Internet of Things 33 (2025) 101694

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Signer-optimal multiple-time post-quantum hash-based signature for heterogeneous IoT Systems

Kiarash Sedghighadikolaei *, Attila A. Yavuz, Saif E. Nouma

Bellini College of Artificial Intelligence, Cybersecurity and Computing University of South Florida, Tampa, FL, USA

ARTICLE INFO

Dataset link: https://github.com/kiarashsedghigh/mumhors

Keywords:
Lightweight cryptography
Post-quantum security
Internet of Things (IoT)
Digital signatures
Data structures

ABSTRACT

Heterogeneous Internet of Things (IoTs) harboring resource-limited devices like wearable sensors are essential for next-generation networks. Ensuring the authentication and integrity of security-sensitive telemetry in these applications is vital. Digital signatures provide scalable authentication with non-repudiation and public verifiability, making them essential tools for IoTs. However, current NIST-PQC standards are significantly resource-intensive for practical use on constrained IoT devices. This highlights a critical need for lightweight PQ-secure digital signatures that align with the limitations of low-end IoTs.

We propose a new multiple-time hash-based signature called *Maximum Utilization Multiple HORS* (MUM – HORS) that offers PQ security, short signatures, fast signing, and high key utilization for an extended lifespan. MUM – HORS addresses the inefficiency and key loss issues of HORS in offline/online settings by introducing compact key management data structures and optimized resistance to weak-message attacks. We tested MUM – HORS on two embedded platforms (ARM Cortex A-72 and 8-bit AVR ATmega2560) and commodity hardware. Results show 40× lower resource usage at the same signing capacity (2^{20} messages, 128-bit security) than multiple-time HORS. Furthermore, MUM – HORS achieves 2× and up to 4000× faster signing than conventional secure schemes on the ARM Cortex and state-of-the-art PQ-secure schemes for IoTs, respectively.

1. Introduction

Authentication and integrity are essential for safeguarding sensitive data in next-generation networked systems, enabling the growth of the Internet of Things (IoT) across sectors like healthcare [1], military [2], and industry [3]. In healthcare, wearable devices transmit critical medical data, such as heartbeats and blood sugar levels, where compromised information can adversely affect individual health and security, underscoring the necessity for robust data protection measures.

Digital signatures provide scalable authentication with non-repudiation and public verifiability, offering a trustworthy authentication alternative for embedded medical settings [4–7]. However, traditional signatures like RSA [8] and ECDSA [9] are resource-intensive (due to operations such as modular exponentiation and elliptic curve scalar multiplication), causing performance degradation, leading to issues like frequent battery replacements in embedded medical devices [10–12]. While lightweight conventional alternatives employ a pre-computation approach [13] to reduce computation, they favor increased storage. Hence, optimizing storage and computation is vital for extending device utility.

Besides performance hurdles, emerging quantum computers can break the conventional secure signatures and their aforementioned lightweight variants via Shor's algorithm [14]. Although Yin et al. [15] and Li et al. [16] have developed efficient

* Corresponding author.
E-mail addresses: kiarashs@usf.edu (K. Sedghighadikolaei), attilaayavuz@usf.edu (A.A. Yavuz), saifeddinenouma@usf.edu (S.E. Nouma).

https://doi.org/10.1016/j.iot.2025.101694
Received 30 October 2024; Received in revised form 19 May 2025; Accepted 24 June 2025
Available online 10 July 2025

quantum digital signature schemes that reduce post-processing delays, support quantum key distribution compatibility, and enable information-theoretically secure signing, there is still a critical need for PQ-secure digital signatures that can respect the resource capabilities of low-end IoT devices.

1.1. Related work, contribution, and potential use cases

Symmetric primitives like HMAC [17] and block ciphers [18] are efficient in IoT settings [19], but lack scalability and public verifiability, which are crucial for audits [20] and non-repudiation [21]. On the other hand, while digital signatures [8,9,22] provide scalable and verifiable alternatives, they incur higher computational costs, and their deployment on resource-constrained devices can significantly reduce battery life and increase maintenance demands [10,11].

NIST has standardized three PQ-safe signatures [23,24]: lattice-based ($\mathcal{LB}$) ML-DSA [25] and FALCON [26], and hash-based ($\mathcal{HB}$) SLH-DSA [27]. ML-DSA is based on Fiat-Shamir with Abort [28] and offers a compact public key size but with larger signatures. Falcon relies on a hash-and-sign scheme using NTRU lattices [29] but necessitates more complex operations and floating-point arithmetic. The integration of PQC constructions with IoT has been explored in several studies [30]. Kumari et al. [31] proposed a delay-efficient method for polynomial multiplication in a $\mathcal{LB}$ signature, achieving a signing time of 13.299 ms. For blockchain-enabled IoT systems, Bagchi et al. [32] introduced a $\mathcal{LB}$ security framework that facilitates secure data transmission from wearable or medical devices. Regarding Space Information Networks, Ma et al. [33] developed a $\mathcal{LB}$ access authentication scheme, addressing scalability and resource constraints through a semi-aggregated signature and session key agreement. In addition, efforts like ANT [34] aim to create lightweight PQ-secure signatures by transforming $\mathcal{LB}$ one-time signatures into many-time signatures using distributed key computation under honest-but-curious assumptions. Despite these advancements, $\mathcal{LB}$ schemes remain computationally intensive and involve large key and signature sizes, limiting their practicality for resource-limited devices.

$\mathcal{HB}$ standards like XMSS [35] and SLH-DSA [27] provide strong PQ security through hash functions with minimal assumptions. They build upon Few Time Signatures (FTSs) [36–38] that permit efficient signing but only for a few messages. Stateful schemes like XMSS^{MT} rely on Merkle Hash Trees (MHT) to enable Multiple-Time Signatures (MTS) from FTS, while stateless schemes such as SLH-DSA use hypertree structures. Several studies investigate the practical integration of $\mathcal{HB}$ signatures in IoT, emphasizing lightweight, quantum-secure designs with enhanced performance via both algorithmic and hardware-level optimizations. Tandel et al. [39] proposed a stateful Merkle-based signature with XOR masking, achieving a signing time of 1–2 s on ESP32. Bos et al. [40] improved XMSS verification using Merkle traversal, attaining 1s signing on general-purpose systems. For 16-bit smart cards, Hulsing et al. [41] optimized XMSS and reduced the signing time to 105 ms through enhancements to key and tree generation. Moreover, Wang et al. [42] demonstrated a software-hardware co-design for XMSS with SHA-256 acceleration, delivering substantial speedups. Other approaches [43,44] leverage secure enclaves such as Intel SGX [45] for enhanced key management, though their applicability in IoT is limited by hardware dependencies and known vulnerabilities [46].

Instead of general-purpose tree-based approaches, an alternative MTS is a conventional-secure public-key-based online/offline model. In this approach, public keys are pre-computed and stored at the verifier, while a hash-chain strategy is used at the signer with near-optimal efficiency [47,48]. However, a straightforward transformation of FTSs such as HORS signatures [36] into MTS via such an online/offline approach is shown to be inefficient [47]. For instance, a HORS configuration with relatively short signatures leads to a key discard rate as high as 98%, even without considering weak message security [49]. These high loss rates are detrimental to the life span and practicality of the target application.

In Appendix C, we provide a more comprehensive revision of various alternative signatures and their pros and cons for low-end heterogeneous IoT applications. Overall, the state-of-the-art analysis suggests that *there is a vital need for lightweight $\mathcal{HB}$ signatures that permit multiple-time signing capability but with significantly better signer performance than existing PQ-secure alternatives*.

1.2. Our contribution

We created a new multiple-time $\mathcal{HB}$ signature referred to as *Maximum Utilization Multiple HORS* (MUM-HORS), which is specifically designed for heterogeneous IoT applications with embedded/wearable medical devices as a representative use case. MUM-HORS achieves PQ and weak-message security with fast signing and compact signatures while permitting maximum public key utilization. Central to our scheme is a storage-efficient, fixed-size 2D bitmap that facilitates efficient key derivation and management, public key management, and signature failure handling. The key features of our scheme are summarized as follows:

- *Fast Signing and Efficient Weak Message Resiliency*: The signing process of MUM-HORS is similar to efficient HORS, offering fast signing, short signatures, and lower energy consumption. Additionally, we present an optimized mitigation method for weak message attacks [49] using XOR operations, which is significantly more efficient than techniques in HORSIC [37], HORSIC+ [50], and PORS [51]. We evaluated the signing efficiency (cycles and energy consumption) of MUM-HORS on various devices, including two embedded platforms (ARM Cortex A-72 and 8-bit AVR ATmega2560) and a commodity device. Our evaluations show that on commodity hardware, MUM-HORS achieves faster signing compared to XMSS-based schemes by Tandel et al. [39] (3900×) and Ghosh et al. [52] (78×), and outperforms the $\mathcal{LB}$ scheme of Kumari et al. [31] by 600×. On the ARM Cortex-A72, it achieves a 2× improvement over conventional signatures, such as ECDSA, and 160-400× improvement over XMSS and XMSS^{MT}. These improvements are even more pronounced on the 8-bit AVR ATmega (Table 1).
- *Compact Key Management Data Structures*: Our key management data structure maintains a constant size while offering a high capacity for signature generation. For instance, the size of our data structure is bounded as low as 1.4 KB for a signing capacity of 2^{20} signatures with 128-bit security, matching state-of-the-art multiple-time schemes like XMSS [35] and [47,48]. We provide a

Table 1
Comparison of signature schemes on embedded devices and evaluation of verifier storage overhead in multi-user setting (κ = 128-bit security, $M = 2^{20}$).

