KARYA ILHAM PUBLISHING
CONNECTING MINDS, BRIDGING IDEAS, INSPIRING READERS

Journal of Advanced Research in Computing and Applications

Journal homepage:
https://karyailham.com.my/index.php/arca/index
ISSN: 2462-1927



Implementation of an Identity-Based Blockchain Signature Protocol on Smart Phones

Yi Liu[1,2,*], Sharifah Md Yasin[1,3], Mohd Izuan Hafez Ninggal[1], Aziah Asmawi[1]

[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Jalan Universiti 1, 43400 Serdang, Selangor, Malaysia
[2] Guizhou Traffic Technician and Transpartation College,Guiyang, 550008, Guizhou, China
[3] Institute of Mathematical Research, Universiti Putra MalaysiaMalaysia, Jalan Universiti 1, 43400 Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 20 August 2026
Received in revised form 20 September 2026
Accepted 30 September 2026
Available online 12 October 2026

Keywords:
identity-based signature; blockchain; smartphone; trusted execution environment; hardware-backed keystore; pairing-free elliptic-curve cryptography

ABSTRACT

As of now, people's primary devices for carrying their personal digital identity are smartphones, and thus a genuine need has arisen for trustworthy signatures and private keys that cannot be stolen from a phone. Identity-Based Signatures (IBS) are used to provide an identity string as the public key without requiring certificate management, and can be applied to a blockchain as the root of trust. Most pairing-free IBS schemes that have been combined with blockchain have only been tested on IoT boards or servers, and how a phone's hardware-backed keystore protects an IBS private key has rarely been validated in practice. This paper implements a pairing-free elliptic-curve IBS protocol on commodity Android smartphones, seals the user's private key in the device keystore (TEE by default, StrongBox when available), and anchors identity bindings and revocation status on a permissioned blockchain. We test the protocol on three real devices and a four-node permissioned network. On-device signing is around 3.1ms for operation and about 5.1ms for verification on a high-end mobile phone; it is about 12.5 times faster than the pairing-based baseline for signing, and although richer in functions, it is measured to be about 1.6 times slower than plain ECDSA. The cost of the highest-security level is concentrated in the per-signature key unsealing of StrongBox (about 29ms), not in the signature generation process. The primary cause of the extended end-to-end latency is chain confirmation; the proportion of phone-side cryptography in the total at high-end TEE devices is about 1.4%. Based on the above results, a pairing-free IBS for smartphones is feasible, and TEE has a good balance of security and performance.

1. Introduction

Smartphones are now used for mobile payment and e-government services, so users need a trusted signature and secure key storage on their phones [1]. The traditional Public Key Infrastructure has a costly certificate management problem in mobile and decentralized environments. The spread of decentralized identifiers and self-sovereign identity has also increased the demand for mobile phones as identity devices [2-3].

* *Corresponding author.*
E-mail address: 987014608@qq.com

Implementing IBS inside a commodity smartphone raises three practical challenges. First, computation is constrained: bilinear pairings are costly on mobile processors and drain the battery, which motivates pairing-free elliptic-curve constructions that keep signing within a few milliseconds on a handset [4]. Second, key management is constrained by the platform, because commodity hardware keystores natively expose only standard algorithms; an IBS private key therefore cannot execute inside the keystore and must be sealed at rest and unsealed into the application for each signature, creating a runtime exposure window that has to be reasoned about explicitly. Third, hardware security is heterogeneous: the Trusted Execution Environment has a documented attack surface [5], the StrongBox secure element is absent on most mid- and low-end devices, and empirical measurements show that secure-element operations carry a non-trivial performance penalty [6]. These constraints, rather than the signature mathematics, dominate a real deployment and motivate the measurement study reported in this paper.

Identity-Based Signatures (IBS) use a user's identity as the public key to avoid certificate management and are thus more suitable for the trust-minimizing design of blockchains [7]. Pairing-based IBS, however, is computationally expensive for phones, and most pairing-free IBS schemes integrated with blockchain have been evaluated on IoT development boards or server-class machines [4], [8], [9]. Identity-based signatures have also been benchmarked on phones, smartwatches and Raspberry Pi devices [10], but those studies keep the private key in software; how a phone's hardware-backed keystore should protect an IBS private key, and how the TEE, StrongBox and software tiers compare in practice, has rarely been studied empirically [6].

