An Efficient SM9-Compatible Identity-Based Blind Signature Scheme for One-Time Issuance Identities

Zhiwei Wang

Nanjing University of Posts and Telecommunications, Nanjing 210023, China
zhwwang@njupt.edu.cn

Abstract. We present an identity-based blind signature scheme built on the SM9 Chinese national cryptographic standard. The scheme outputs standard SM9 signatures, removes all online pairing operations through a single setup-time pre-computation, and adds only one user-side exponentiation to achieve blindness. We prove two properties rigorously: *restricted target-identity unforgeability* (RTU) under the Q-BCAA1 assumption, and *computational blindness* against a malicious signer in the random oracle model (ROM), where the signer is given the final signatures and still cannot link them to its protocol views. We state clearly that RTU is strictly weaker than the standard one-more unforgeability (OMU) notion for blind signatures: RTU forbids signing queries on the target identity and is therefore meaningful only when that identity encodes a fresh, one-time issuance context, as occurs in coin-, ballot-, and credential-issuance protocols. The Q-BCAA1 assumption is the inversion-type assumption that matches the SM9 private-key structure $d_{\mathrm{ID}} = [s/(s+\tau)]P_1$; it lets the reduction answer adaptive key-extraction queries without any decisional (Gap) oracle. We give a partial-blind extension with a formal partial-blindness model, discuss implementation requirements, and compare costs against prior SM9-based proposals.
Keywords. Blind signature; SM9; Identity-based cryptography; Restricted unforgeability; Partial blindness; Q-BCAA1 assumption; Random oracle model.

1 Introduction

Blind signatures, introduced by Chaum [1], allow a signer to produce a valid signature on a message without learning its content. They are a building block of privacy-preserving systems such as electronic cash [2], anonymous credentials [26], and electronic voting [25]. The SM9 identity-based cryptographic standard [28], promulgated by the State Cryptography Administration of China and adopted as ISO/IEC 14888-3:2018, removes the certificate-management overhead of public-key infrastructures and is increasingly deployed in systems that must comply with Chinese national standards. Combining the privacy of blind signatures with the certificate-free deployment of SM9 is therefore a natural and practically relevant goal.

The SM9 private key has the form $d_{\mathrm{ID}} = [s/(s+\tau)]P_1$, where $\tau = H_1(\mathrm{ID}\|\mathsf{hid})$ and s is the master secret. This algebraic structure is the reason SM9 security analyses rely on *inversion-type* q-assumptions, such as q-BCAA1, rather than on the plain q-SDH assumption [29,30]. Any reduction that must answer adaptive key-extraction queries needs values of the form $[1/(s+\tau)]P_1$, which an inversion oracle provides directly and a plain q-SDH instance does not.

Contributions. This paper makes the following contributions. We give an SM9-compatible blind signing protocol (Algorithm 2) whose output is a standard SM9 signature, whose online cost contains no pairings, and which adds a single user-side blinding exponentiation. We formalize a security model, *restricted target-identity unforgeability* (Definition 2, Figure 1), and we state explicitly (Remark 2) that it is weaker than standard OMU and identify the deployments in which it is the appropriate guarantee. We prove RTU under the Q-BCAA1 assumption (Theorem 1) with a reduction that answers adaptive extraction queries through the inversion oracle and extracts the target key by the General Forking Lemma [9], and we prove computational blindness against a malicious signer in the ROM (Theorem 2), in a model that hands the signer the final signatures. We give a partial-blind extension (Algorithm 3) with a formal model (Definition 4, Figure 3) and a proof (Theorem 3), discuss implementation requirements (Section 9), and compare costs against prior SM9-based proposals (Tables 1–3).

Scope of the security claims. We prove a restricted unforgeability notion in which the adversary may not request blind signatures on the target identity ID^*. This is *not* the standard OMU notion [4,5], in which an adversary obtains ℓ signatures under ID^* and must output $\ell + 1$. Our notion is the special case $\ell = 0$. It is meaningful precisely when ID^* encodes a one-time issuance context that is never legitimately signed before a forgery is attempted; Section 4.3 describes how an issuer enforces this at the application layer and is explicit about what RTU does *not* cover. We do not claim full OMU, and we record it as an open problem in Section 10.

Organization. Section 2 surveys related work. Section 3 fixes notation, recalls SM9, and states the Q-BCAA1 assumption. Section 4 gives the security models and the one-time-issuance setting. Section 5 presents the construction. Section 6 contains the proofs. Section 7 reports performance. Section 8 gives the partial-blind extension with its formal model. Section 9 discusses implementation. Section 10 concludes.

2 Related Work

Blind signatures and their security model. Chaum [1] introduced blind signatures for untraceable payments, and Chaum, Fiat, and Naor [2] built the first untraceable e-cash system on top of them. Pointcheval and Stern [3,4] gave the first provably secure constructions and established the standard security model,

in which unforgeability is captured by the *one-more* notion: after ℓ completed signing interactions the adversary must not produce $\ell+1$ valid signatures. Juels, Luby, and Ostrovsky [5] gave a complexity-theoretic treatment of the same notion. Their proofs introduced the forking technique, later abstracted by Bellare and Neven [9] into the General Forking Lemma that we use. Hauck, Kiltz, and Loss [10] gave a modular treatment of blind signatures from identification schemes, and Abe [8] gave a three-move scheme supporting polynomially many signatures. The fragility of the one-more notion under concurrency was exposed by the ROS attacks [11], which break a family of Schnorr-type blind signatures once enough sessions are opened concurrently [12,13]; these results are part of the reason we treat full OMU with care rather than claiming it.

Partial and round-optimal blind signatures. Abe and Fujisaki [6] introduced partial blind signatures, in which a common public part of the message is fixed by agreement while the rest remains blinded; Abe and Okamoto [7] gave the first provably secure construction, and Chow et al. [24] gave pairing-based variants. A separate line of work targets round-optimal schemes: in the standard model from interactive or q-type assumptions [14,15], and in the ROM from lattices [19,16,17] and from pairing-free groups [18]. These constructions achieve full OMU but at a cost in signature size, assumptions, or rounds that is not aligned with producing a *standard* SM9 signature, which is our constraint.

Identity-based blind signatures. Zhang and Kim [22] proposed the first identity-based blind and ring signatures from pairings, and Cao et al. [23] improved efficiency. These early constructions use several online pairings and were analysed informally. Inversion-type assumptions such as those of Boneh and Boyen [20], and the gap-group techniques of Boldyreva [21], underlie much of the subsequent provable-security work on short and identity-based signatures. Identity-based blind signatures remain attractive for privacy systems with simple key management, for example verifiable e-voting [27].

SM9 and SM9-based blind signatures. Cheng [29] and Lai et al. [30] analysed the SM9 key agreement, signature, and encryption algorithms, establishing security under q-type and Gap-q-BCAA1 assumptions and clarifying why the SM9 key structure calls for inversion-type assumptions. Several SM9-based blind signature schemes have since been proposed [31,32,33,34]. Zhang and Peng [31] target e-commerce; Xu et al. [32] give a partial-blind variant; Li et al. [33] give a two-party cooperative scheme with a split key; and Zhang et al. [34] add fuzzy extractors and multi-factor authentication for data-privacy applications. These proposals generally provide informal security arguments, retain online pairings, or add components that increase cost. Table 3 summarizes the comparison. Our work differs in that it outputs standard SM9 signatures, eliminates online pairings, and provides explicit proofs under a clearly delimited model and a single inversion-type assumption.