Scheme	Signature Generation		Private Key (KB)	Signature Size (KB)	Verifier Storage (PK Size (KB))			PQ	ME
	8-bit AVR ATmega (cycles)	ARM Cortex A-72 (μs)			$N = 1$	$N = 8$	$N = 32$		
ECDSA [9]	79 185 664	249.021	0.031	0.046	0.062	0.496	1.984	✗	L
Ed25519 [53]	22 688 583	212.176	0.031	0.062	0.031	0.248	0.992	✗	L
SchnorrQ [54]	3 740 000	196.395	0.031	0.062	0.031	0.248	0.992	✗	L
SEMECS [47]	195 776	5.83	0.031	0.031	96 MB	768 MB	3 GB	✗	L
HORS [36]	342 976	46.8	0.031	0.78	32 GB	256 GB	1 TB	✓	L
HORSE [55]	342 976 4 979 776	46.69 682.52	790 MB 480	0.078 0.078	32	256	1 MB	✓	H
MSS [56]	5 792 000	N/A	1.438	2.295	0.016	0.124	0.496	✓	M
XMSS [35]	N/A	20 943.975	1.34	2.44	0.062	0.496	1.984	✓	H
$XMSS^{MT}$ [57]	N/A	55 099.507	5.86	4.85	0.062	0.496	1.984	✓	H
MUM − HORS	**637 376**	**129.32**	**1.43**	**0.78**	**800 MB**	**6.4 GB**	**25.6 GB**	✓	L

PQ refers to resilience against quantum attacks. ME denotes the memory expansion and code size, where (H, M, L) denote high, medium, and low, respectively. N denotes the number of users (signers) in the system. Detailed parameter selection has been given in Section 5.1.

comprehensive theoretical and numerical analysis of the compactness and practicality of our data structure for constrained devices (e.g., 8-bit AVR ATmega2560).

• *Full-fledge Implementation*: MUM−HORS full implementation is available at https://github.com/kiarashsedghigh/mumhors.

Potential Use Cases - Prioritizing Signer Optimality for Heterogeneous IoT Settings: MUM−HORS is a multiple-time signature designed for delay-tolerant and heterogeneous IoT applications that prioritizes signer efficiency and security (see Section 3). A typical application considered in state-of-the-art multiple-time signatures (e.g., [44,47,48]) is medical wearables (e.g., [58,59]) and sensory devices in digital twins [60]. In these scenarios, battery-powered IoT devices periodically measure, sign, and upload data to a cloud server. Ensuring battery longevity, minimal cryptographic overhead, and long-term PQ security is critical. To achieve this, MUM−HORS adopts an offline/online strategy, similar to prior secure multiple-time schemes [44,47,48], where pre-computed public keys are stored on a resource-rich verifier (e.g., cloud server). As detailed in Appendix C, this heterogeneous public-key model departs from general-purpose tree-based $\mathcal{HB}$ signatures (e.g., XMSS) used by Tandel et al. [39] and Ghosh et al. [52], trading signer efficiency for increased cloud storage. This trade-off is favorable, as cloud servers can store larger public keys to enhance the security and longevity of resource-constrained IoT devices. Hence, wearables using MUM−HORS can authenticate with minimal overhead, prioritizing measurement accuracy, while verifiers can support many devices without retaining numerous long-term keys.

Organization: Section 2 covers preliminaries and notation; Section 3 outlines models and use cases; Section 4 details the proposed scheme; Sections 5 and 6 present performance and security analyses; Section 7 discusses advantages and limitations; and Section 8 concludes with commercial potential and future work.

2. Notations and preliminaries

Notations: $\|$ and $|x|$ denote concatenation and the bit length of x, respectively. $x \xleftarrow{\$} S$ means x is chosen uniformly at random from the set S. $m \in \{0,1\}^*$ is a finite-length binary message. $\{q_i\}_{i=a}^{b}$ denotes $\{q_a, q_{a+1}, \dots, q_b\}$. $\log x$ is $\log_2 x$. $[1,n]$ denotes all integers from 1 to n. $f : \{0,1\}^* \rightarrow \{0,1\}^L$ and $H : \{0,1\}^* \rightarrow \{0,1\}^L$ denote one-way and cryptographic hash functions, respectively. $a * b$ and $a \overset{q}{*} b$ mean $a = a * b$ and $a = (a * b \mod q)$, where $*$ is an arbitrary operation.

Definition 1. A $\mathcal{HB}$ digital signature SGN consists of three algorithms:

- $(sk, PK, I_{SGN}) \leftarrow \texttt{SGN.Kg}(1^\kappa)$: Given the security parameter κ, it outputs the private key sk, the public key PK, and the system-wide parameters I_{SGN}.
- $\sigma \leftarrow \texttt{SGN.Sig}(sk, m)$: Given the sk and message m, it returns a signature σ.
- $b \leftarrow \texttt{SGN.Ver}(PK, m, \sigma)$: Given PK, message m, and its corresponding signature σ, it returns a bit b, with $b = 1$ meaning valid, and $b = 0$ otherwise.

Definition 2. Hash to Obtain Random Subset HORS [36] is a $\mathcal{HB}$ digital signature consists of three algorithms:

- $(sk, PK, I_{\texttt{HORS}}) \leftarrow \texttt{HORS.Kg}(1^\kappa)$: Given the security parameter κ, it selects $I_{\texttt{HORS}} \leftarrow (t, k, l)$, generates t random l-bit strings $\{s_i\}_{i=1}^{t}$, and computes $v_i \leftarrow f(s_i), \forall i = 1, \dots, t$. Finally, it sets $sk \leftarrow \{s_i\}_{i=1}^{t}$ and $PK \leftarrow \{v_i\}_{i=1}^{t}$.
- $\sigma \leftarrow \texttt{HORS.Sig}(sk, m)$: Given sk and m, it computes $h \leftarrow H(m)$ and splits h into k $\log t$-sized substrings $\{h_j\}_{j=1}^{k}$ and interprets them as integers $\{i_j\}_{j=1}^{k}$. It outputs $\sigma \leftarrow \{s_{i_j}\}_{j=1}^{k}$.
- $b \leftarrow \texttt{HORS.Ver}(PK, m, \sigma)$: Given PK, m, and σ, it computes $\{i_j\}_{j=1}^{k}$ as in HORS.Sig(.). If $v_{i_j} = f(\sigma_j), \forall j = 1, \dots, k$, it returns $b = 1$, otherwise $b = 0$.

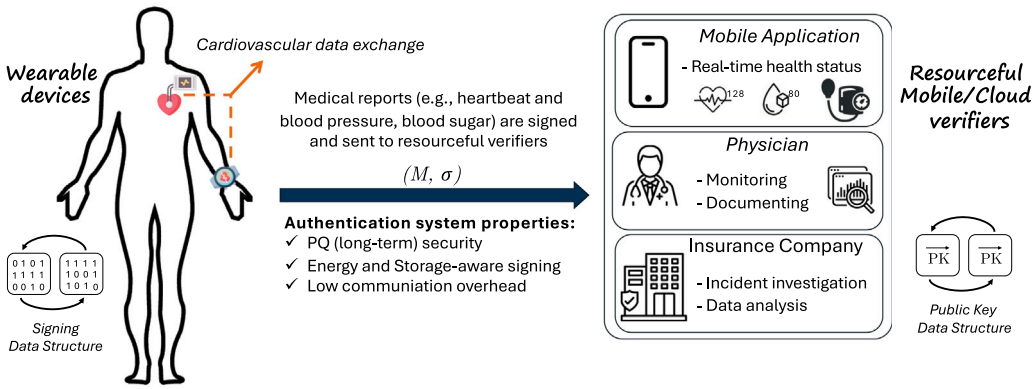

Fig. 1. MUM-HORS system model for a resource-limited wearable medical IoT use-case.

Definition 3. Let $\mathcal{H} = \{H_{i,t,k,L}\}$ be a function family indexed by i, where $H_{i,t,k,L}$ maps an arbitrary length input to a L-bit subset of k elements from the set $\{0, 1, \ldots, t-1\}$. $\mathcal{H}$ is r-subset (RSR) and second-preimage resistant (SPR), if, for every probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ running in time $\leq T$:

$$\begin{aligned}\text{InSec}_{\mathcal{H}}^{RSR}(T) = \max_{\mathcal{A}}\{&\Pr[(M_1, M_2, \ldots, M_{r+1}) \leftarrow \mathcal{A}(i,t,k) \\ &\text{s.t. } H_{i,t,k,L}(M_{r+1}) \subseteq \bigcup_{j=1}^{r} H_{i,t,k,L}(M_j)]\} < negl(t,k)\end{aligned} \tag{1}$$

$$\begin{aligned}\text{InSec}_{\mathcal{H}}^{SPR}(T) = \max_{\mathcal{A}}\{&\Pr[x \leftarrow \{0,1\}^*; x' \leftarrow \mathcal{A}(x) \text{ s.t. } x \neq x' \\ &\text{and } H_{i,t,k,L}(x) = H_{i,t,k,L}(x')]\} < negl(L)\end{aligned} \tag{2}$$

3. Models and use cases

System Model and Use-case: We assume a traditional public-key-based authentication model for heterogeneous IoT-cloud applications, wherein low-end IoT devices gather information in a store-and-forward manner for a delay-tolerant setting. Embedded healthcare devices such as pacemakers and implantable devices are prominent examples [61,62], in which the medical sensor takes continuous measurements (e.g., heart rate), digitally signs them, and then periodically uploads the telemetry and corresponding signatures into a cloud server for analysis. It is noted that other miscellaneous applications operate in similar settings, such as some smart cities [63] and drone services [64].