In this paper, the blockchain is used to record the auditable history of public parameters, identity-to-public-key mappings and revocation statuses, but not to store application data. This work is an implementation and measurement study rather than a new cryptographic construction: the pairing-free IBS we use is standard, and the novelty lies in its on-device realisation, hardware-backed key protection, and tier-level evaluation. Our contributions are threefold. First, we implement a pairing-free elliptic-curve IBS protocol on commodity Android phones and protect the user's private key with the device hardware-backed keystore (TEE/StrongBox). Second, we anchor identity bindings and revocation status in a permissioned blockchain and reserve an algorithm-identifier field for crypto-agile migration. Third, we measure the protocol on three real devices and report the performance and selection comparison of the TEE, StrongBox and software tiers. The following are the other parts of this paper: Section 2 shows the materials and methods; Section 3 presents the results; Section 4 discusses the results; and Section 5 concludes.

2. Materials and Methods

2.1 Protocol and system model

The four modules of the system are: the private key generator (PKG), the smartphone signer, the verifier and the nodes of a permissioned blockchain. The PKG derives a user's private key from a master key and the user's identity, and key escrow at the PKG is a known attribute of this setup [11]. The four algorithms of the protocol are Setup, KeyDerive, Sign and Verify; it is a pairing-free, Schnorr-type construction based on the elliptic-curve discrete logarithm problem (ECDLP) that meets the computational requirements of a phone [12]. The on-chain payload is limited to a hash of the public parameters, identity-to-public-key bindings, revocation status, and an algorithm-identifier field; signatures and application data are kept off-chain to limit the volume of data. The whole structure is shown in Figure 1.

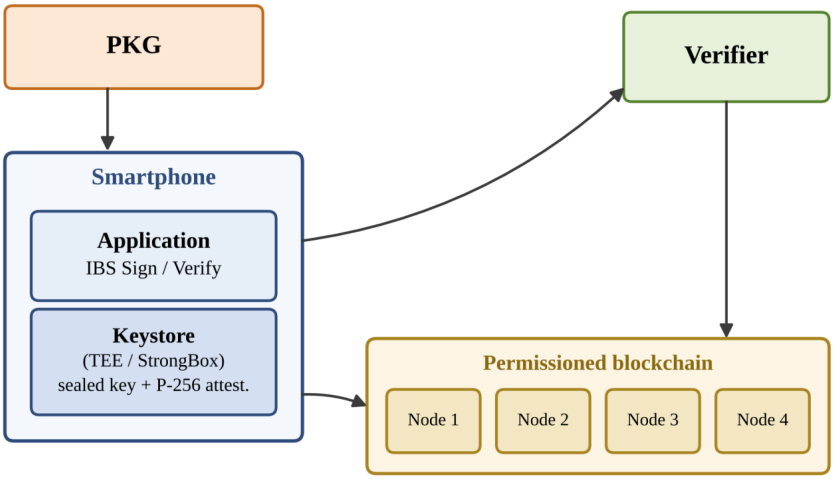


Fig. 1. System Architecture of the Identity-based Blockchain Signature Protocol on Smartphones. The PKG derives and seals the user key in the phone; the application layer performs IBS signing and verification; the hardware-backed keystore (TEE/StrongBox) stores the sealed key and a non-exportable P-256 attestation key; identity bindings, revocation status, and the algorithm identifier are written to the permissioned blockchain, and signature verification is performed off-chain

2.2 Smartphone implementation and key protection

The protocol is realised by an Android application that performs IBS operations using the standard elliptic-curve library on the secp256r1 curve with SHA-256, which provides a 128-bit security level. Since the commodity hardware keystores natively support only standard algorithms, the IBS user private key is sealed at rest in the Android keystore (TEE by default, StrongBox when available) and unsealed into the application for signing; the signature computation itself is performed at the application layer. A non-exportable hardware P-256 key is generated and used for key attestation of the identity binding to create a verifiable device-to-identity link. The security properties and known attack surfaces of mobile Trusted Execution Environments are discussed in [5], and the feasibility of pairing-free elliptic-curve schemes on resource-constrained devices is supported by [4]. A three-tier model is used to fall back from StrongBox to TEE and finally to software, with some key operations gated by biometric authentication. The whole-chain signing-to-verification process is shown in Figure 2.