3 Preliminaries

3.1 Notation and Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be cyclic groups of prime order p with generators $P_1 \in \mathbb{G}_1$, $P_2 \in \mathbb{G}_2$, and $g \in \mathbb{G}_T$. A bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is bilinear ($e([a]P_1, [b]P_2) = e(P_1, P_2)^{ab}$ for all $a, b \in \mathbb{Z}_p$), non-degenerate ($e(P_1, P_2) \neq 1_{\mathbb{G}_T}$), and efficiently computable. We use Type-3 pairings, with no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. Throughout, p denotes the group order, Q denotes a polynomial bound on the extraction queries answered by the assumption oracle, and $q_{H_1}, q_{H_2}, q_e, q_s$ denote bounds on the adversary's queries to H_1, H_2, the extraction oracle, and the signing oracle. We keep p and Q distinct to avoid conflating the group order with the parameter of the inversion-type assumption. We write $a \xleftarrow{\$} S$ for a uniform sample from a finite set S.

3.2 The Q-BCAA1 Assumption

Because the SM9 private key is $[s/(s+\tau)]P_1$, the matching computational assumption is the inversion-type Q-BCAA1 assumption [29,30], stated below in a key-extraction-oracle form convenient for the reduction.

Definition 1 (Q-BCAA1). *Let $s \xleftarrow{\$} \mathbb{Z}_p^*$ and $P_{\mathsf{pub}} = [s]P_2$. The adversary is given $(P_1, P_2, P_{\mathsf{pub}})$ and adaptive access to the oracle*

$$\mathcal{O}_s(\tau) \;=\; \left[\frac{1}{s+\tau}\right] P_1, \qquad \tau \in \mathbb{Z}_p,\ s+\tau \neq 0,$$

for at most Q distinct queries. The adversary wins if it outputs a fresh pair $\big(\tau^*, [1/(s+\tau^*)]P_1\big)$ with $\tau^* \notin \{\text{queried } \tau\}$. The Q-BCAA1 assumption states that $\mathbf{Adv}_{\mathcal{B}}^{Q\text{-BCAA1}}(\lambda) := \Pr[\mathcal{B} \text{ wins}] \leq \mathsf{negl}(\lambda)$ for all PPT $\mathcal{B}$.

Remark 1 (Why Q-BCAA1 and not q-SDH). A plain q-SDH instance $(P_1, [s]P_1, \ldots, [s^q]P_1, P_2, [s]P_2)$ does not directly provide values $[1/(s+\tau)]P_1$, and answering adaptive SM9 key-extraction queries from it requires an auxiliary polynomial simulation whose degree approaches p, which is infeasible for polynomially bounded q. The Q-BCAA1 assumption is the infeasibility-matched choice for SM9-style keys, and unlike Gap-Q-BCAA1 it needs no decisional oracle.

3.3 SM9 Identity-Based Signature

The SM9 signature scheme [28] is recalled in Algorithm 1; the hash functions are modelled as random oracles $H_1, H_2 : \{0,1\}^* \to \mathbb{Z}_p^*$ mapping onto nonzero scalars, following the standard encoding (an implementation that obtains 0 resamples or maps according to the specification).

Correctness follows from $e([k-h]d_{\mathrm{ID}}, [\tau]P_2 + P_{\mathsf{pub}}) = e([k-h]P_1, [s]P_2) = g^{k-h}$, so $w' = g^k = w$.

Algorithm 1 SM9 Identity-Based Signature

Setup:

KGC picks $s \xleftarrow{\$} \mathbb{Z}_p^*$, sets $P_{\mathsf{pub}} = [s]P_2$.
Publish $\mathsf{mpk} = (P_1, P_2, P_{\mathsf{pub}}, H_1, H_2)$ with $H_1, H_2 : \{0,1\}^* \to \mathbb{Z}_p^*$.

Extract$(\mathsf{msk}, \mathrm{ID})$:

$\tau \leftarrow H_1(\mathrm{ID} \| \mathsf{hid})$; **return** $d_{\mathrm{ID}} = [s/(s+\tau)]P_1$.

Sign(d_{ID}, M):

$k \xleftarrow{\$} \mathbb{Z}_p^*$; $w = e(P_1, P_{\mathsf{pub}})^k = g^k$.
$h = H_2(M \| w)$; $S = [k-h]d_{\mathrm{ID}}$; **return** $\sigma = (h, S)$.

Verify$(\mathrm{ID}, M, \sigma = (h, S))$:

$\tau \leftarrow H_1(\mathrm{ID} \| \mathsf{hid})$; $w' = g^h \cdot e(S, [\tau]P_2 + P_{\mathsf{pub}})$.
accept iff $h = H_2(M \| w')$.

4 Security Models

4.1 Syntax and the Signing Oracle

An identity-based blind signature scheme consists of Setup, $\mathsf{Extract}$, an interactive protocol $\langle \mathsf{User}(M), \mathsf{Signer}(d_{\mathrm{ID}}) \rangle$, and Verify, with the usual completeness requirement that an honest interaction yields a signature accepted by Verify.

In the unforgeability experiment the signing oracle is *interactive*. A query to $\mathcal{O}^{\mathsf{Sign}}(\mathrm{ID}, \cdot)$ starts a session of $\langle \mathsf{User}, \mathsf{Signer}(d_{\mathrm{ID}}) \rangle$ in which the challenger plays the honest signer holding d_{ID} and the adversary plays a possibly malicious user; the adversary controls scheduling and message contents, and sessions may be interleaved. A session is counted towards q_s once the signer sends its first protocol message; aborted sessions produce no signature but still count. This is weaker than full OMU only in the restriction we impose next, not in the power granted to the adversary within a session.

4.2 Restricted Target-Identity Unforgeability

Definition 2 (Restricted Target-Identity Unforgeability). *A scheme is* restricted target-identity unforgeable *(RTU) if for all PPT $\mathcal{A}$ the advantage* $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{RTU}}(\lambda) := \Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathrm{RTU}}(\lambda) = 1]$ *is negligible, where* $\mathsf{Exp}_{\mathcal{A}}^{\mathrm{RTU}}$ *is the experiment of Figure 1.*

Remark 2 (RTU is strictly weaker than OMU). The standard OMU model lets $\mathcal{A}$ run up to ℓ signing sessions under ID^* and requires $n > \ell$ forgeries. RTU is the case $\ell = 0$ with $n \geq 1$: it does not let the adversary obtain any signature under the target identity, so it does not capture the one-more property that is the defining security goal of blind signatures. In particular, RTU does *not* prevent a malicious user from turning one completed signing session under an identity into two valid signatures under that same identity; such over-issuance is exactly the one-more property and lies outside our model. RTU is the appropriate guarantee only when ID^* encodes a fresh, one-time issuance context that is never

Experiment $\mathsf{Exp}^{\mathrm{RTU}}_{\mathcal{A}}(\lambda)$:

1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $(\mathrm{ID}^*, \{(M_i, \sigma_i)\}_{i=1}^{n}) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{Ext}}, \mathcal{O}^{\mathsf{Sign}}, H_1, H_2}(\mathsf{mpk})$, where $\mathcal{O}^{\mathsf{Sign}}$ is the interactive oracle of Section 4.
3. If ID^* was queried to $\mathcal{O}^{\mathsf{Ext}}$: **return** 0.
4. If a session of $\mathcal{O}^{\mathsf{Sign}}(\mathrm{ID}^*, \cdot)$ was started: **return** 0. *(RTU restriction)*
5. If $n < 1$ or $\exists\, i \neq j : M_i = M_j$: **return** 0.
6. If $\exists\, i : \mathsf{Verify}(\mathsf{mpk}, \mathrm{ID}^*, M_i, \sigma_i) = 0$: **return** 0.
7. **return** 1.