In our target use case involving wearable medical devices, the longevity, security, and efficiency of the low-end device are of primary concern. Specifically, our digital signature scheme focuses on energy-efficient computation (battery life and processing limits) and storage on resource-limited signers (e.g., 8-bit microcontrollers). Meanwhile, resourceful verifiers (e.g., cloud servers) manage public key storage and message authentication. Our system model is given in Fig. 1.

Threat and Security Model: We consider a Probabilistic Polynomial Time (PPT) adversary equipped with quantum computational capabilities. The adversary can execute passive (e.g., network traffic monitoring) and active (e.g., message interception and modification, generation of forged message-signature pairs) attacks. The security model of our proposed digital signature scheme captures our threat model and follows the Existential Unforgeability under Chosen Message Attacks ($EU\text{-}CMA$), is defined as follows:

Definition 4. The $EU\text{-}CMA$ experiment for SGN is defined as follows:

- $(sk, PK, I_{\text{SGN}}) \leftarrow \text{SGN.Kg}(1^{\kappa})$, $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\text{SGN.Sig}_{sk}(.)}(PK, I_{\text{SGN}})$
- $\mathcal{A}$ wins the experiment after sending q queries if $1 \leftarrow \text{SGN.Ver}(PK, m^*, \sigma^*)$ and m^* was not queried to the signing oracle $\text{SGN.Sig}_{sk}(.)$.

$$Succ_{\text{SGN}}^{EU\text{-}CMA}(\mathcal{A}) = Pr[Expt^{EU\text{-}CMA}(\mathcal{A}) = 1] \tag{3}$$

$$InSec_{\text{SGN}}^{EU\text{-}CMA}(T) = \max_{\mathcal{A}}\{Succ_{\text{SGN}}^{EU\text{-}CMA}(\mathcal{A})\} < negl(T) \tag{4}$$

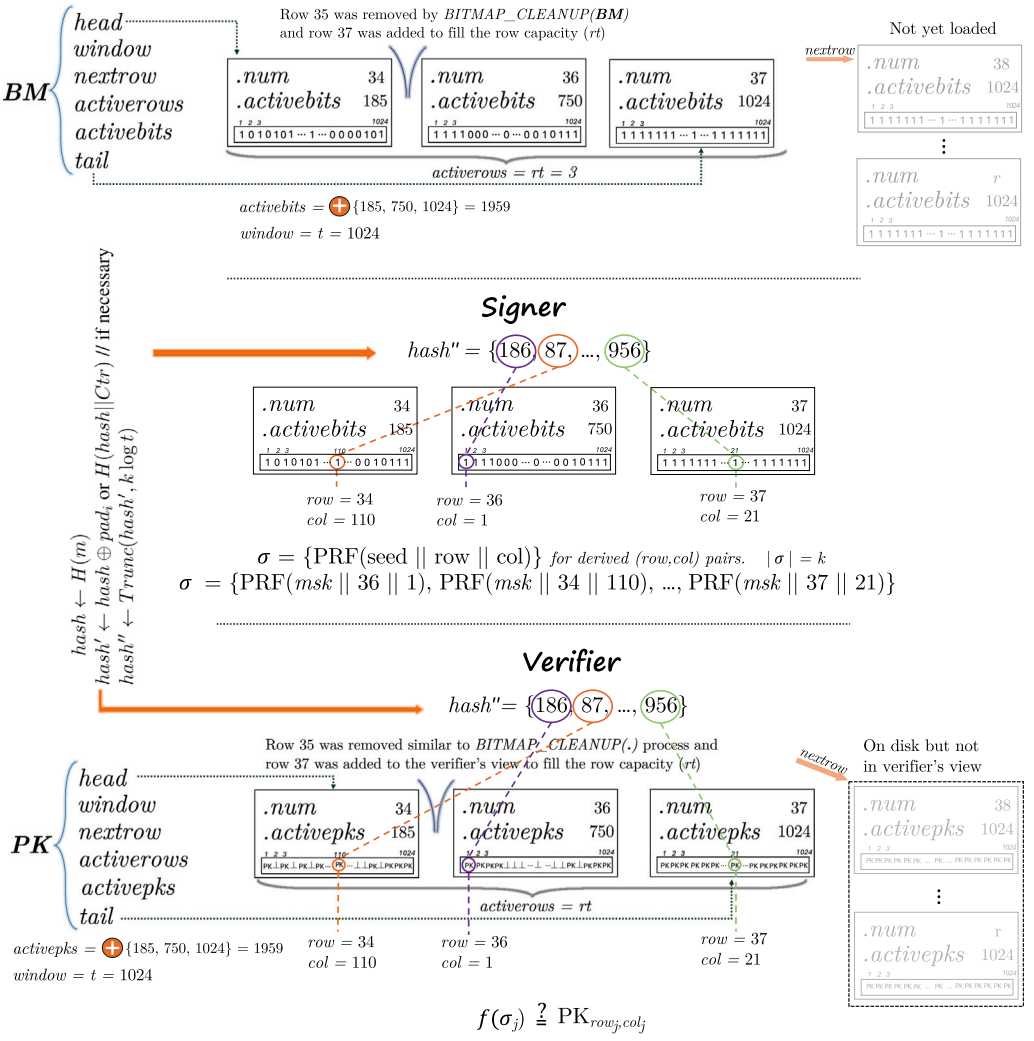

Fig. 2. MUM-HORS's internal structure using bitmap data structure ($rt = 3$).

4. Proposed scheme

Main Idea: We propose MUM-HORS, a multiple-time hash-based signature scheme, which addresses severe key utilization limitations of HORS for multiple-time settings while ensuring maximum signer efficiency and weak message security. MUM-HORS follows public-key offline-online model, in which public keys are pre-computed from a cryptographic hash chain approach (offline) and maintained at the verifier side, while the signer efficiently generates a signature from a master key (as in [47]). However, as discussed in Section 1 and shown in [47], this approach results in extremely large public keys and wastes individual components in public keys, because HORS selects small values for k to optimize signing efficiency and minimize signature size. However, following the offline model results in the waste of significant $t - k$ keys. For instance, for the 128-bit security parameter setting ($t = 1024$, $k = 25$), only 2.44% of public keys are effectively utilized, greatly degrading the authentication usability and verifier storage. Moreover, HORS is susceptible to weak message attacks [49] due to potential repetition in the k index values derived during signing from the message's hash value, violating r-subset-resiliency (see Section 2).

To address the outlined challenges, MUM-HORS incorporates a storage-efficient two-dimensional bitmap (BM) with rt rows, each containing a t-bit list and metadata for correctness and optimization. During the signing, the first t set bits indicate available keys, with row and column indices aiding key derivation using the master key. Marked keys are used, while the remaining $t - k$ keys stay unchanged, and depleted rows are replaced with new ones to maintain key availability. Moreover, MUM-HORS includes an efficient weak key mitigation strategy (consisting of XOR operations) through algorithm design and parameter setup.

Outline of MUM-HORS Algorithms: Fig. 2 illustrates the internal structure of MUM-HORS and its integration with BM ($rt = 3$ for demonstration), Algorithm 1 outlines MUM-HORS, and Algorithm 2 illustrates BM operations as below:

Algorithm 1 Maximum Utilization Multiple HORS (MUM-HORS)

$(sk, PK, \texttt{BM}, I_{\texttt{MUM-HORS}}) \leftarrow \texttt{MUM} - \texttt{HORS.Kg}(1^\kappa)$:

Set $I_{\texttt{MUM-HORS}} \leftarrow (I_{\texttt{HORS}}, r, rt, pad_i \xleftarrow{\$} \{0,1\}^\kappa, i \in [1,3])$, $msk \xleftarrow{\$} \{0,1\}^\kappa$
Initialize bitmap $\texttt{BM} = \{row_i\}_{i=1}^{rt}$ with bitmap and row-specific variables
Create public keys $PK \leftarrow \{pk_i\}_{i=1}^{r}$ as an array of rows and initialize PK and row-specific variables similar to the BM. Each row of PK is set as $pk_i \leftarrow \{f(PRF(msk||i||j))\}_{1 \le j \le t}$
Store the private key $sk \leftarrow (msk)$ and BM on the signer and the public keys PK on the verifier
return $(sk, PK, \texttt{BM}, I_{\texttt{MUM-HORS}})$

$\sigma \leftarrow \texttt{MUM} - \texttt{HORS.Sig}(sk, m)$: Set $Ctr = 1$ and $hash \leftarrow Trunc(H(m), k \log t)$

Split $hash$ into k $\log t$-bit substrings $\{hash_j\}_{j=1}^{k}$ and interpret each as an integer $\{i_j\}_{j=1}^{k}$
if there exists a duplicate index i_j **then**
    $hash \oplus= pad_i$ and **goto** 1, $i \in [1,3]$
else goto 8
Split $hash' \leftarrow Trunc(H(hash||Ctr), k \log t)$ as step 1 into integer indices $\{i_j\}_{j=1}^{k}$
if there exists a duplicate index i_j **then**
    Ctr += 1 and **goto** 5
for $j = 1 \dots k$ **do**
    $\{(r_j, c_j)\} \leftarrow$ BITMAP_GET_ROW_COLUMN(BM, i_j)
return $\sigma \leftarrow (\{PRF(sk||r_j||c_j)\}_{j=1}^{k}, Ctr)$

Signer is Idle

for $j = 1 \dots k$ **do**
    BITMAP_UNSET_INDICES(BM, i_j)
if BITMAP_EXTEND_MATRIX(BM) = `False` **then abort** ▷ No more private keys to sign

$b \leftarrow \texttt{MUM} - \texttt{HORS.Ver}(PK, m, \sigma)$: Set $hash \leftarrow Trunc(H(m), k \log t)$

if $hash$ or any $pad_{1,2,3}$ yield distinct i_j as steps 1-4 of $\texttt{MUM} - \texttt{HORS.Sig}(.)$, **then goto** 5
Split $hash' \leftarrow Trunc(H(hash||Ctr), k \log t)$ into integer indices $\{i_j\}_{j=1}^{k}$
if there exists a duplicate index i_j **then**
    Execute steps 5-7 of $\texttt{MUM} - \texttt{HORS.Sig}(.)$ and **return** $b = 0$
for $j = 1 \dots k$ **do**
    Retrieve public key pk_{i_j} for every index i_j similar to BITMAP_GET_ROW_COLUMN(.)
    if $pk_{i_j} \neq f(\sigma_j)$ **then**
        return $b = 0$
return $b = 1$

Verifier is Idle

Invalidate $\{PK_{i_j}\}_{j=1}^{k}$ as in BITMAP_UNSET_INDICES(.)
Extend the view of the public keys similar to BITMAP_EXTEND_MATRIX(.)