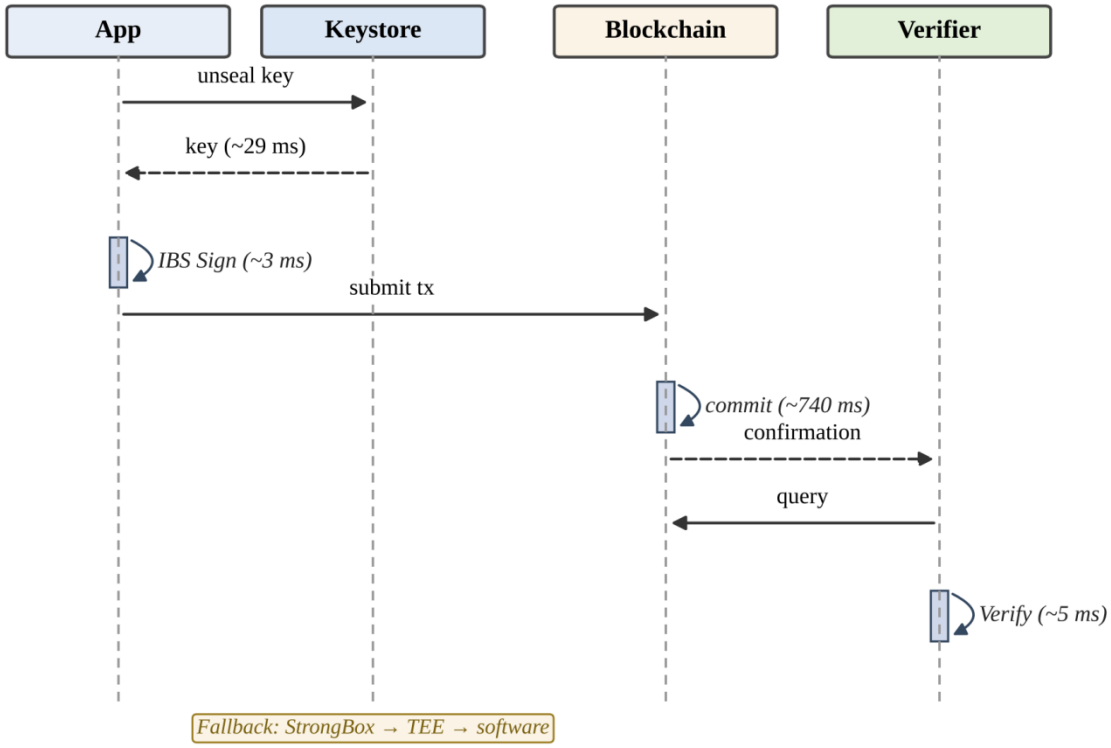


Fig. 2. Signing-to-Verification Sequence across Application, Hardware Keystore, Permissioned Blockchain and Verifier, with StrongBox→TEE→Software Fallback. The main unsealing is about 29ms (StrongBox), signing is about 3ms, chain commit is around 740ms, and verification is about 5ms

2.3 Blockchain anchoring and revocation

We use a permissioned blockchain (Hyperledger Fabric type) because of its controlled membership, predictable confirmation latency, and lack of a public-chain token [13]. On-chain data structures will store the identity-to-public-key binding record, the revocation list, and a crypto-agile algorithm identifier field for future migration. Revocation has a scalable mechanism to prevent signing-key exposure [14]. A verifier queries the chain to check for a binding and revocation status, then performs signature verification off-chain.

2.4 Implementation and evaluation methodology

The three real Android phones used as devices under test are a high-end phone supporting StrongBox (Device-H), a mid-range phone with only TEE (Device-M), and an entry-level phone using the software keystore (Device-L). A local four-node permissioned blockchain network (two endorsing nodes, one ordering node and one committing node) is used, Raft ordering is applied, the block size limit is 2MB, and the batch timeout is 1000ms. The evaluation indicators are key-derivation, unsealing, signing and verification latency; the cost of per-signature unseal-plus-sign; signature and record sizes; and end-to-end latency. Each operation is repeated 1000 times for each condition in the main runs, and the cost model and comparison conventions for IBS on the blockchain follow [8]. The three control groups are pairing-based IBS, software ECDSA, and the proposed hardware-keystore scheme. At batch timeouts of 500ms, 1000ms and 2000ms, chain confirmation was performed; for messages of length 64, 256 and 1024 bytes, two sensitivity analyses of signing latency were carried out. Mann-Whitney U tests were employed to compare the data of two groups, and Kruskal-Wallis

tests were conducted for three or more groups; at a significance level of 0.05, the effect size was reported as a latency ratio. Between March and June 2026, data were collected. After excluding the warm-up call, excluding within-group outliers exceeding five times the condition median (0.42%-1.27% per condition), and retaining only the calls with a self-verified signature, there were 8001 valid records for analysis. All reported statistics are based on the cleaned dataset. The experimental structure is shown in Table 1.