Fig. 1. The restricted target-identity unforgeability experiment.

legitimately signed before a forgery, as detailed in Section 4.3. It does not suffice for a general blind signing service that issues many signatures under one identity, nor for batch credential issuance, both of which require full OMU. Establishing OMU for a zero-online-pairing SM9-compatible design is left open (Section 10). RTU is best read as identity-based existential unforgeability restricted to a single target identity, rather than as a blind-signature OMU notion.

4.3 One-Time Issuance Identities

RTU becomes a meaningful guarantee once the deployment binds each target identity to a single issuance event. The standard way to do this is to derive the SM9 identity from a per-issuance context rather than from a long-lived account name, for example

$$\mathrm{ID} = \mathsf{BankID} \,\|\, \mathsf{Epoch} \,\|\, \mathsf{Account} \,\|\, \mathsf{Ctx}, \qquad \mathrm{ID} = \mathsf{ElectionID} \,\|\, \mathsf{VoterToken},$$

where Ctx and $\mathsf{VoterToken}$ are fresh, unpredictable per-issuance values. The issuer maintains a set $\mathcal{S}$ of identities for which a signing session has been opened, and the protocol is gated by a one-time check: on a request for ID the issuer aborts if $\mathrm{ID} \in \mathcal{S}$ and otherwise inserts ID into $\mathcal{S}$ before running Algorithm 2. Under this policy each target identity is signed at most once, which is exactly the condition that the RTU experiment of Figure 1 models by forbidding $\mathcal{O}^{\mathsf{Sign}}(\mathrm{ID}^*, \cdot)$.

It is important to be precise about what this buys and what it does not. RTU rules out forging a signature for a context that the issuer never signed; it is the right notion when the threat is the *creation of a valid credential out of nothing*, and when each context is consumed by a single issuance. It does not, by itself, rule out a user extracting two signatures from one granted session on the same context. The one-time policy is what closes that gap at the application layer: because the issuer opens at most one session per context and downstream spending or tallying rejects repeated contexts, a second signature on the same context is unusable even if a malicious user could produce it. In one-time e-cash along the lines of [2], a coin is bound to a unique serial context, so a

Experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{A},b}(\lambda)$:

1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $\mathcal{A}(\mathsf{mpk})$ outputs $(M_0, M_1, \mathrm{ID}^*, \mathsf{state})$.
3. Two honest user instances run concurrently with $\mathcal{A}$ as the (possibly malicious) signer for ID^*, which chooses all signer messages: the instance carrying M_b is run in session 0 and the instance carrying M_{1-b} in session 1.
4. Let σ_{M_0} and σ_{M_1} be the signatures produced by the instances that ran on M_0 and on M_1, respectively. If either instance aborts, set $\sigma_{M_0} = \sigma_{M_1} = \perp$.
5. $\mathcal{A}(\mathsf{state}, \sigma_{M_0}, \sigma_{M_1})$ outputs a bit b'; **return** $[b' = b]$.

Fig. 2. The computational blindness experiment against a malicious signer. The final signatures are revealed in message order, so $\mathcal{A}$ must link them to its session views to win.

double-issuance attempt is rejected by the same check that prevents double spending downstream; the residual cryptographic goal handled by RTU is that no party can mint a coin for a context the bank never signed. In token-based e-voting [25,27], each ballot credential is derived from a single-use voter token, so a credential identity is signed once and only once. We stress the converse: if an identity is reused across issuance events, or if the application accepts more than one signature per context, the one-time check no longer holds and RTU no longer captures the relevant threat model, so the construction should not be deployed in that regime without the stronger OMU guarantee discussed in Section 10.

4.4 Computational Blindness

We use the standard malicious-signer blindness game, in which the adversary not only schedules and answers both sessions but is also handed the *final signatures* in message order at the end. This is the strong formulation: the signer must be unable to link a completed signature back to the session that produced it.

Definition 3 (Computational Blindness). *A scheme has* computational blindness *in the ROM if for all PPT* $\mathcal{A}$*,*

$$\mathbf{Adv}^{\mathsf{blind}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{A},0}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{A},1}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{A},b}$ *is the experiment of Figure 2.*

5 Our Construction

The blind signing protocol is given in Algorithm 2. A user-side multiplicative factor α blinds the effective session value $\tilde{w}$, so that the signer never sees the

Algorithm 2 SM9-Compatible Blind Signing Protocol

Input: User holds message M; Signer holds d_A for identity A; public $g = e(P_1, P_{\mathsf{pub}})$.

1: **User:** $k_1 \xleftarrow{\$} \mathbb{Z}_p^*$; $w_1 = g^{k_1}$; send w_1.
2: **Signer: if** $w_1 \notin \mathbb{G}_T$ or $w_1 = 1_{\mathbb{G}_T}$ **then abort**.
3: **Signer:** $k_2 \xleftarrow{\$} \mathbb{Z}_p^*$; $w_{\mathrm{raw}} = w_1^{k_2}$; send w_{raw}.
4: **User: if** $w_{\mathrm{raw}} \notin \mathbb{G}_T$ or $w_{\mathrm{raw}} = 1_{\mathbb{G}_T}$ **then abort**.
5: **User:** $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ (secret); $\tilde{w} = w_{\mathrm{raw}}^{\alpha}$.
6: **User:** $h = H_2(M\|\tilde{w})$; $h'' = (k_1\alpha)^{-1}\, h \bmod p$; send h''.
7: **Signer:** $l = k_2 - h'' \bmod p$; $S' = [l]\, d_A$; send S'.
8: **User: if** $S' \notin \mathbb{G}_1$ **then abort**; $S = [k_1\alpha]\, S'$; **return** $\sigma = (h, S)$.

Verify: run SM9 verification (Algorithm 1) on (A, M, σ).

random oracle input that fixes the challenge h. All user-side scalars are sampled from $\mathbb{Z}_p^*$, so that $(k_1\alpha)^{-1}$ is defined, and both parties reject identity and out-of-group inputs; in particular the user rejects $w_{\mathrm{raw}} = 1_{\mathbb{G}_T}$ (line 4), which is what makes blindness hold against an arbitrary, possibly dishonest, w_{raw}.