Algorithm 2 Bitmap Manipulation Algorithms

$b \leftarrow$ BITMAP_EXTEND_MATRIX(BM):

if Not enough bits (BM.*window*) exists in the bitmap **then**
    if No available rows exist **then return** `False`
    if BITMAP_CLEANUP(BM) == 0 **then**
        Iterate over the rows and set i to the index of the row with the minimum *.activebits*
        Update BM.*activerows* −= 1, BM.*activebits* −= BM[i].*activebits*, and REMOVE_ROW(i)
    Fill the bitmap to its full capacity (BM.*rt*) by adding new rows (fresh row-internal parameters). Update bitmap variables BM.*tail* to last row, BM.*activebits* with newly added bits, and increment BM.*nextrow* by one
return `True`

$n \leftarrow$ BITMAP_CLEANUP(BM): Set $cleaned = 0$

Iterate over the rows $BM[i]$ and **if** BM[i].*activebits* == 0 **then** REMOVE_ROW(i) and $cleaned$ += 1
return number of deleted rows $cleaned$

$(row, col) \leftarrow$ BITMAP_GET_ROW_COLUMN(BM, $index$):

Find the row $BM[i]$ that contains the $(index+1)^{th}$ set bit of the bitmap and set $colidx$ to the index of it in $BM[i]$
return row and column indices of the requested bit as (BM[i].*num*, $colidx$)

BITMAP_UNSET_INDICES(BM, **indices**):

for $index$ **in indices do**
    Find the row $BM[i]$ that contains the $(index+1)^{th}$ set bit of the bitmap and then set it to 0 (Unsetting)
    Update bitmap and row variables as: BM.*activebits* −= 1 and BM[i].*activebits* −= 1

REMOVE_ROW($index$):

if $index$ == BM.*head* **then** BM.*head* = BM.*head* $\overset{rt}{+=}$ 1
else if $index$ == BM.*tail* **then** BM.*tail* $\overset{rt}{-=}$ 1
else
    Shift rows from BM.*head* if $index$ is closer; otherwise shift from BM.*tail*

The MUM − HORS.Kg(.) initializes system parameters $I_{\text{MUM-HORS}}$, including the HORS and BM parameters (r, rt), representing the total and maximum bitmap rows, respectively. It generates the master key msk and three random pads (Step 1). An rt-row bitmap is created as a circular queue (with middle node deletion support) with rt cells, initializing global parameters $head$ (first row's index), $tail$ (last row's index), $window$ (window size in bits = t), $nextrow$ (next row number), and $activerows$ (current number of rows). Row-specific parameters include num (current row number), $activebits$ (number of active bits in the row), and $bits[.]$ (the t-bit list) (Step 2). Public keys are generated by concatenating msk with each bit's row and column number (Step 3). The public key PK shares the bitmap parameters to ensure synchronization. The private key sk and BM are stored on the signer, while PK is stored on the verifier (Step 4).

The MUM − HORS.Sig(.) first checks if the message's hash produces distinct indices. If not, the hash is XORed with three pads; signing proceeds if any pad is effective (Steps 2–3). If none work, a counter Ctr is concatenated with the hash and iteratively hashed until k distinct $\log t$-sized indices are obtained (Steps 5–7). Note that the hash output is truncated to $k \log t$ bits to ensure κ-bit security (Steps 0 and 5). Once k distinct indices are identified, row and column indices of the first $window$ set bits are retrieved from the BM using BITMAP_GET_ROW_COLUMN(.) (Steps 8–9) to reconstruct the private keys as in key generation. The BITMAP_GET_ROW_COLUMN(.) (Algorithm 2) iterates over the bitmap rows to locate the $(index + 1)^{th}$ set bit and returns its row and column indices. Finally, the signature is generated (Step 10).

When the signer is idle after sending a signature, it unsets the derived k indices in the bitmap using BITMAP_UNSET_INDICES(.) (Algorithm 2) by locating each index's position in the bitmap, setting the corresponding bit to 0, and updating the bitmap's global and row parameters (Steps 1–3). Separating unsetting from index retrieval was a performance-driven decision. After updating, the signer checks if there are enough private keys for future messages by invoking BITMAP_EXTEND_MATRIX(.) (Algorithm 2) (Step 13). This function extends the matrix if fewer than $window$ active bits are present. It first checks for and removes empty rows using BITMAP_CLEANUP(.) (Steps 2–3), which removes rows with zero active bits and returns the number of removed rows. If no rows are empty, the row with the fewest active bits is removed (Steps 4–5). Additional rows are then added to fill the BM to its full capacity (rt rows) (Step 6). If no extension was needed or it was successful, it returns True (Step 7).

MUM − HORS.Ver(.) first verifies that the message's hash and pads produce distinct indices (Step 1). If not, it checks the Ctr and, in the case of an invalid Ctr, the verifier rejects the signature and computes a valid Ctr to update the public key storage (Steps 3–4). It then retrieves the public key corresponding to the $i_j{}^{th}$ public key within the first $window$ public keys. If any public key is rejected, the signature is deemed invalid; otherwise, the signature is accepted (Steps 5–9). Finally, during idle periods, the verifier removes public keys from storage, similar to BITMAP_UNSET_INDICES(.), and if additional public keys are required, the verifier follows the BITMAP_EXTEND_MATRIX(.) procedure (Steps 10–11). To address potential resynchronization (a common challenge in MTSs [65]) between the signer and verifier caused by transmission noise or adversarial corruption, we propose a mitigation algorithm (see Appendix B) that helps with state recovery.

5. Performance analysis and comparison

This section analyzes the performance of MUM-HORS and its counterparts, focusing on signer storage overhead, key and signature sizes, and signing and verification times, with key generation occurring offline. We discuss the optimal selection of the row threshold (rt) in the bitmap and evaluate the computational complexity of each bitmap operation. Our analysis focuses on the implementation of BM as a circular queue with support for middle node deletion. An alternative linked-list version, detailed in Appendix A, enhances performance but requires additional memory for storing each node's successor address.

In the following, we first present the experimental setup and analytical parameter selection of our proposed MUM-HORS and our selected counterparts in Section 5.1. Section 5.2 provides a detailed performance evaluation on both commodity hardware and our selected embedded platforms.

5.1. Experimental setup and parameter selection

The evaluation of MUM-HORS and its counterparts was conducted with a 128-bit security parameter on 32-byte messages, as shown in Tables 1–3.

Hardware and Software Configurations: We used three types of devices for our evaluations. First, we used a desktop with an Intel i9-11900K @ 3.5 GHz processor and 64 GB of RAM to evaluate the signature generation and verification performance. Second, to demonstrate the signature generation performance of MUM-HORS for low-end (embedded) IoT settings, we used an 8-bit ATmega2560 @ 16MHz microcontroller with 256 KB flash memory, 8 KB SRAM, and 8 KB EEPROM, as well as a Raspberry Pi 4 with a Quad-core Cortex-A72 @ 1.8 GHz and 8 GB of RAM. For the one-way and cryptographic hash functions (i.e., f and H), we choose Blake2[1] due to its high efficiency on commodity hardware and low-end embedded devices.

MUM-HORS's Parameter Selection and Tuning: For MUM-HORS, the parameters relationship can be derived from the bitmap's 2D structure, with M as the total number of messages to be signed and r as the total required number of rows:

$$M = (r - 1) \cdot \frac{t}{k} + 1 \tag{5}$$

[1] https://github.com/BLAKE2/

BM loads rt rows at a time, each containing t-bit vector, along with metadata (row number and the number of active bits). Therefore, the size of the bitmap is:

$$|\mathtt{BM}| = rt \cdot (t + \log t + \log r) \text{ bits} \tag{6}$$

The parameter rt affects the bitmap size and private key usage (as well as the number of signable messages). Increasing rt raises the chance of having rows with no active bits, as the remaining $t - k$ bits are distributed across more rows before invoking BITMAP_EXTEND_MATRIX(.). This reduces key loss by permitting the replacement of empty rows through BITMAP_CLEANUP(.) rather than removing the row with fewer active bits. For example, with parameters $(t = 1024, k = 25, l = 256, r = 25601, rt = 11, M = 2^{20})$, the private key usage is 100%, allowing all messages to be signed. However, setting $rt = 8$ results in a loss of 10,536 bits and a reduction of 463 signable messages experimentally.