Table 1
Experimental devices, Permissioned-Blockchain Network, and Evaluation Parameters

Field	Value / setting	Unit / grouping
Device (anonymized)	Device-H / Device-M / Device-L	high / mid / baseline
Highest hardware security tier	StrongBox+TEE / TEE only / software	per device
SoC / chipset	Tensor G4 / Snapdragon 7s Gen 2 / Helio G99	per device (H / M / L)
Secure hardware (type, version)	Titan M2 secure element (StrongBox) / TrustZone TEE / software keystore	per device
Android / API level	15 / 14 / 14 (API 35 / 34 / 34)	per device
RAM	16 / 8 / 4	GB
Release year	2024 / 2024 / 2024	year
Elliptic curve	secp256r1 (P-256)	all devices
Hash function	SHA-256	all devices
Blockchain platform	Hyperledger Fabric style, v2.5	single network
Nodes	4 (2 endorsing + 1 ordering + 1 committing)	count
Ordering / block params	Raft; block 2 MB; batch timeout 1000 ms	fixed
Repetitions	signing/verification 1000; sensitivity 500	runs per condition
Reported statistics	mean, SD, median (P50), P95	ms or bytes
Control groups	pairing-based IBS / software ECDSA / proposed	3 groups

3. Results

3.1 Signing and verification performance

The pairing-free IBS on the high-end device signs in 3.08 ± 0.90ms and verifies in 5.13 ± 1.18ms. The latency of the paired baseline signs was 38.48 ± 5.91ms and the verification time was 33.98 ± 5.38ms, so the proposed scheme was about 12.5 times faster for signing (Mann-Whitney U, $p < 0.001$; latency ratio 0.078). Software ECDSA (1.88 ± 0.50 ms) is relatively fast; however, the IBS scheme is approximately 1.6 times slower ($p < 0.001$; ratio 1.63) due to the addition of identity-based functions. The mid-range device is slower than the high-end device (4.42 ± 1.30 ms; $p < 0.001$). Kruskal-Wallis tests for the three schemes are significant ($H = 3232.1$, total $n = 5188$, $p < 0.001$). This pairing-free ECDLP-based cost profile is consistent with the previous construction [15]. Figure 3 is the comparison with standard-deviation error bars of the actual measurement spread.

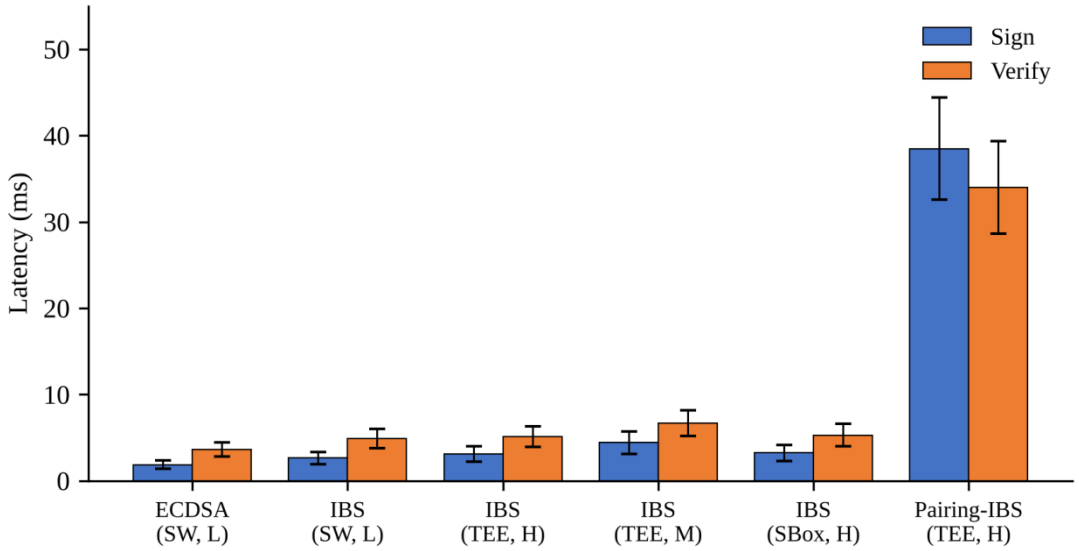


Fig. 3. Signing and Verification Latency by Scheme and Device Tier (ms; error bars are SD)