Proposition 1 (Correctness). *If both parties are honest, the output $\sigma = (h, S)$ of Algorithm 2 is a valid SM9 signature on M under identity A.*

Proof. With an honest signer $w_{\mathrm{raw}} = g^{k_1 k_2}$, so $\tilde{w} = g^{k_1 k_2 \alpha} = g^k$ for $k := k_1\alpha k_2$. Then

$$S = [k_1\alpha]S' = [k_1\alpha][k_2 - h'']d_A = [k_1\alpha k_2 - k_1\alpha(k_1\alpha)^{-1}h]d_A = [k - h]d_A,$$

and $h = H_2(M\|\tilde{w}) = H_2(M\|g^k)$, which is exactly the SM9 signature on M with randomness k. Hence (h, S) passes Verify of Algorithm 1. □

Efficiency. The online costs are: *User*—2 exponentiations in $\mathbb{G}_T$, one inversion in $\mathbb{Z}_p^*$, and one scalar multiplication in $\mathbb{G}_1$; *Signer*—1 exponentiation in $\mathbb{G}_T$ and one scalar multiplication in $\mathbb{G}_1$; total online pairings 0. The single pairing $g = e(P_1, P_{\mathsf{pub}})$ is pre-computed at setup. The protocol exchanges four messages of total length $2|\mathbb{G}_T| + |\mathbb{Z}_p| + |\mathbb{G}_1|$ bits, where $|\mathbb{G}_T|$, $|\mathbb{Z}_p|$, and $|\mathbb{G}_1|$ are the encoded sizes of a $\mathbb{G}_T$ element, a scalar, and a $\mathbb{G}_1$ element in the chosen SM9 parameter set; the exact figure depends on the representation and compression of $\mathbb{G}_T$ elements. These costs are compared with prior work in Table 1.

6 Security Analysis

6.1 Restricted Target-Identity Unforgeability

Theorem 1 (RTU under Q-BCAA1). *Let H_1, H_2 be random oracles and let $Q \geq q_e$. For any PPT $\mathcal{A}$ making q_{H_1} queries to H_1, q_{H_2} queries to H_2, q_e extraction queries, and no signing session on* ID^**, there is a PPT $\mathcal{B}$ with*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{RTU}}(\lambda) \;\leq\; (q_{H_1}+1)\sqrt{q_{H_2}\,\mathbf{Adv}_{\mathcal{B}}^{Q\text{-BCAA1}}(\lambda)} \;+\; \frac{(q_{H_1}+2)\,q_{H_2}}{p-1} \;+\; \frac{q_{H_1}}{p-1}.$$

Proof. $\mathcal{B}$ receives a Q-BCAA1 instance $(P_1, P_2, P_{\mathsf{pub}} = [s]P_2)$ with oracle $\mathcal{O}_s(\tau) = [1/(s+\tau)]P_1$ and simulates the RTU experiment.

Setup and target guess. $\mathcal{B}$ sets $g = e(P_1, P_{\mathsf{pub}}) = e(P_1, P_2)^s$ and gives $\mathsf{mpk} = (P_1, P_2, P_{\mathsf{pub}}, H_1, H_2)$ to $\mathcal{A}$. It guesses $j^* \xleftarrow{\$} \{1, \ldots, q_{H_1} + 1\}$, the index of the H_1 query that will define ID^*. The extra index $q_{H_1} + 1$ covers the case where $\mathcal{A}$ outputs an ID^* that it never submitted to H_1; in that case $\mathcal{B}$ lazily samples $\tau^* = H_1(\mathrm{ID}^* \| \mathsf{hid})$ at the verification step and treats it as the target value. Thus the guess is correct with probability at least $1/(q_{H_1} + 1)$.

H_1 simulation. On the j-th distinct query $H_1(\mathrm{ID}_j \| \mathsf{hid})$, sample $\tau_j \xleftarrow{\$} \mathbb{Z}_p^*$ and return it; record (ID_j, τ_j). Sampling from $\mathbb{Z}_p^*$ ensures $\tau_j \neq 0$. If a programmed value satisfies $s + \tau_j = 0$, then $\mathcal{O}_s(\tau_j)$ is undefined and $\mathcal{B}$ aborts. Since τ_j is uniform in $\mathbb{Z}_p^*$ and $-s$ is a single fixed value, this happens with probability at most $1/(p-1)$ per identity, hence at most $q_{H_1}/(p-1)$ over all of them.

H_2 simulation. Lazy sampling: return the stored value if present, otherwise sample $h \xleftarrow{\$} \mathbb{Z}_p^*$, store (M, w, h), and return h.

Extraction simulation. On $\mathcal{O}^{\mathsf{Ext}}(\mathrm{ID}_j)$ with $j \neq j^*$, query $D_{\tau_j} = \mathcal{O}_s(\tau_j) = [1/(s+\tau_j)]P_1$ and return

$$d_{\mathrm{ID}_j} = \left[\frac{s}{s+\tau_j}\right] P_1 = P_1 - [\tau_j]\, D_{\tau_j}. \tag{1}$$

If $j = j^*$, abort; a correct guess of j^* implies ID^* is never extracted, since the experiment forbids it. At most $q_e \leq Q$ oracle queries are used.

Signing simulation. A session $\mathcal{O}^{\mathsf{Sign}}(\mathrm{ID}_j, \cdot)$ with $j \neq j^*$ is run honestly as the signer using d_{ID_j} from (1). The RTU restriction guarantees no session is started on ID^*, so d_{ID^*} is never needed during simulation.

Forgeries correspond to H_2 queries. Suppose $\mathcal{A}$ outputs $n \geq 1$ valid forgeries $\{(M_i, (h_i, S_i))\}$ for ID^* with distinct messages. Writing $\tau^* = \tau_{j^*}$, each valid forgery satisfies

$$w_i = g^{h_i} \cdot e(S_i, [\tau^*]P_2 + P_{\mathsf{pub}}), \qquad h_i = H_2(M_i \| w_i).$$

If $\mathcal{A}$ never queried $H_2(M_i \| w_i)$, the equation $h_i = H_2(M_i \| w_i)$ holds with probability at most $1/(p-1)$ over the lazily sampled value. By a union bound over the $n \leq q_{H_2}$ forgeries, except with probability $q_{H_2}/(p-1)$ every valid forgery corresponds to a prior H_2 query.

Forking and key extraction. $\mathcal{B}$ runs a wrapper that executes the simulation, checks that the target guess j^* is correct, that no abort occurred, and that the forgeries correspond to prior H_2 queries. It then selects the first valid forgery whose corresponding H_2 query appears in the transcript and outputs the index of that query together with the forgery. The success probability of this wrapper is at least

$$\text{acc} \;\geq\; \frac{1}{q_{H_1}+1}\left(\epsilon - \frac{q_{H_2}}{p-1} - \frac{q_{H_1}}{p-1}\right),$$

where $\epsilon = \mathbf{Adv}^{\mathrm{RTU}}_{\mathcal{A}}$. The General Forking Lemma [9] rewinds $\mathcal{A}$ to this query, keeping all prior oracle answers and coins fixed and replacing only the answer to the forked query. With output set of size $p-1$ and at most q_{H_2} candidate queries, $\mathcal{B}$ obtains with probability at least $\text{acc}(\text{acc}/q_{H_2} - 1/(p-1))$ a second forgery on the *same* w_{i^*} but with a fresh challenge $h'_{i^*} \neq h_{i^*}$. By non-degeneracy in the prime-order group the discrete exponents are unique, so both forgeries satisfy $S = [k-h]d_{\mathrm{ID}^*}$ for the common k with $g^k = w_{i^*}$, and

$$d_{\mathrm{ID}^*} = [\Delta^{-1}](S_{i^*} - S'_{i^*}) = \left[\frac{s}{s+\tau^*}\right] P_1, \qquad \Delta = h'_{i^*} - h_{i^*} \neq 0.$$

Since $P_1 - d_{\mathrm{ID}^*} = [\tau^*/(s+\tau^*)]P_1$ and $\tau^* \neq 0$,

$$\left[\frac{1}{s+\tau^*}\right] P_1 = [(\tau^*)^{-1}]\,(P_1 - d_{\mathrm{ID}^*}).$$

Because ID^* was never extracted, τ^* was never sent to $\mathcal{O}_s$, so $(\tau^*, [(\tau^*)^{-1}](P_1 - d_{\mathrm{ID}^*}))$ is a fresh Q-BCAA1 solution.

