1007/978-3-642-19074-2_21
51. Ling, S., Nguyen, K., Phan, D.H., Tang, H., Wang, H.: Zero-knowledge proofs for committed symmetric boolean functions. In: PQCrypto. LNCS, vol. 12841, pp. 339–359 (2021). https://doi.org/10.1007/978-3-030-81293-5_18
52. Ling, S., Nguyen, K., Phan, D.H., Tang, K.H., Wang, H., Xu, Y.: Fully dynamic attribute-based signatures for circuits from codes. In: PKC Part I. LNCS, vol. 14601, pp. 37–73 (2024). https://doi.org/10.1007/978-3-031-57718-5_2
53. Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: PKC. LNCS, vol. 7778, pp. 107–124 (2013). https://doi.org/10.1007/978-3-642-36362-7_8
54. Lipmaa, H., Mohassel, P., Sadeghian, S.: Valiant's universal circuit: improvements, implementation, and applications. Cryptology ePrint Archive, Paper 2016/017 (2016), https://ia.cr/2016/017
55. Liu, H., Yu, Y., Zhao, S., Zhang, J., Liu, W., Hu, Z.: Pushing the limits of Valiant's universal circuits: simpler, tighter and more compact. In: CRYPTO Part II. LNCS, vol. 12826, pp. 365–394 (2021). https://doi.org/10.1007/978-3-030-84245-1_13
56. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: ACISP. pp. 325–335. LNCS (2004). https://doi.org/10.1007/978-3-540-27800-9_28

57. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. In: CT-RSA. LNCS, vol. 6558, pp. 376–392 (2011). https://doi.org/10.1007/978-3-642-19074-2_24
58. Nguyen, K., Guo, F., Susilo, W., Yang, G.: Multimodal private signatures. In: CRYPTO Part II. LNCS, vol. 13508, pp. 792–822 (2022). https://doi.org/10.1007/978-3-031-15979-4_27
59. Nguyen, K., Roy, P.S., Susilo, W., Xu, Y.: Bicameral and auditably private signatures. In: ASIACRYPT Part II. LNCS, vol. 14439, pp. 313–347 (2023). https://doi.org/10.1007/978-981-99-8724-5_10
60. Ouyang, Y., Tang, D., Xu, Y.: Code-based zero-knowledge from VOLE-in-the-head and their applications: simpler, faster, and smaller. In: ASIACRYPT Part V. LNCS, vol. 15488, pp. 436–470 (2025). https://doi.org/10.1007/978-981-96-0935-2_14
61. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: ASIACRYPT. LNCS, vol. 2248, pp. 552–565 (2001). https://doi.org/10.1007/3-540-45682-1_32
62. Delpech de Saint Guilhem, C., Orsini, E., Tanguy, T.: Limbo: efficient zero-knowledge MPCitH-based arguments. In: CCS. pp. 3022–3036 (2021). https://doi.org/10.1145/3460120.3484595
63. Sakai, Y.: Succinct attribute-based signatures for bounded-size circuits by combining algebraic and arithmetic proofs. In: SCN. LNCS, vol. 13409, pp. 711–734 (2022). https://doi.org/10.1007/978-3-031-14791-3_31
64. Tang, K.H., Pham, N.M., Ngo, C.N.: RAMenPaSTA: parallelizable scalable transparent arguments of knowledge for RAM programs. Cryptology ePrint Archive, Paper 2024/336 (2024), https://ia.cr/2024/336
65. Tran, N., Nguyen, K., Liu, D., Pieprzyk, J., Susilo, W.: Improved multimodal private signatures from lattices. In: ACISP. LNCS, vol. 14896, pp. 3–23 (2024). https://doi.org/10.1007/978-981-97-5028-3_1
66. Valiant, L.G.: Universal circuits (preliminary report). In: STOC. pp. 196–203 (1976). https://doi.org/10.1145/800113.803649
67. Xu, S., Yung, M.: Accountable ring signatures: a smart card approach. In: CARDIS. IFIP, vol. 153, pp. 271–286 (2004)
68. Xu, Y., Safavi-Naini, R., Nguyen, K., Wang, H.: Traceable policy-based signatures and instantiation from lattices. Inf. Sci. **607**, 1286–1310 (2022). https://doi.org/10.1016/j.ins.2022.06.031
69. Yan, Y., Tang, K.H., Chu, H., Chow, S.S.M., Ling, S., Wang, H., Zhang, K.: Sovereign modal signatures. Cryptology ePrint Archive (2026), full version
70. Zhang, C., Zhou, H.S., Katz, J.: An analysis of the algebraic group model. In: ASIACRYPT Part IV. LNCS, vol. 13794, pp. 310–322 (2022). https://doi.org/10.1007/978-3-031-22972-5_11
71. Zhang, T., Wu, H., Chow, S.S.M.: Structure-preserving certificateless encryption and its application. In: CT-RSA. pp. 1–22. LNCS (2019). https://doi.org/10.1007/978-3-030-12612-4_1
72. Zhao, S., Yu, Y., Zhang, J., Liu, H.: Valiant's universal circuits revisited: an overall improvement and a lower bound. In: ASIACRYPT Part I. LNCS, vol. 11921, pp. 401–425 (2019). https://doi.org/10.1007/978-3-030-34578-5_15