3.2 Effect of the hardware-backed keystore tier

The computation of the signature itself is relatively independent of the tier (3.08ms in TEE-H, 3.23ms in StrongBox-H, 2.63ms in software), and a signature can be generated at the application level. The tier cost is shown instead in per-signature key unsealing: 0.39 ± 0.09 ms for software, 2.32 ± 0.71 ms for TEE-H, and 28.58 ± 5.97 ms for StrongBox-H. As a result, the full unseal-plus-sign path is 5.40 ± 1.16 ms for TEE-H and 31.81 ± 6.05 ms for StrongBox-H. At the same time, the key generation speed at all levels is relatively high (around 3.05ms for TEE-H and 3.21ms for StrongBox-H), so it is repeated unsealing overhead rather than an initial setup cost for StrongBox. Figure 4 shows the separation of the unsealing component from the signing-path total, and a higher security level is thus more costly per signature [16].

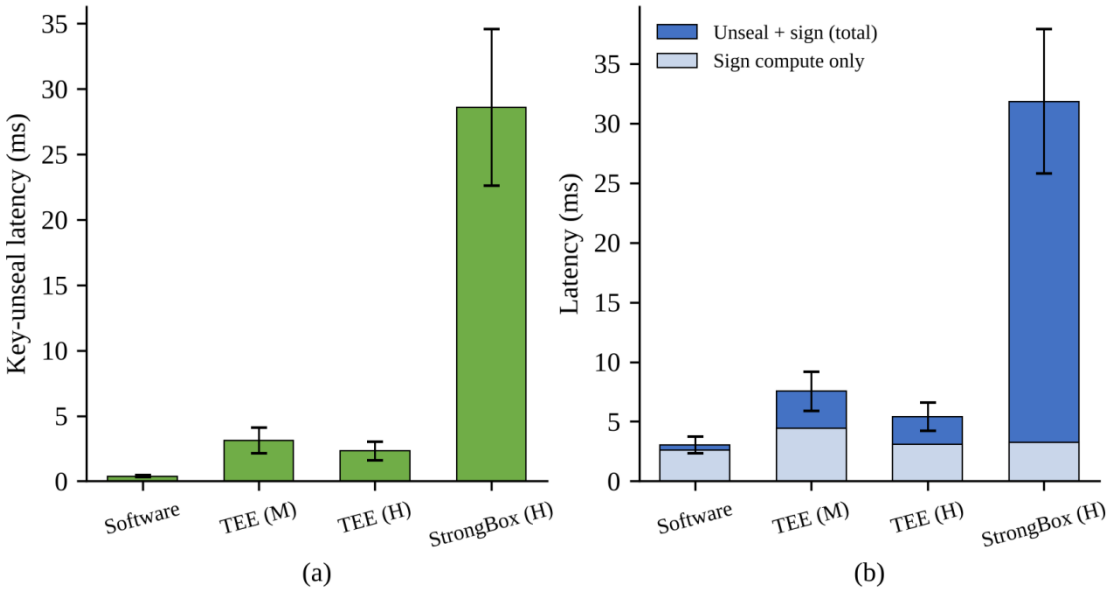


Fig. 4. Key-operation overhead by keystore tier: (a) per-signature unsealing; (b) unseal-plus-sign total vs sign compute (ms; error bars are SD)

3.3 Storage and communication overhead

The length of the IBS signature is fixed at 64 bytes, the user's private key is 32 bytes long, and the length of public parameters is 96 bytes. The pair-based baseline has a larger signature (about 80 bytes on average) and 160-byte public parameters. Each on-chain binding/status record is approximately 580-660 bytes under the conditions (e.g., 581 bytes for TEE-H and 652 bytes for StrongBox-H), which is less than 1 KB, and no application data is stored on-chain. The size figures and the bandwidth comparison are in line with the message-recovery convention in [17]. The above values are shown together with the main latency results in Table 2.