Advantage. Let $A := q_{H_1} + 1$, $\delta := (q_{H_2} + q_{H_1})/(p-1)$, and $x := \epsilon - \delta$, so $\text{acc} \geq x/A$. The forking bound gives

$$\mathbf{Adv}^{Q\text{-BCAA1}}_{\mathcal{B}} \;\geq\; \frac{x^2}{A^2 q_{H_2}} - \frac{x}{A(p-1)}.$$

Treating this as a quadratic in x and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$,

$$x \leq \frac{A q_{H_2}}{p-1} + A\sqrt{q_{H_2}\,\mathbf{Adv}^{Q\text{-BCAA1}}_{\mathcal{B}}}.$$

Substituting A and δ and collecting terms gives the stated bound. □

6.2 Computational Blindness

Theorem 2 (Computational Blindness in the ROM). *Let H_2 be a random oracle. Against any PPT malicious signer $\mathcal{A}$ making q_{H_2} queries to H_2, with the user-side validation of Algorithm 2 enforced and with the final signatures revealed as in Figure 2,*

$$\mathbf{Adv}^{\mathsf{blind}}_{\mathcal{A}}(\lambda) \;\leq\; \frac{2q_{H_2}}{p-1}.$$

Proof. We make no assumption about how $\mathcal{A}$ chooses its protocol messages. In a session $j \in \{0,1\}$ the signer's interaction view is $(w_1^{(j)}, w_{\mathrm{raw}}^{(j)}, h''^{(j)})$, where $w_1^{(j)} = g^{k_1^{(j)}}$ with $k_1^{(j)} \xleftarrow{\$} \mathbb{Z}_p^*$, the value $w_{\mathrm{raw}}^{(j)}$ is chosen *by* $\mathcal{A}$, and

$$h''^{(j)} = (k_1^{(j)}\alpha^{(j)})^{-1} H_2\big(M_{b\oplus j} \,\|\, \tilde{w}^{(j)}\big), \qquad \tilde{w}^{(j)} = (w_{\mathrm{raw}}^{(j)})^{\alpha^{(j)}},$$

with $\alpha^{(j)} \xleftarrow{\$} \mathbb{Z}_p^*$ secret. After the two interactions $\mathcal{A}$ additionally receives the signatures $\sigma_{M_0}, \sigma_{M_1}$ in message order.

Step 1 (first message carries no message dependence). $w_1^{(j)} = g^{k_1^{(j)}}$ depends only on $k_1^{(j)}$, so its distribution is independent of b and is identical in both worlds.

Step 2 (the hidden point is uniform for any w_{raw}). The user proceeds only if $w_{\mathrm{raw}}^{(j)} \in \mathbb{G}_T \setminus \{1_{\mathbb{G}_T}\}$. Since $\mathbb{G}_T$ has prime order p, every non-identity element is a generator, so for the adversarially chosen $w_{\mathrm{raw}}^{(j)} \neq 1_{\mathbb{G}_T}$ the map $\alpha \mapsto (w_{\mathrm{raw}}^{(j)})^{\alpha}$ is a bijection from $\mathbb{Z}_p^*$ onto $\mathbb{G}_T \setminus \{1_{\mathbb{G}_T}\}$. As $\alpha^{(j)}$ is uniform on $\mathbb{Z}_p^*$ and is never sent to $\mathcal{A}$, the value $\tilde{w}^{(j)}$ is uniform on $\mathbb{G}_T \setminus \{1_{\mathbb{G}_T}\}$ from the signer's view and is never transmitted. This step uses only the user's check on line 4.

Step 3 (the signer misses the hidden point). Let $\mathcal{E}$ be the event that $\mathcal{A}$ never queries H_2 at $(\cdot \,\|\, \tilde{w}^{(0)})$ or $(\cdot \,\|\, \tilde{w}^{(1)})$. By Step 2 and a union bound over q_{H_2} queries and two sessions, $\Pr[\neg\mathcal{E}] \leq 2q_{H_2}/(p-1)$.

Step 4 (challenges are uniform and message-independent). Conditioned on $\mathcal{E}$, each $H_2(M_{b\oplus j}\|\tilde{w}^{(j)})$ is a fresh uniform element of $\mathbb{Z}_p^*$, and multiplication by the nonzero $(k_1^{(j)}\alpha^{(j)})^{-1}$ leaves $h''^{(j)}$ uniform on $\mathbb{Z}_p^*$, independent of which message is routed to which session.

Step 5 (the revealed signatures add nothing). A non-aborting instance on message M outputs $\sigma = (h, S)$ with $h = H_2(M\|\tilde{w})$ and $S = [k-h]d_A$ for the unique k with $g^k = \tilde{w}$. By Step 2 the value $\tilde{w}$ is uniform on $\mathbb{G}_T \setminus \{1_{\mathbb{G}_T}\}$ and independent across the two sessions, so σ is distributed exactly as a fresh honest SM9 signature on M (Algorithm 1) and depends only on M, not on the session index that produced it. The same holds for an abort, which yields $\perp$ in both worlds by construction. Hence the pair $(\sigma_{M_0}, \sigma_{M_1})$, presented in message order, has a distribution independent of b and reveals no linkage to the sessions.

Step 6 (conclusion). Under $\mathcal{E}$ the full signer view—interaction transcripts together with the revealed signatures—has the same distribution for $b = 0$ and $b = 1$, so $\mathcal{A}$'s outputs are identically distributed and $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{blind}} \leq \Pr[\neg\mathcal{E}] \leq 2q_{H_2}/(p-1)$, which is negligible for $p \approx 2^{256}$. □

Remark 3 (Role of the blinding factor α). The factor k_1 randomizes $w_1 = g^{k_1}$ and enables unblinding via $S = [k_1\alpha]S'$, while α blinds the hash input so that $\tilde{w} = w_{\mathrm{raw}}^{\alpha}$ is hidden. Without α one would set $\tilde{w} = w_{\mathrm{raw}}$ and send $h'' = k_1^{-1}H_2(M\|w_{\mathrm{raw}})$; since the signer knows $w_1 = g^{k_1}$ and w_{raw}, it could test a candidate M_c by checking $w_1^{h''} \stackrel{?}{=} g^{H_2(M_c\|w_{\mathrm{raw}})}$ and recover the message. Introducing α forces the signer to query H_2 at the hidden $\tilde{w}$, which by Theorem 2 happens only with negligible probability.

Table 1. Theoretical complexity comparison (signing phase).