Optimal Values for rt: To derive the optimal bound for rt, we analyze the bitmap when the total number of bits is t, just before allocating a new row. The aim is to ensure that, with high probability, at least one row has fewer than k active bits (ideally close to zero) so that key loss is minimized during row replacement when selecting k bits among the rows. We formally define the problem as follows:

Question 1: Given rt rows and t bits uniformly distributed among them, what is the minimum rt such that, with high probability, at least one row has a maximum load of k bits or less?

To answer this, we translate the problem into the Balls in Bins problem [66]. Specifically, given m balls and n bins, we aim to determine the maximum load in a bin with high probability after uniformly distributing the m balls. For this purpose, we apply Theorem 1 from [66]:

Theorem 1. *Let M be the random variable that counts the maximum number of balls in any bin. We throw m balls independently and uniformly at random into n bins. Then with high probability, $M > load_{max}$ and we have:*

$$load_{max} = \frac{m}{n} + \sqrt{2\frac{m\log n}{n}\left(1 - \frac{1}{\alpha}\frac{\log\log n}{2\log n}\right)}, \text{if } m \gg n \cdot (\log n)^3, 0 < \alpha < 1 \tag{7}$$

In the above context, m balls correspond to t bits, n bins correspond to rt BM rows, and α is the smoothing parameter. Increasing α provides a more conservative estimate of the maximum load, ensuring that, with high probability, the maximum load is almost the estimated bound. In essence, a higher α shifts the uniform ball distribution $\frac{m}{n}$ towards $load_{max}$. While this approach determines an upper bound on the maximum load, we are interested in having $\leq k$ bits in at least one row with high probability (minimizing the load). Thus, we define:

Dual of Question 1: Given a bitmap with $rt \cdot t$ bits, where t bits are present, and $rt \cdot t - t$ bits are marked as used, minimizing the number of unmarked bits in at least one row is equivalent to maximizing the number of marked bits in at least one row. What should rt be to ensure that, with $rt \cdot t - t$ marked bits added, the maximum load in at least one row is $t - k$ or ideally t?

To solve this, we set $m = rt \cdot t - t$, $n = rt$ and $load_{max} = t - k$ and we get:

$$t - k = \frac{rt \cdot t - t}{rt} + \sqrt{2\frac{(rt \cdot t - t)\log rt}{rt}\left(1 - \frac{1}{\alpha}\frac{\log\log rt}{2\log rt}\right)}, \quad \text{if } rt \cdot t - t \gg rt \cdot (\log rt)^3, \quad 0 < \alpha < 1 \tag{8}$$

We analyze the parameters by implementing a numerical solver (in MATLAB, available in our code repository[2]). With parameters $t = 1024$ and $k = 25$ and setting $\alpha = 0.999$, our solver outputs a row threshold of 10.903, assuming a maximum load of $t - k$. To increase the likelihood of full depletion, assuming $load_{max} = t$, the solver outputs a safer margin of 13.94 rows. For evaluations, the MUM-HORS parameters are set as $(t, k, l, r, rt) = (1024, 25, 256, 25601, 11)$.

Counterparts' Parameter Selection: The schemes HORS, HORSE, and HORST use parameters $(t, k, l) = (1024, 25, 256)$, with HORSE additionally defined by $d = 25290$, computed as $d = M \cdot (1 - e^{-k/t})$. HORSE can reduce memory consumption by increasing the number of hash function evaluations, applying Jakobsson's fractal traversal technique [67], which trades signing time for a more compact private key. XMSS is instantiated as XMSS-SHA2_20_256, and XMSS^{MT} as XMSSMT-SHA2_20/2_256. To our knowledge, there are no existing performance benchmarks for XMSS or its multi-tree variants on 8-bit microcontrollers. Other schemes follow parameterizations from their respective literature.

5.2. Efficiency evaluation and comparison

The performance evaluation of MUM-HORS on commodity hardware is presented in Table 2, while its performance on embedded platforms is detailed in Table 1, with corresponding energy consumption analysis provided in Table 3.

Signer and Verifier Memory Usage: The MUM-HORS signer requires just 1.43 KB of storage, comprising a 256-bit master key and a 1.4 KB bitmap, representing a significant improvement over HORSE (300×). While its storage is comparable to that of MSS and

[2] https://github.com/kiarashsedghigh/mumhors/blob/main/Optimizations/row_threshold.m

Table 2
Comparison of PQ-secure signature schemes on commodity hardware and the evaluation of verifier storage overhead in multi-user setting (κ = 128-bit security, $M = 2^{20}$).

Scheme	Signature Generation (µs)	Signature Size (KB)	Private Key (KB)	Signature Verification (µs)	Verifier Storage (PK Size (KB))		
					$N = 1$	$N = 8$	$N = 32$
Tandel et al. [39]	68 571.42	4.59	4.25	10 285.71	5.31	42.48	169.92
Ghosh et al. [52]	1375.47	2.93	2.04	1375.47	0.062	0.496	1.984
Kumari et al. [31]	11 019.17	1.4	886.42	609.13	1.8 MB	14.4 MB	57.6 MB
Bos et al. [40]	40 000	2.44	1.34	1874.28	0.062	0.496	1.984
HORS [36]	6.12	0.78	0.031	6.35	32 GB	256 GB	1TB
HORSE [55]	6.19 86.25	0.78	790 MB 480	6.41 6.46	32	256	1 MB
HORST [68]	614.38	8.59	0.031	82.81	32 MB	256 MB	1 GB
MUM − HORS	**17.56**	**0.78**	**1.43**	**19.17**	**800 MB**	**6.4 GB**	**25.6 GB**

XMSS, it is 4× more efficient than XMSS^{MT}. Compared to XMSS-based schemes by Tandel et al. [39] and Ghosh et al. [52], MUM−HORS achieves at least 1.4× and 3× greater storage savings, respectively. Against the $\mathcal{LB}$ scheme by Kumari et al. [31], MUM−HORS is 600× more compact. While not smaller than traditional full-time signatures, it remains comparable to Falcon-512 and up to 1.7× smaller than ML-DSA among NIST-PQC candidates.

Verifier storage is where MUM−HORS shows a major advantage over its efficient counterpart, HORS. In single-user settings, it requires 40× less public key storage, with this benefit scaling further as the number of users increases (e.g., $N = 8$, $N = 32$). Although MUM−HORS requires more verifier storage than conventional schemes like ECDSA and tree-based schemes by Tandel et al. [39] and Ghosh et al. [52], it offers a favorable tradeoff with significantly improved signature generation speed and smaller signature size.

Communication Bandwidth Overhead: Smaller k values in MUM−HORS enable reduced signature size with fewer private keys. MUM−HORS's signature size is comparable to HORS and HORSE, and notably shorter than MSS and XMSS (3×), XMSS^{MT} (6×), and HORST (11×). Moreover, MUM−HORS offers 3-6× smaller signature than schemes [39,52] and Bos et al. [40], and 1.7× smaller than the $\mathcal{LB}$ scheme by Kumari et al. [31]. Compared to NIST-PQC standards, it is comparable to Falcon-512, 3× smaller than ML-DSA, and 21× smaller than SLH-DSA.

Signing on the Commodity Device and Verification: Compared to HORS and HORSE, MUM−HORS incurs a 3× higher signing time due to bitmap operations. Specifically, BITMAP_GET_ROW_COLUMN(.), active during signing, takes 0.23 µs per index on commodity hardware, while BITMAP_CLEANUP(.) and BITMAP_UNSET_INDICES(.) take 0.027 µs and 0.18 µs, respectively. Notably, BITMAP_EXTEND_MATRIX(.) is invoked conditionally, depending on the presence of $window$ bits in the BM. Despite this overhead, MUM−HORS achieves at least 80× faster signing compared to schemes [39,40,52], and 2000× faster than Kumari et al. [31]. In verification, MUM−HORS performs 3× slower compared to HORS and HORSE, while being 4× faster than HORST. It also achieves 32× faster verification than [31], and 500-700× faster than [39,52].

Signature Generation on Embedded Devices: The efficiency of MUM−HORS on embedded platforms is summarized in Table 1. On ARM Cortex-A72, MUM−HORS signing is 2.8× slower than HORS and HORSE due to bitmap operations, which enable a 41× reduction in public key size. Specifically, BITMAP_CLEANUP(.) takes 0.6 µs, BITMAP_UNSET_INDICES(.)0.89µs per index, and BITMAP_GET_ROW_COLUMN(.)1.53µs. Despite this, MUM−HORS outperforms XMSS and XMSS^{MT} by 150× and 400×, respectively, and exceeds conventionals by 1.8×. Greater gains can be observed on the 8-bit AVR ATmega.

Energy Impact and Cost of Signing on Embedded devices: In Table 3, we present the analysis of the impact of signature generation on energy consumption and battery life in 8-bit AVR microcontrollers using the energy estimation model from [69] based on the MICAz sensor node. The MICAz node features an ATmega128L microcontroller (16 MHz, 4 KB EEPROM, 128 KB flash memory) and a ZigBee 2.4 GHz radio chip (CC2420). It is powered by AA batteries with a maximum energy capacity of 6750J. According to [69], the ATmega128L MCU consumes 4.07nJ per cycle, while the CC2420 transceiver chip draws 0.168μJ per bit transmitted. Using these results, we estimated the energy consumption for signing and transmission for MUM−HORS and its counterparts assuming one signature is generated and transmitted every 10 s or every 1 min. Note that the battery lifespan denotes the expected operation time of the IoT device. We assume the device is executing cryptographic operations only.