3.4 End-to-end latency

The whole process of the proposed scheme (sign, submit, chain confirmation, verify) is dominated by chain confirmation. For the four IBS conditions in the permissioned network, the confirmation latency is 743-796ms and the end-to-end latency is 761-785ms. Cryptography in the phone part is a relatively small proportion; on-device computation (unseal, sign, verify) is about 10.5ms for TEE-H, 14.2ms for TEE-M, and 37.1ms for StrongBox-H, making up about 1.4%, 1.8% and 4.8% of the total end-to-end time, respectively. Therefore, the performance bottleneck is in the chain rather than in the handset [9]. The software-ECDSA and pairing baselines were run with a lighter commit setting (a fixed 50ms nominal commit and a 78ms path), so their end-to-end figures cannot be directly compared with those of the consortium-anchored IBS and are only reported for reference. The main results and the end-to-end breakdown are shown in Table 2, and the chain-configuration sensitivity is presented in Table 3 in Section 3.

Table 2
Main results of the cleaned dataset (mean ± SD): Computation performance, storage and end-to-end latency. The four IBS conditions are in the permissioned network, and the two baselines have a lighter commit setting and are shown for reference. The effective sample sizes for different metrics vary: n = 3319 was obtained for signing and chain confirmation of IBS-TEE-H; unsealing, verification and end-to-end had n = 1819, as some runs only logged the signing and commit timestamps, and the other cells had n = 932-940

Metric	IBS TEE-H	IBS TEE-M	IBS SBox-H	IBS SW-L	Pairing-IBS	ECDSA	Unit
Signing latency	3.08 ± 0.90	4.42 ± 1.30	3.23 ± 0.91	2.63 ± 0.69	38.48 ± 5.91	1.88 ± 0.50	ms
Verification latency	5.13 ± 1.18	6.68 ± 1.50	5.30 ± 1.30	4.91 ± 1.12	33.98 ± 5.38	3.63 ± 0.79	ms
Key unsealing	2.32 ± 0.71	3.10 ± 0.99	28.58 ± 5.97	0.39 ± 0.09	2.80 ± 0.66	0.40 ± 0.10	ms
Unseal + sign (total)	5.40 ± 1.16	7.53 ± 1.65	31.81 ± 6.05	3.02 ± 0.70	41.28 ± 5.95	2.28 ± 0.52	ms
Signature length	64	64	64	64	~80	~80	bytes
On-chain record	581	656	652	651	662	—	bytes
Chain confirmation	796 ± 258	778 ± 127	743 ± 122	759 ± 126	78 ± 12	50 ± 0	ms
End-to-end	761 ± 126	785 ± 127	775 ± 121	762 ± 126	119 ± 14	52 ± 3	ms
Phone-side share	1.4	1.8	4.8	1.0	—	—	%

Table 3
Robustness: Sensitivity of chain confirmation to the batch-timeout setting and signing latency to the message length (mean ± SD)

Analysis	Level	n	Mean	SD	Median	P95
Batch timeout (ms) → confirmation	500	500	516.21	97.68	518.61	675.01
	1000	500	740.33	123.57	745.98	931.96
	2000	500	1280.89	195.49	1268.24	1613.36
Message length (bytes) → signing	64	444	3.08	0.92	2.96	4.95
	256	444	3.08	0.91	2.93	4.92
	1024	444	3.08	0.91	2.93	4.95

Confirmation latency increases approximately linearly with the batch timeout; it is 516ms at a batch timeout of 500ms and 1281ms at 2000ms, and as the chain slows down, the phone-side share also decreases because the phone-side computation remains unchanged. Signing latency is not affected by the length of the message (3.08 ms at 64, 256, and 1024 bytes, each with n = 444), as expected for a hash-then-sign construction, and the difference is within the measurement error. No confirmed-transaction failures were found in the retained data.

4. Discussion

4.1 Security discussion

Schnorr-type constructions built upon ECDLP provide existential unforgeability under the standard assumption, and saving the private key in a hardware-backed keystore reduces the risk of at-rest key extraction. However, this protection is only for the at-rest state; during signing, the key is unsealed into the application, and thus a runtime window for attack remains, as well as known attack cases involving trusted execution environments [5]. Immutability of the on-chain binding also prevents public-key substitution and supports non-repudiation in a blockchain-anchored environment [18]. Although the scheme has non-repudiation and traceability via on-chain binding, PKG key escrow is still a known trust assumption and is only partially mitigated by hardware attestation. We will not present the complete formal proof of the security argument.