Scheme	Pairings	T_e	T_{m_1}	T_{m_2}	Extra	Security
Zhang–Kim [22]	3	2	2	0	–	Informal
Cao et al. [23]	2	3	2	0	–	Informal
Zhang–Peng [31]	2	2	2	1	–	Informal
Li et al. [33]	3	4	4	2	Key split	Informal
Zhang et al. [34]	4	5	4	1	FE, MFA	Informal
Ours	$0^\dagger$	3	2	0	–	**Proven (RTU)**

†The single pairing $g = e(P_1, P_{\mathsf{pub}})$ is pre-computed at setup, so online signing uses zero pairings. FE: fuzzy extractor; MFA: multi-factor authentication.

Table 2. Estimated signing latency (ms, Type-A, indicative).

Scheme	Pairing cost	Other cost	Total (est.)
Zhang–Kim [22]	$3 \times 4.5 = 13.5$	≈ 4.0	≈ 17.5
Cao et al. [23]	$2 \times 4.5 = 9.0$	≈ 5.2	≈ 14.2
Zhang–Peng [31]	$2 \times 4.5 = 9.0$	≈ 4.4	≈ 13.4
Li et al. [33]	$3 \times 4.5 = 13.5$	≈ 8.8	≈ 22.3
Zhang et al. [34]	$4 \times 4.5 = 18.0$	≈ 10.4	≈ 28.4
Ours	0	≈ 5.2	$\approx$ **5.2**

7 Performance Comparison

Let $T_p, T_e, T_{m_1}, T_{m_2}$ denote the costs of a pairing, a $\mathbb{G}_T$ exponentiation, and a $\mathbb{G}_1$ and $\mathbb{G}_2$ scalar multiplication. The figures below use PBC benchmarks (Type-A, 80-bit security) on a 2.4 GHz Core i7, with $T_p \approx 4.5$, $T_e \approx 1.2$, $T_{m_1} \approx 0.8$, $T_{m_2} \approx 1.0$ ms, and are indicative only. A production SM9 deployment should be re-benchmarked on the SM9-specified pairing-friendly curve; benchmarks on widely used 128-bit curves such as BLS12-381 may be reported for reference but are not SM9-compliant parameters. The operation counts for prior schemes are taken from their signing phase as described in the cited papers. Table 1 compares signing-phase complexity, Table 2 the estimated latency, and Table 3 the security guarantees.

Under these indicative figures, the elimination of all online pairings reduces the estimated signing latency in Table 2 by roughly 70% relative to Zhang–Kim [22], 63% relative to Cao et al. [23], 61% relative to Zhang–Peng [31], 77% relative to Li et al. [33], and 82% relative to Zhang et al. [34]. The relative ordering is expected to persist on SM9-compliant parameters, since the pairing remains the dominant operation.

As Table 3 records, prior SM9-based proposals give informal arguments, whereas Theorems 1 and 2 provide explicit reductions under Q-BCAA1 and in the ROM. None of the listed schemes, including ours, achieves full OMU.

Table 3. Security comparison.

Scheme	Unforgeability	Blindness	Assumption	OMU?
Zhang–Kim [22]	Informal	Informal	–	no
Cao et al. [23]	Informal	Informal	–	no
Zhang–Peng [31]	Informal	Informal	–	no
Li et al. [33]	Informal	Informal	–	no
Zhang et al. [34]	Informal	Informal	–	no
Ours	RTU (proven)	Comp. (proven)	Q-BCAA1 + ROM	no

Algorithm 3 SM9-Compatible Partial Blind Signing Protocol

Input: User holds $(M_{\mathsf{pub}}, M_{\mathsf{priv}})$; Signer holds d_A and sees M_{pub}.
Agreement: both parties agree on M_{pub} (sent in plaintext).
User: $k_1 \xleftarrow{\$} \mathbb{Z}_p^*$; $w_1 = g^{k_1}$; send w_1.
Signer: if $w_1 \notin \mathbb{G}_T$ or $w_1 = 1_{\mathbb{G}_T}$ **then abort**.
Signer: $k_2 \xleftarrow{\$} \mathbb{Z}_p^*$; $w_{\mathrm{raw}} = w_1^{k_2}$; send w_{raw}.
User: if $w_{\mathrm{raw}} \notin \mathbb{G}_T$ or $w_{\mathrm{raw}} = 1_{\mathbb{G}_T}$ **then abort**.
User: $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ (secret); $\tilde{w} = w_{\mathrm{raw}}^{\alpha}$.
User: $h = H_2(M_{\mathsf{pub}} \| M_{\mathsf{priv}} \| \tilde{w})$; $h'' = (k_1\alpha)^{-1} h \bmod p$; send h''.
Signer: $l = k_2 - h'' \bmod p$; $S' = [l]\, d_A$; send S'.
User: if $S' \notin \mathbb{G}_1$ **then abort**; $S = [k_1\alpha]\, S'$; **return** $\sigma = (h, S)$.
Verify: run SM9 verification on $(A, M_{\mathsf{pub}} \| M_{\mathsf{priv}}, \sigma)$.

8 Extension: Partial Blind Signatures

Partial blind signatures [6,7] keep part of the message visible to the signer while blinding the rest, which is useful for auditable e-cash, conditional credentials, and traceable e-voting. Let $M = (M_{\mathsf{pub}}, M_{\mathsf{priv}})$, where M_{pub} is agreed in the clear and M_{priv} is blinded. Algorithm 3 gives the extension; it differs from Algorithm 2 only in binding M_{pub} into the hash.

8.1 Partial-Blindness Model

Partial blindness asks that, for a fixed common public part, the signer cannot tell which of two private parts it signed, even after seeing the final signatures. The experiment of Figure 3 is the standard partial-blindness game [7] adapted to the malicious-signer setting: the adversary commits to a common info $(= M_{\mathsf{pub}})$ and two private parts, both sessions use the same info, the challenge bit decides which private part is routed to which session, and the resulting signatures are revealed in private-message order. Fixing a common info is essential, since a signer handed two different public parts could trivially distinguish the sessions.

Definition 4 (Computational Partial Blindness). *A partial blind signature scheme has* computational partial blindness *in the ROM if for all PPT $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathsf{pblind}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{pblind}}_{\mathcal{A},0}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{pblind}}_{\mathcal{A},1}(\lambda) = 1]\right| \le \mathsf{negl}(\lambda),$$

Experiment $\mathsf{Exp}^{\mathsf{pblind}}_{\mathcal{A},b}(\lambda)$:

1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $\mathcal{A}(\mathsf{mpk})$ outputs $(\mathsf{info}, M_0, M_1, \mathrm{ID}^*, \mathsf{state})$, where info is the common public part shared by both sessions.
3. Two honest user instances run concurrently with public part info and $\mathcal{A}$ as the (possibly malicious) signer for ID^*: the instance carrying M_b runs in session 0 and the instance carrying M_{1-b} in session 1.
4. Let $\sigma_{M_0}, \sigma_{M_1}$ be the signatures produced by the instances that ran on M_0 and M_1. If either aborts, set $\sigma_{M_0} = \sigma_{M_1} = \bot$.
5. $\mathcal{A}(\mathsf{state}, \sigma_{M_0}, \sigma_{M_1})$ outputs a bit b'; **return** $[b' = b]$.