The signing capability depends on both signature generation and transmission efficiency. Notably, MUM−HORS can support a larger number of signatures (2^{20}) than the practical limit of signing operations achievable on a MICAz node ($\approx 2 \cdot 10^6$). This translates into a longer battery lifespan to operate in autonomous and challenging environments. Our $\mathcal{HB}$ counterpart, the Merkle Signature Scheme (MSS) [56], supports only $\frac{1}{14}$ of the signatures achievable with MUM−HORS with 2.9× larger signature size. MSS uses Winternitz One-Time Signature (WOTS) and constructs a Merkle tree, as in XMSS [35], leading to higher signing costs due to the WOTS signature process and path computation in the tree. Our second $\mathcal{HB}$ counterpart, HORS, offers 1.48× more signatures than MUM−HORS, but at the cost of a significantly larger public key, approximately 40× larger. This has been demonstrated to be impractical on low-end verifiers when the number of sensory devices increases. The elliptic-curve-based SEMECS can generate more signatures than the MICAz node's practical limit. However, it does not provide post-quantum security. MUM−HORS offers the optimal balance between maximum signing capability, post-quantum security, and compact key sizes, directly translating into longer battery life and optimal verifier storage.

Table 3
Energy usage of signature generation and transmission on an AVR ATMega2560 MCU.

Scheme	Signature Generation		Signature Transmission		Energy Cost (mJ)	Max Signing Operations	Battery Lifespan	
	Time (cycles)	Energy Cost (mJ)	Sig Size (KB)	Energy Cost (mJ)			Freq = 10 s	Freq = 1 min
ECDSA [9]	79 185 664	332.285	0.046	0.065	332.35	20 316	0 y 2 d	0 y 14 d
Ed25519 [53]	22 688 583	92.343	0.062	0.086	92.429	73 063	0 y 8 d	0 y 50 d
SchnorrQ [54]	3 740 000	15.222	0.062	0.086	15.308	440 946	0 y 51 d	0 y 306 d
SEMECS [47]	195 776	0.797	0.031	0.043	0.84	8 035 714	2 y 200 d	15 y 105 d
HORS [36]	342 976	1.396	0.78	1.075	2.471	2 731 688	0 y 316 d	5 y 72 d
[a]MSS [56]	5 792 000	23.573	2.295	29.964	53.537	126 081	0 y 144 d	2 y 359 d
MUM − HORS	**637 376**	**2.594**	**0.78**	**1.075**	**3.669**	**1 839 738**	**0 y 283 d**	**4 y 332 d**

[a] The maximum MSS signing capability is 2^{16} due to EEPROM write/erase endurance limits.

6. Security analysis

Theorem 2. *MUM−HORS is EU-CMA secure if $H()$ is r-subset-resilient and second-preimage resistant:*

$$InSec^{EU\text{-}CMA}_{\text{MUM−HORS}}(T) = InSec^{RSR}_{H}(T) + InSec^{SPR}_{H}(T) < negl(t, k, L)$$

Proof. Given a set of adaptively chosen and queried q valid message-signature pairs $\{(m_i, \sigma_i)\}_{i=1}^{q}$, there are the below cases where the adversary $\mathcal{A}$ can forge a signature:

- *$\mathcal{A}$ breaks r-subset-resilient of H*: The $\mathcal{A}$ identifies a message m^* such that its k distinct elements (as in MUM − HORS.Sig(.)) are among the observed $q \cdot k$ elements from the last q messages, with a success probability of $(\frac{q \cdot k}{t})^k$. We note that our efficient mitigation method against weak message attacks ensures the derivation of k distinct indices from a message, either by using its initial hash XORed with random pads secured by long-term certificates on the verifier, or through an iterative procedure using an incremental counter Ctr.
  The success probability $(\frac{q \cdot k}{t})^k$ decreases to $(\frac{k}{t})^k$ due to the bitmap design. Once the signer selects and uses k bits from the first window, they are marked as used, preventing reuse in future rounds. Uniqueness is maintained by the row parameter *num*, ensuring each bit's combination $(msk \| row \| col)$ is distinct. Even when BITMAP_EXTEND_MATRIX(.) is invoked, new rows have unique row numbers, enforced by the global *nextrow* parameter. Moreover, the verifier maintains identical global and row parameters to manage public keys, ensuring synchronization with the signer. This guarantees that once a private key is used, the corresponding public key is invalidated during verification, preventing reuse.
  In summary, since the bitmap replaces k out of t private keys, and the remaining $t - k$ unused keys are independent and hidden from the attacker, the success probability per round is reduced to that of HORS as an OTS, which is negligible for suitable k and t. This probability of forging a signature on HORS, in addition to $(\frac{q \cdot k}{t})^k$, entails the likelihood of inverting the one-way function $f(.)$ to derive private keys from the verifier's stored public keys. Grover's algorithm [70] can reverse a black-box function with input size N in $O(\sqrt{N})$ steps and $O(\log_2 N)$ qubits. For L-bit output $f(.)$, the probability of reverting k public keys is $2^{-k \cdot \frac{L}{2}}$.
- *$\mathcal{A}$ breaks the second-preimage resistance of H*: We evaluate the security of our hash function $H(.)$ using Grover's model. The attacker could find m^* such that $H(m_i) = H(m^*)$ and produce a valid (m^*, σ_i). Given identical hashes, the k indices for m^* will match those for m_i by any method (initial hash, pads, or Ctr). For an $L = 256$-bit hash function like Blake2-256, the collision probability is $\frac{1}{2^{\frac{L}{2}}}$, providing 128-bit security, which is negligible. Moreover, the parameters k and t impact the security of $H(.)$, requiring $k \log t = L$. If $k \log t < L$, the attacker's success probability increases to $\frac{1}{2^{\frac{k \log t}{2}}}$. To maintain security, we ensure $k \log t = L$ by truncating hash outputs to $k \log t$ bits using the $Trunc(.)$ function during the signing.
  Overall, we conclude with an upper bound:

$$\text{InSec}^{EU\text{-}CMA}_{\text{MUM−HORS}}(T) < max(\frac{1}{2^{\frac{L}{2}}}, \frac{1}{2^{\frac{k \log t}{2}}}) + 2^{-k \cdot \frac{L}{2}} + (\frac{k}{t})^k \tag{9}$$

We set the length of the master key ($|msk|$) and private key (HORS l parameter) to κ bits to ensure the minimum required security.

7. Advantages and limitations

Our proposed MUM−HORS builds on the HORS signature scheme to enable fast signing, compact signatures, and post-quantum security, making it well-suited for secure and resource-efficient IoT applications. To enhance security against weak message attacks [49], MUM−HORS employs random pads with lightweight XOR operations, offering a more efficient alternative to rejection sampling and random number generator-based approaches used in HORSIC, HORSIC+, and PORS.

The proposed BM data structure enables MUM−HORS to function as a multi-time signature by efficiently storing private keys as a two-dimensional bit array recoverable from a seed, reducing computational overhead on the signer. It also optimizes public

key storage on the verifier (e.g., achieving a 40-fold reduction for 128-bit security and 2^{20} messages) without relying on hardware assumptions like secure enclaves. Moreover, the BM structure supports implementation via a circular queue or linked list, offering computational trade-offs. For instance, when the number of rows (rt) is large, using a linked list instead of a circular queue can improve row removal performance by eliminating the overhead of row shifting. Nonetheless, we mitigate this cost in the circular queue through optimized shifting from the *head* or *tail* based on the row index (see Appendix A).

While MUM-HORS as a stateful signature offers significant improvements over prior schemes, it inherits the state synchronization challenge common to MTSs based on OTSs [65], where message corruption may desynchronize the BM between signer and verifier. To address this, we propose a probabilistic synchronization algorithm (see Appendix B), though the most reliable approach involves resetting both parties to a fresh row after a corruption threshold is reached. Additionally, MUM-HORS supports only a predetermined number of signings (e.g., 2^{20} or 2^{30}), limiting it to multiple-time rather than full-time usage. Nonetheless, the internal BM offers computational and storage advantages, particularly for applications requiring long-term, infrequent signing.

8. Conclusion and future work

As next-generation networks increasingly depend on resource-constrained IoT devices, secure and scalable authentication becomes critical. Quantum attacks threaten traditional digital signatures, while PQC schemes remain impractical for IoT applications due to their inefficiency. The proposed Maximum Utilization Multiple HORS (MUM-HORS) scheme offers a lightweight, PQ-secure alternative tailored for IoT, enabling fast signing, compact signatures, and efficient key usage. Our scheme demonstrates strong performance and is a viable, secure, and resource-efficient IoT authentication solution. IoT wearables devices (e.g., Google Fitbit) collecting critical data can leverage MUM-HORS to authenticate transmissions with minimal computational and energy overhead, allowing greater emphasis on measurement accuracy. Thus, verifiers (e.g., smartphones) can support multiple devices without storing large numbers of long-term public keys.

Future work can strengthen confidentiality and authenticity by adopting authenticated encryption schemes, such as the lightweight Ascon suite, which was recently standardized by NIST and provides a promising foundation for achieving efficient and secure encryption and signing in resource-constrained environments. Additionally, incorporating secure enclaves (e.g., Intel SGX) on the verifier can enhance key management and eliminate the need for public key preloading.