4.2 Comparison with existing work

Most of the previous schemes have only been tested on development boards or servers; this work, however, has been implemented in a real smartphone secure hardware environment, a pairing-free construction to reduce on-device costs, protection of the private key via a keystore, and small on-chain data. The measured 12.5-fold signing advantage over the pairing baseline is consistent with the rationale for avoiding pairings on handsets. Compared with representative work that benchmarks IBS on phones, smartwatches, and Raspberry Pi devices [10], this paper focuses specifically on the deployment and tier selection of the smartphone hardware-backed keystore and reports where the cost of each tier actually lies. Table 4 sets out a quantitative comparison of the proposed scheme against the two baselines measured in this study.

Table 4
Quantitative comparison of the proposed scheme against the two baselines measured in this study (high-end device; values from the cleaned dataset in Table 2 and Section 3.3). A dash denotes not applicable; the baselines were run with a lighter commit setting and are reported for reference.

Dimension	Proposed (this work)	Pairing-based IBS	Software ECDSA
Construction	Pairing-free Schnorr-type IBS (ECDLP)	Pairing-based IBS	ECDSA
Pairing-free	Yes	No	Yes
Identity-based (certificate-free)	Yes	Yes	No
Security model	EUF-CMA under ECDLP (ROM)	Bilinear-pairing assumption	EUF-CMA under ECDLP
Curve / hash / security	secp256r1 / SHA-256 / 128-bit	pairing-friendly curve (baseline)	secp256r1 / SHA-256 / 128-bit
Signature length	64 B	~80 B	~80 B
Public parameters	96 B	160 B	—
Signing latency, high-end	3.08 ± 0.90 ms	38.48 ± 5.91 ms	1.88 ± 0.50 ms
Verification latency, high-end	5.13 ± 1.18 ms	33.98 ± 5.38 ms	3.63 ± 0.79 ms
Private-key storage	Hardware-backed keystore (TEE/StrongBox)	Software (baseline)	Software (baseline)
On-chain anchoring	Permissioned blockchain (identity + revocation)	Reference only	Reference only

4.3 Deployment reality and crypto-agility toward post-quantum

The two deployment facts are used for measurement. First of all, the highest-security tier is not free of charge; StrongBox adds about 29ms of unsealing time per signature, so for applications that need frequent signing, TEE may need to be used instead, as it has an on-device path of about 5ms and still protects the key at rest. Second, StrongBox is not commonly used in mid- and low-end handsets; thus, a top-tier version is unavailable, and due to necessity, a StrongBox-to-TEE-to-software fallback must be implemented. The algorithm-identifier field can be used for a smooth migration to a lattice-based scheme in the future [19], but here we only consider it migration-ready because lattice signatures are significantly larger and heavier on phones today, and we have not measured them.

4.4 Limitations and threats to validity

Given that the scale of the device and network is relatively small, extrapolation of the numbers should be done cautiously; the mid-range chain confirmation and a high variance at the TEE-H confirmation (796 ± 258 ms) indicate that end-to-end latency is sensitive to chain configuration and background load. The security argument is informal and only applies to the random-oracle model; PKG key escrow is a structural trust assumption, and because the two cross-scheme baselines used lighter commit settings, no end-to-end comparison of the schemes has been conducted. In addition, the security evaluation is analytical only: no adversarial, side-channel or fault-injection testing was performed against the runtime window in which the key is unsealed into the application, so runtime key-exposure resistance remains unverified. The measured overheads are also specific to the OEM

keystore implementations of the three test devices and may not transfer to other systems-on-chip or vendor keystores. These are the principal threats to validity.

5. Conclusion

This study shows that a pairing-free identity-based signature can be deployed inside the secure hardware of ordinary Android smartphones, moving the evaluation of mobile IBS from development boards and servers to an in-situ, keystore-aware setting. Three findings carry the main significance. First, hardware-backed protection of an IBS key is practical on commodity phones, and the Trusted Execution Environment offers the best balance of security and performance, giving practitioners a concrete basis for tier selection. Second, the cost of a higher assurance level lies not in the signature itself but in the per-signature unsealing of the secure element; because this cost is amortisable and configurable, it is a design lever rather than a barrier. Third, the end-to-end latency is governed by chain confirmation rather than handset cryptography, which relocates where future optimisation effort should be spent. Together these results advance mobile identity-based cryptography from feasibility arguments toward an engineering model for deployment. The study is nonetheless limited by a small device and network sample, by overheads that are specific to particular keystore implementations, and by the absence of a formal security proof and of adversarial security testing; these should be addressed before the numbers are generalised. Future work will build a larger measurement platform, reduce the key-operation cost inside the secure element, and investigate lightweight, phone-friendly post-quantum algorithms.