Fig. 3. The computational partial-blindness experiment against a malicious signer, with final signatures revealed.

where $\mathsf{Exp}^{\mathsf{pblind}}_{\mathcal{A},b}$ *is the experiment of Figure 3.*

8.2 Security of the Partial-Blind Construction

Theorem 3 (Security of Algorithm 3). *Under the assumptions of Theorems 1 and 2, Algorithm 3 is restricted target-identity unforgeable and computationally partially blind. Concretely, its RTU advantage is bounded as in Theorem 1, and against any PPT malicious signer making q_{H_2} queries to H_2,* $\mathbf{Adv}^{\mathsf{pblind}}_{\mathcal{A}}(\lambda) \le 2q_{H_2}/(p-1)$.

Proof. Unforgeability. The reduction of Theorem 1 applies unchanged. The only difference in Algorithm 3 is that the hashed value is $M_{\mathsf{pub}} \| M_{\mathsf{priv}} \| \tilde{w}$ rather than $M \| \tilde{w}$. During signing simulation the reduction knows M_{pub} (it is agreed in the clear) and treats the concatenation $M_{\mathsf{pub}} \| M_{\mathsf{priv}}$ as the signed message; the H_1 programming, the BCAA1-based key extraction of (1), and the forking step at the target H_2 query are independent of how the message is parsed. Hence the same bound holds.

Partial blindness. Fix the common public part info output by $\mathcal{A}$. In session $j \in \{0,1\}$ the signer's interaction view is $(w_1^{(j)}, w_{\mathrm{raw}}^{(j)}, h''^{(j)})$ with

$$h''^{(j)} = (k_1^{(j)} \alpha^{(j)})^{-1} H_2\big(\mathsf{info} \,\|\, M^{(j)} \,\|\, \tilde{w}^{(j)}\big), \qquad \tilde{w}^{(j)} = (w_{\mathrm{raw}}^{(j)})^{\alpha^{(j)}},$$

where $M^{(j)} \in \{M_0, M_1\}$ is the private part routed to session j. As in Theorem 2, $w_1^{(j)}$ depends only on $k_1^{(j)}$ and is message-independent; the user's check on $w_{\mathrm{raw}}^{(j)} \in \mathbb{G}_T \setminus \{1_{\mathbb{G}_T}\}$ together with the prime order of $\mathbb{G}_T$ makes $\tilde{w}^{(j)}$ uniform on $\mathbb{G}_T \setminus \{1_{\mathbb{G}_T}\}$ from the signer's view, for any adversarial $w_{\mathrm{raw}}^{(j)}$. Let $\mathcal{E}$ be the event that $\mathcal{A}$ never queries H_2 at the correct $(\cdot \,\|\, \tilde{w}^{(0)})$ or $(\cdot \,\|\, \tilde{w}^{(1)})$; then $\Pr[\neg\mathcal{E}] \le 2q_{H_2}/(p-1)$. Conditioned on $\mathcal{E}$, the common info is identical in both worlds, each $H_2(\mathsf{info} \| M^{(j)} \| \tilde{w}^{(j)})$ is a fresh uniform scalar, and multiplication

by the nonzero $(k_1^{(j)}\alpha^{(j)})^{-1}$ keeps $h''^{(j)}$ uniform on $\mathbb{Z}_p^*$ independently of which private part is routed where. Finally, as in Step 5 of Theorem 2, each revealed signature $\sigma_{M^{(j)}}$ is distributed as a fresh honest SM9 signature on $\mathsf{info}\|M^{(j)}$ determined by the session-independent uniform point $\tilde{w}^{(j)}$, so presenting $(\sigma_{M_0}, \sigma_{M_1})$ in private-message order leaks no linkage. Thus the signer's view is identically distributed for $b = 0$ and $b = 1$, and $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{pblind}} \leq \Pr[\neg\mathcal{E}] \leq 2q_{H_2}/(p-1)$. □

Algorithm 3 supports auditable e-cash (the bank sees the denomination and expiry in M_{pub} while the serial number in M_{priv} is blinded), conditional credentials (a public attribute such as "age ≥ 18" is visible while exact attributes are blinded), and traceable e-voting (election metadata is visible while the choice is blinded). The signer learns nothing about M_{priv} beyond what M_{pub} already reveals.

9 Implementation Considerations

The security proofs assume that a few protocol-level checks are enforced, and a deployment that omits them can lose either unforgeability or blindness regardless of the analysis above. This section collects the requirements.

Group-membership and subgroup checks. Both parties must verify that every received element lies in the correct group before using it. The signer checks $w_1 \in \mathbb{G}_T$ and the user checks $w_{\mathrm{raw}} \in \mathbb{G}_T$ and $S' \in \mathbb{G}_1$ (Algorithm 2, lines 2, 4, 8). On curves with a non-trivial cofactor these must be full subgroup-membership checks, not just field-range checks, so that a malicious peer cannot inject a small-subgroup element. Rejecting the identity element $1_{\mathbb{G}_T}$ is what Theorem 2 relies on: a signer that returns $w_{\mathrm{raw}} = 1_{\mathbb{G}_T}$ would otherwise fix $\tilde{w} = 1_{\mathbb{G}_T}$ and learn the hash input.

Scalar sampling. All user-side scalars k_1, α and the signer scalar k_2 must be sampled uniformly from $\mathbb{Z}_p^*$, never from $\mathbb{Z}_p$. The unblinding step computes $(k_1\alpha)^{-1}$, which is defined only when $k_1, \alpha \neq 0$, and the correctness of Algorithm 2 relies on $\tilde{w} = g^{k_1 k_2 \alpha} \neq 1_{\mathbb{G}_T}$, which requires $k_2 \neq 0$. Separately, the nonzero range of H_1 keeps $\tau = H_1(\mathrm{ID}\|\mathsf{hid})$ away from 0, matching the H_1 simulation in the proof of Theorem 1.

Unambiguous hash encoding. The argument of H_2 must use a fixed, injective encoding of its inputs so that distinct logical inputs never collide as byte strings. For the blind protocol this is $M\|\mathsf{enc}_{\mathbb{G}_T}(\tilde{w})$, and for the partial-blind protocol it is $\mathsf{enc}(\mathsf{info})\|\mathsf{enc}(M_{\mathsf{priv}})\|\mathsf{enc}_{\mathbb{G}_T}(\tilde{w})$ with length-prefixed fields, so that the boundary between the public and private parts is unambiguous. Without length separation, an adversary could shift bytes between M_{pub} and M_{priv} and break the partial-blindness guarantee of Theorem 3.

Randomness handling and constant-time arithmetic. The per-session scalars must come from a cryptographically secure generator and must not be reused across sessions: reusing α can link signatures issued to the same user, and reusing k_1 leaks relations between sessions. The operations that touch secret material—scalar multiplication in $\mathbb{G}_1$, exponentiation in $\mathbb{G}_T$, and the modular inverse $(k_1\alpha)^{-1}$—should be implemented in constant time to avoid side-channel leakage of the blinding factors.

One-time issuance enforcement. For the RTU guarantee to apply, the issuer must enforce the single-use policy of Section 4.3: maintain the set of identities for which a session has been opened and refuse a second session on the same identity. This check is an application-layer obligation; the cryptographic proof of Theorem 1 models it as the prohibition on $\mathcal{O}^{\mathsf{Sign}}(\mathrm{ID}^*, \cdot)$ in Figure 1.