CRediT authorship contribution statement

Kiarash Sedghighadikolaei: Writing – original draft. **Attila A. Yavuz:** Supervision. **Saif E. Nouma:** Writing – original draft.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Attila A. Yavuz reports financial support was provided by National Science Foundation (NSF CNS-2350213). Attila A. Yavuz has patent A Lightweight Multiple-Time Post-quantum Signature for Heterogeneous Internet of Things pending to University of South Florida. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is supported by the NSF grant CNS-2350213.

Appendix A. Bitmap functionalities using linked list

The proposed BM data structure can be implemented on the signer using linked lists instead of a circular queue, introducing trade-offs between memory usage and computational efficiency. A linked list adds an 8-byte pointer to the next row per node, resulting in a 44-byte overhead for $rt = 11$ rows on a 32-bit system. However, it improves row manipulation, particularly deletion, by avoiding the costly row shifting required in circular queues. Our evaluations show that BITMAP_CLEANUP(.) takes 0.027 μs, BITMAP_UNSET_INDICES(.) 0.18 μs per index, and BITMAP_GET_ROW_COLUMN(.) 0.23 μs on commodity hardware. On Raspberry Pi 4, the performance is: BITMAP_CLEANUP(.) takes 0.51 μs, BITMAP_UNSET_INDICES(.) 0.77 μs per index, and BITMAP_GET_ROW_COLUMN(.) 1.47 μs. The results indicate the additional computational overhead of the array-based design.

Algorithm 3 MUM-HORS Verifier with SCA Algorithm

$b \leftarrow \texttt{MUM-HORS.Ver}(PK, m, \sigma)$: Set $hash \leftarrow Trunc(H(m), k \log t)$ and $b = 1$

1: **if** $hash$ or any $pad_{1,2,3}$ yield distinct i_j as steps 1-2 of $\texttt{MUM-HORS.Sig}(.)$, **then goto** 3
2: Split $hash' \leftarrow Trunc(H(hash||Ctr), k \log t)$ into integer indices i_j. **if** there exists a duplicate index i_j **then** Execute steps 3-4 of $\texttt{MUM-HORS.Sig}(.)$ and **return** $b = 0$.
3: **for** $j = 1$ to k **do**
4: Find the $(i_j)^{th}$ public key as in BITMAP_GET_ROW_COLUMN(.) and count all the pk^*_{ij} until that point as $doubt$
5: **if** s_j is verified **then** mark the corresponding public key as $\perp$
6: **else**
7: Mark the s_j's public key as pk^*_{ij} and try the next $doubt$ non-deleted public keys
8: **if** pk_{ij} was verified **then**
9: // Assume the last verified signature as $s_{j'}$ $(j' < j)$
10: **if** there are $j - j'$ intermediate public keys **then**
11: Mark all the pk^*_{ij} in between as pk_{ij}
12: **else**
13: **if** $j' = j + 1$ **then**
14: Mark all the pk^*_{ij} as the $\perp$
15: **else return** $b = 0$
16: **return** b

Verifier is Idle

17: Invalidate all the pk_j corresponding to the derived i_j as in BITMAP_UNSET_INDICES(.)
18: Extend the view of the public keys similar to BITMAP_EXTEND_MATRIX(.)

Appendix B. Mum-hors signer and verifier index synchronization

Given the challenge of state synchronization in MTSs built on OTSs [65], MUM-HORS is vulnerable to desynchronization if messages are corrupted in transit. To address this, we propose the Second Chance Algorithm (SCA), allowing the verifier to recover from out-of-sync states by retaining mismatched public keys. This self-correcting approach enables the verifier to restore synchronization using future valid message-signature pairs without requiring communication with the signer. The new MUM-HORS verifier is given in Algorithm 3.

The first two steps and steps 17-18 are similar to the MUM-HORS's verifier presented in Algorithm 1. SCA algorithm allows each rejected public key to remain in the list for a second chance, as the rejection cause (message or signature corruption) may be unclear. Normal public keys are denoted as pk_{ij} and those given a second chance as pk^*_{ij}. Using the property of HORS key indices (distance between each key), we apply the following rules: (i) For a verified pair (s_i, s_j) with $i < j$, if there are $j - i$ public keys between them, mark all pk^*_{ij} in between as pk_{ij}. (ii) For a verified pair (s_i, s_{i+1}), mark all pk^*_{ij} in between as $\perp$ (deleted). If a signature cannot be validated, we not only mark the pk_{ij} as pk^*_{ij} but also delete the actual value of the public key for the received private key and mark it as $\perp$ if the signature is not corrupted (Steps 3-16).

It is essential to note that, according to the security proof in Section 6, the advantage in forging a signature does not arise from corrupting the signature, as its security relies on the pre-image resistance of the one-way function used. Therefore, we can assume that any signature corruption results from transmission errors, which can be corrected using error correction codes. Therefore, the SCA algorithm can be optimized under the assumption of message corruption. However, the strict rules of the SCA algorithm may hinder immediate state recovery during many consecutive corruptions, as the number of corruptions could exceed $j - j'$ between two indices i_j and $i_{j'}$.

The SCA algorithm provides a probabilistic solution to the synchronization issue. However, the most reliable approach is to reset both the signer and verifier to a fresh row number – flushing the bitmap and verifier's memory – once the number of corruptions exceeds a specified threshold. This can be managed through an additional communication mechanism and software checks.

Appendix C. Extended related work

Symmetric-key based Approaches: Two primary schemes, namely Hash-based Message Authentication Code (HMAC) [17] and symmetric ciphers [18], are commonly used in IoT systems for their computational efficiency [19]. In addition, Authenticated Encryption (AE) [71], which combines symmetric encryption with MAC, simultaneously ensures confidentiality, integrity, and authenticity, making it well-suited for securing wireless IoT communications. However, symmetric approaches lack scalability in large, distributed systems and do not provide public verifiability essential for applications like health audits [20] and non-repudiation needed in legal contexts [21].

Conventional Digital Signatures: Conventional digital signatures like ECDSA [9], Ed25519 [22], RSA [8], and BLS [72] are considered for IoT [73], wireless spectrum management [74], 6G [75], and various other network domains. However, security protocols relying on these primitives have been shown to deplete battery life in resource-constrained devices and may require frequent undesirable maintenance [10,11]. For example, RSA offers fast verification but requires costly signing and large keys, while ECDSA uses smaller keys but remains computationally intensive. Moreover, relying on traditional intractability assumptions leaves these schemes vulnerable to quantum attacks like Shor's algorithm [14], with ECC-based ones being more susceptible than RSA [76–79].

Lightweight Conventional Signatures with Special Trade-offs: Efforts to develop lightweight digital signatures that improve conventional methods like ECDSA and SchnorrQ often rely on advanced pre-computation and storage techniques, as done by SCRA [13] and Nouma et al. in [80]. These schemes rely on the online/offline signature paradigm [81] involving pre-computed tokens. For instance, Rapid Authentication (RA) [82] leverages pre-computed token aggregation for fast online signature generation. SEMECS [47] optimizes EC-based signature schemes by reducing signature and private key sizes of [12], addressing the computational burden of deriving ephemeral keys in Schnorr [83]-like signatures (e.g., ECDSA, SchnorrQ) through pre-computing them and storing their hash commitments on the verifier. Despite their merits, these schemes are vulnerable to quantum attacks due to relying on traditional intractability assumptions (e.g., (EC)DLP).

Quantum Digital Signatures (QDSs): Quantum Digital Signatures (QDS) have recently gained significant interest due to their potential for information-theoretic security. Yin et al. [15] proposed an efficient QDS scheme combining secret sharing, one-time pads, and one-time universal hashing (OTUH), enabling the signing of multi-bit message hashes. Subsequently, Li et al. [16] showed that quantum keys with imperfect secrecy can still ensure authenticity and integrity and introduced an OTUH-QDS protocol using asymmetric quantum keys without perfect secrecy. Their scheme maintains information-theoretic security and achieves up to a 10^6-fold increase in signature rate over long distances. Zhang et al. [84] proposed a Continuous-Variable (CV) QDS scheme based on OTUH that uses coherent optical setups and homodyne/heterodyne detection to encode message hashes into continuous degrees of freedom. Their approach improves efficiency by signing multi-bit hashes with a single key string and ensures security against general coherent attacks via discrete-modulated CV key generation. Evaluations show high signing rates—1000 signatures per second over 40 km for 1 KB messages and 35 km for 1 MB messages.

Further developments include Coladangelo et al. [85], who proved the impossibility of constructing QDS schemes with classical signatures from pseudorandom states using black-box techniques, highlighting fundamental limits. This complements the earlier work by Yamakawa et al. [86], who proposed a one-time secure QDS scheme but left multi-time security unresolved. Qin et al. [87] also introduced the likely bit string method, which can boost signature rates and extend secure transmission distances when integrated into quantum key distribution-based QDS protocols. While these results show significant progress in advancing QDS, their reliance on quantum-related infrastructure limits their applicability in resource-constrained IoT environments.

PQ-secure Signatures: The NIST-PQC standardization [24,88] features $\mathcal{LB}$ signatures ML-DSA [25] and Falcon [26], alongside $\mathcal{HB}$ SLH-DSA [27] for PQ-secure signatures. ML-DSA, built on the Fiat-Shamir with Aborts paradigm [28] and M-LWE/M-SIS assumptions [89,90], reduces public key size via uniform sampling but produces slightly larger signatures. Falcon [91], based on NTRU lattices [29], relies on double-precision arithmetic and operations such as Fast Fourier Transform and matrix decomposition, making it impractical for platforms without hardware floating-point support.