References

[1] Ngo, T. T. T., T. A. Dang, V. V. Huynh, and T. C. Le. “A Systematic Literature Mapping on Using Blockchain Technology in Identity Management.” IEEE Access 11 (2023): 26004–26032.
[2] Soltani, R., U. T. Nguyen, and A. An. “A Survey of Self-Sovereign Identity Ecosystem.” Security and Communication Networks 2021 (2021): Article 8873429.
[3] Satybaldy, A., M. S. Ferdous, and M. Nowostawski. “A Taxonomy of Challenges for Self-Sovereign Identity Systems.” IEEE Access 12 (2024): 16151–16177.
[4] Kasyoka, P., M. Kimwele, and S. M. Angolo. “Certificateless Pairing-Free Authentication Scheme for Wireless Body Area Network in Healthcare Management System.” Journal of Medical Engineering & Technology 44, no. 1 (2020): 12–19.
[5] Muñoz, A., R. Ríos, R. Román, and J. López. “A Survey on the (In)Security of Trusted Execution Environments.” Computers & Security 129 (2023): Article 103180.
[6] Blessing, J., R. Anderson, and A. Beresford. “KeyDroid: A Large-Scale Analysis of Secure Key Storage in Android Apps.” Proceedings on Privacy Enhancing Technologies 2026, no. 3 (2026): 215–235. https://doi.org/10.56553/popets-2026-0079.
[7] Shamir, A. “Identity-Based Cryptosystems and Signature Schemes.” In Advances in Cryptology – CRYPTO 1984, 47–53. Lecture Notes in Computer Science 196. Springer, 1985.
[8] Bao, Z., D. He, C. Peng, M. Luo, and K.-K. R. Choo. “An Identity-Based Adaptor Signature Scheme and Its Applications in the Blockchain System.” IEEE Open Journal of the Computer Society 4 (2023): 231–242.
[9] Wang, L., Y. Yuan, and Y. Ding. “Analysis and Design of Identity Authentication for IoT Devices in the Blockchain Using Hashing and Digital Signature Algorithms.” International Journal of Distributed Sensor Networks 2023 (2023): Article 2524051.
[10] Wahaballa, A. “Lightweight and Secure IoT-Based Payment Protocols from an Identity-Based Signature Scheme.” Electronics 11, no. 21 (2022): Article 3445.
[11] Sahana, S. C., M. L. Das, and B. Bhuyan. “A Provable Secure Key-Escrow-Free Identity-Based Signature Scheme without Using Secure Channel at the Phase of Private Key Issuance.” Sādhanā 44, no. 6 (2019): Article 132. https://doi.org/10.1007/s12046-019-1117-x.
[12] Schnorr, C. P. “Efficient Signature Generation by Smart Cards.” Journal of Cryptology 4, no. 3 (1991): 161–174.
[13] Ucbas, Y., A. Eleyan, M. Hammoudeh, and M. Alohaly. “Performance and Scalability Analysis of Ethereum and Hyperledger Fabric.” IEEE Access 11 (2023): 67156–67167.

[14] Wang, C.-J., H. Huang, and Y. Yuan. “Efficient Pairing-Free Provably Secure Scalable Revocable Identity-Based Signature Scheme.” Journal of Internet Technology 21, no. 2 (2020): 503–509.

[15] Jiang, Y., L. Deng, and B. Ning. “Identity-Based Partially Blind Signature Scheme: Cryptanalysis and Construction.” IEEE Access 9 (2021): 78017–78024.

[16] Reddy, P. V., A. R. Babu, and N. B. Gayathri. “Efficient and Secure Identity-Based Strong Key-Insulated Signature Scheme without Pairings.” Journal of King Saud University – Computer and Information Sciences 33, no. 10 (2021): 1211–1218.

[17] James, S., G. Thumbur, and P. V. Reddy. “Pairing-Free Identity-Based Proxy Signature Scheme with Message Recovery.” International Journal of Information Security and Privacy 15, no. 1 (2021): 117–137.

[18] Xu, G., and J. Dong. “A Certificateless Signcryption Mechanism Based on Blockchain for Edge Computing.” IEEE Internet of Things Journal 10, no. 14 (2023): 11960–11974.

[19] Zhang, Z., J. Zhang, Y. Yuan, and Z. Li. “Post-Quantum Secure Identity-Based Signature Scheme with Lattice Assumption for Internet of Things Networks.” Sensors 24, no. 13 (2024): Article 4188.