SM9 interoperability. The output $\sigma = (h, S)$ of Algorithm 2 is, bit for bit, a standard SM9 signature, so it verifies under an unmodified SM9 verifier (Algorithm 1). The blinding layer lives entirely in the interactive issuance phase; the key-generation centre and the verifier require no changes, which is what makes the scheme drop-in compatible with existing SM9 deployments.

10 Conclusion and Open Problems

We presented an SM9-compatible identity-based blind signature scheme (Algorithm 2) that outputs standard SM9 signatures, removes all online pairings, and adds a single user-side blinding exponentiation. We proved restricted target-identity unforgeability under the Q-BCAA1 assumption (Theorem 1) and computational blindness against a malicious signer that is handed the final signatures (Theorem 2), and we gave a partial-blind extension (Algorithm 3) with a formal model (Definition 4, Figure 3) and proof (Theorem 3). We have been explicit that RTU is weaker than standard OMU and is the appropriate guarantee only for the one-time issuance identities of Section 4.3; for that setting the scheme is, to our knowledge, the first SM9-compatible blind signature with explicit reductions and zero online pairings.

Open problem: full one-more unforgeability. Establishing OMU—in which the adversary obtains ℓ signatures under ID^* and must produce $\ell + 1$—for a zero-online-pairing SM9-compatible design remains the main open problem. The obstacle is that the reduction does not know the target key yet must simulate signing sessions under ID^*, which seems to require extractable commitments (and the loss of efficiency that zero-knowledge proofs bring) or a different algebraic structure; the ROS attacks [11] also caution against optimistic concurrency claims for Schnorr-style designs. Trapdoor-based blinding and dual-system techniques are plausible directions. A second direction is to validate the indicative performance of Section 7 on SM9-compliant 128-bit parameters and to study batch verification of blind-issued SM9 signatures for high-throughput e-cash and voting systems.

References

1. Chaum, D.: Blind signatures for untraceable payments. In: Crypto 1982, pp. 199–203. Plenum Press (1983)
2. Chaum, D., Fiat, A., Naor, M.: Untraceable electronic cash. In: Crypto 1988, LNCS, vol. 403, pp. 319–327. Springer (1990)
3. Pointcheval, D., Stern, J.: Provably secure blind signature schemes. In: Asiacrypt 1996, LNCS, vol. 1163, pp. 252–265. Springer (1996)
4. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptology **13**(3), 361–396 (2000)
5. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures. In: Crypto 1997, LNCS, vol. 1294, pp. 150–164. Springer (1997)
6. Abe, M., Fujisaki, E.: How to date blind signatures. In: Asiacrypt 1996, LNCS, vol. 1163, pp. 244–251. Springer (1996)
7. Abe, M., Okamoto, T.: Provably secure partially blind signatures. In: Crypto 2000, LNCS, vol. 1880, pp. 271–286. Springer (2000)
8. Abe, M.: A secure three-move blind signature scheme for polynomially many signatures. In: Eurocrypt 2001, LNCS, vol. 2045, pp. 136–151. Springer (2001)
9. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: CCS 2006, pp. 390–399. ACM (2006)
10. Hauck, E., Kiltz, E., Loss, J.: A modular treatment of blind signatures from identification schemes. In: Eurocrypt 2019, LNCS, vol. 11478, pp. 345–375. Springer (2019)
11. Benhamouda, F., Lepoint, T., Loss, J., Orrù, M., Raykova, M.: On the (in)security of ROS. J. Cryptology **35**(4), 25 (2022); also Eurocrypt 2021
12. Fuchsbauer, G., Plouviez, A., Seurin, Y.: Blind Schnorr signatures and signed ElGamal encryption in the algebraic group model. In: Eurocrypt 2020, LNCS, vol. 12106, pp. 63–95. Springer (2020)
13. Kastner, J., Loss, J., Xu, J.: On pairing-free blind signature schemes in the algebraic group model. In: PKC 2022, LNCS, vol. 13178, pp. 468–497. Springer (2022)
14. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Crypto 2015, LNCS, vol. 9216, pp. 233–253. Springer (2015)
15. Fuchsbauer, G., Hanser, C., Kamath, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model from weaker assumptions. In: SCN 2016, LNCS, vol. 9841, pp. 391–408. Springer (2016)
16. Agrawal, S., Kirshanova, E., Stehlé, D., Yadav, A.: Practical, round-optimal lattice-based blind signatures. In: CCS 2022, pp. 39–53. ACM (2022)
17. Beullens, W., Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Lattice-based blind signatures: short, efficient, and round-optimal. In: CCS 2023, pp. 16–29. ACM (2023)
18. Chairattana-Apirom, R., Hanzlik, L., Loss, J., Lysyanskaya, A., Wagner, B.: Pairing-free blind signatures from standard assumptions in the ROM. In: Crypto 2024, LNCS, vol. 14920, pp. 210–245. Springer (2024)
19. Rückert, M.: Lattice-based blind signatures. In: Asiacrypt 2010, LNCS, vol. 6477, pp. 413–430. Springer (2010)
20. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Eurocrypt 2004, LNCS, vol. 3027, pp. 56–73. Springer (2004)
21. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In: PKC 2003, LNCS, vol. 2567, pp. 31–46. Springer (2003)

22. Zhang, F., Kim, K.: ID-based blind signature and ring signature from pairings. In: Asiacrypt 2002, LNCS, vol. 2501, pp. 533–547. Springer (2002)
23. Cao, X., Kou, W., Du, X.: Efficient identity-based signatures and blind signatures. In: CANS 2005, LNCS, vol. 3810, pp. 120–133. Springer (2005)
24. Chow, S.S.M., Hui, L.C.K., Yiu, S.M., Chow, K.P.: Two improved partially blind signature schemes from bilinear pairings. In: ACISP 2005, LNCS, vol. 3574, pp. 316–328. Springer (2005)
25. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Auscrypt 1992, LNCS, vol. 718, pp. 244–251. Springer (1993)
26. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Eurocrypt 2001, LNCS, vol. 2045, pp. 93–118. Springer (2001)
27. Kumar, M., Chand, S., Katti, C.P.: A secure end-to-end verifiable internet-voting system using identity-based blind signature. IEEE Systems Journal **14**(2), 2032–2041 (2020)
28. State Cryptography Administration of China: GM/T 0044-2016 SM9 Identity-Based Cryptographic Algorithm. SCA (2016); also ISO/IEC 14888-3:2018
29. Cheng, Z.: Security analysis of SM9 key agreement and encryption. In: Inscrypt 2018, LNCS, vol. 11449, pp. 3–25. Springer (2019)
30. Lai, J., Huang, X., He, D., Wu, W.: Security analysis of SM9 digital signature and key encapsulation. Sci. China Inf. Sci. **64**(11), 219101 (2021)
31. Zhang, P., Peng, C.: Blind signature scheme based on SM9 algorithm and its application on e-commerce. In: ICCS 2021 (2021)
32. Xu, J., et al.: Partial blind signature scheme based on SM9 algorithm. Netinfo Security **21**(8), 1–8 (2021)
33. Li, H., et al.: Two-party cooperative blind signature scheme based on SM9. Chinese Journal of Electronics **32**(3), 1–10 (2023)
34. Zhang, J., Li, S., Chen, M.: Multifactor SM9 blind signature large models data privacy preservation approach. Peer-to-Peer Netw. Appl. **18**, 184 (2025)