Post-quantum cryptography for IoT systems has been extensively studied in recent years [30]. Kumari et al. [92] introduced a novel Diffie Supersingular Multiplication to enhance the security of high-performance IoT devices, such as smartphones, smartwatches, and smart security systems, against sophisticated threats like ransomware and botnets. The proposed scheme achieves this with 15% lower power consumption and 20% reduced delay. Subsequently, the authors tackled the issue of polynomial multiplication in $\mathcal{LB}$ cryptography by proposing a delay-efficient technique, resulting in a signing time of 13.299 ms [31]. Bagchi et al. [32] proposed a security framework for ambient intelligence-enabled, blockchain-based IoT systems. Using $\mathcal{LB}$ signatures, their approach enables secure data transmission from wearable or medical devices. These signatures are sent to controller nodes, which forward them to aggregator nodes responsible for generating an aggregate signature after verification. For Space Information Networks, Ma et al. [33] developed a novel $\mathcal{LB}$ access authentication protocol, addressing the challenges of massive IoT deployment and constrained satellite resources. Their protocol integrates a semi-aggregated signature mechanism with session key agreement to ensure mutual authentication, anonymity, data confidentiality, and resistance to quantum and protocol-level attacks. Despite these advances, $\mathcal{LB}$ schemes remain computationally intensive, with large key and signature sizes, rendering them impractical for resource-constrained devices.

Lightweight PQ-secure Signatures with Special Assumptions: Recent lightweight PQ-secure signature schemes rely on specific architectural features and distributed computation to enhance the signer's efficiency. ANT [34] transforms a $\mathcal{LB}$ one-time signature (OTS) into a polynomially-bounded many-time signature through distributed public key computation under the assumption of honest-but-curious servers. HASES [43] and its extension [44] convert $\mathcal{HB}$ HORS signature [36] into a many-time signature, assuming a secure enclave (e.g., Intel SGX [45]) to store the private key, enabling public key derivation and forward security. Despite their signer efficiency, the special assumptions on the verifier can limit the applicability of these constructions in some NextG IoT settings.

Hash-based Signatures and their Building Blocks: Unlike $\mathcal{LB}$ schemes, $\mathcal{HB}$ standards such as XMSS [93] and SLH-DSA [27] rely on minimal intractability and number-theoretical assumptions, offering strong PQ security. These schemes use widely available cryptographic hash functions like SHA-256, facilitating the transition to PQ-secure options. While some $\mathcal{HB}$ signatures provide efficient signing and verification for a limited number of signatures [36–38], others, such as [94–96], offer high security for extended usage but involve large signature sizes and costly signing processes.

Abdullah et al. [97] demonstrated the applicability of OTSs for IoT environments by exploring using Lamport signatures for digital authentication. Hash-based FTSs, which are constructed from OTSs [38,57,98,99], enable a limited number of signatures per key pair, offering improved computational and communication efficiency compared to pure OTSs. However, this comes at the cost of reduced security with each additional signature generated. The first FTS, BiBa [100], prioritized fast verification and small signature sizes but had trade-offs in signing time and public key size. Subsequent FTSs, like HORS [36] and its variants [7,37,50,55,94,101], built on Lamport OTS with Bos-Chaum signatures and cover-free families, along with PORS [51] and FORS [102] enhanced

the robustness against weak message attacks [49]. However, they have large private keys (e.g., HORSE [55]) and signing times (e.g., HORSIC+ [50]), or are time-valid secure (e.g., TV-HORS [101]).

The integration of $\mathcal{HB}$ signatures into IoT applications has been widely studied. Tandel et al. [39] presented a hash-based authentication framework for microcontroller-based client–server IoT applications, utilizing XMSS with WOTS +, SHA-256 and a pseudorandom function. Their evaluation shows signing times of 1–2 s on ESP32 devices, depending on verifier hardware instantiation. Bos et al. [40] optimized XMSS for embedded devices by refining a probabilistic signing method that decouples signing cost from message length, achieving two-fold verification speedup while maintaining a one-second signing time on general-purpose systems. Ghosh et al. [52] proposed a latency-area optimized hardware-software hybrid architecture for XMSS, targeting resource-constrained IoT devices. Designed for 128-bit post-quantum security, the implementation achieves a signing time of 100 ms on IoT edge platforms. Bellini et al. [103] introduced a one-pass XMSS-based protocol that restricts the number of authentications per key pair and achieves a 5.20 ms signing time, minimizing computational and communication overhead. Saldamli et al. [104] evaluated the Merkle Tree signature scheme on a resource-constrained IoT device (Pyboard with STM32F405RG), demonstrating its quantum-resistant properties with a signing time of 500 ms.

Beyond algorithmic optimizations, some approaches leverage hardware acceleration. Wang et al. [42] proposed a software-hardware co-design of XMSS on a RISC-V embedded processor, optimizing the reference SHA-256-based implementation and integrating hardware accelerators achieving 42× improvement for signing while maintaining area efficiency. Despite their advantages, these $\mathcal{HB}$ constructions impose considerable signer overhead and depend on specialized hardware, limiting their practicality for resource-limited IoT applications.

Among $\mathcal{HB}$ signature schemes, HORS [36] is valued for its efficiency and underpins signatures like XMSS^{MT} [35] and SLH-DSA [27]. However, extending HORS FTS to full-time signatures using hypertree-based (e.g., SLH-DSA) or tree-based (e.g., XMSS^{MT}) methods is inefficient for resource-constrained devices. An alternative is the online/offline paradigm (as in [47]), where public keys are pre-computed and stored offline, with one key used per signing round. While HORS benefits from small k values for short signatures, it requires large t for adequate security (e.g., 128-bit). For example, signing 2^{20} messages with 128-bit security requires $t = 1024$ and $k = 25$, resulting in 32 GB of public key storage. In each signing round, 25 keys are used and 999 discarded, resulting in a 97% key loss and only 2.44% effective utilization of the public key storage. This inefficiency affects device utility, authentication lifetime, and computation/storage requirements. To reduce key discards in HORS, private keys can be tracked via indices (one per key), where each key is derived from the master key by padding it with the corresponding index. However, this memory-inefficient approach requires 4 KB of storage for $t = 1024$ (assuming each index occupies 4 bytes). For microcontrollers like the ATmega328, which have limited flash memory and a threshold of 10,000 write/erase cycles, maintaining the index list in SRAM reduces flash write operations and preserves the data without corruption. Thus, this necessitates a compact data structure, making this method challenging. While Jakobsson's hash-based method [55,67] used in HORSE's [55] key generation reduces storage, it incurs high signing overhead, making it impractical.

Data availability

The implementation of `MUM-HORS` has been open-sourced and is available on GitHub at https://github.com/kiarashsedghigh/mumhors.

References

[1] Abdul Ahad, Mohammad Tahir, Muhammad Aman Sheikh, Kazi Istiaque Ahmed, Amna Mughees, Abdullah Numani, Technologies trend towards 5 g network for smart health-care using iot: A review, Sensors 20 (14) (2020) 4047.

[2] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, Dusit Niyato, Octavia Dobre, H. Vincent Poor, 6 g internet of things: A comprehensive survey, IEEE Internet Things J. 9 (1) (2021) 359–383.

[3] Manas Pradhan, Josef Noll, Security, privacy, and dependability evaluation in verification and validation life cycles for military iot systems, in: IEEE Communications Magazine, vol. 58, (8) 2020, pp. 14–20.

[4] Abolfazl Mehbodniya, Julian L. Webber, Rahul Neware, Farrukh Arslan, Raja Varma Pamba, Mohammad Shabaz, Modified lamport merkle digital signature blockchain framework for authentication of internet of things healthcare data, Expert Syst. 39 (10) (2022) 12978.

[5] Maria-Dolores Cano, Antonio Cañavate-Sanchez, Preserving data privacy in the internet of medical things using dual signature ecdsa, Secur. Commun. Netw. 2020 (1) (2020) 4960964.

[6] Muslum Ozgur Ozmen, Attila A. Yavuz, Rouzbeh Behnia, Energy-aware digital signatures for embedded medical devices, in: 2019 IEEE Conference on Communications and Network Security, CNS, IEEE, 2019, pp. 55–63.

[7] Attila A. Yavuz, Saleh Darzi, Saif E. Nouma, Lightweight and scalable post-quantum authentication for medical Internet of Things.

[8] R.L. Rivest, A. Shamir, L.A. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.

[9] Don Johnson, Alfred Menezes, Scott Vanstone, The elliptic curve digital signature algorithm (ecdsa), Int. J. Inf. Secur. 1 (1) (2001) 36–63.

[10] Aleksandr Ometov, Pavel Masek, Lukas Malina, Roman Florea, Jiri Hosek, Sergey Andreev, Jan Hajny, Jussi Niutanen, Yevgeni Koucheryavy, Feasibility characterization of cryptographic primitives for constrained (wearable) iot devices, in: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops, IEEE, 2016.

[11] Giuseppe Ateniese, Giuseppe Bianchi, Angelo T. Capossele, Chiara Petrioli, Dora Spenza, Low-cost standard signatures for energy-harvesting wireless sensor networks, ACM Trans. Embed. Comput. Syst. (TECS) 16 (3) (2017) 1–23.

[12] Attila A. Yavuz, ETA: Efficient and tiny and authentication for heterogeneous wireless systems, in: Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM.

[13] A.A. Yavuz, A. Mudgerikar, A. Singla, I. Papapanagiotou, E. Bertino, Real-time digital signatures for time-critical networks, IEEE Trans. Inf. Forensics Secur. (2017).

[14] Peter W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. (1999).

[15] Hua-Lei Yin, Yao Fu, Chen-Long Li, Chen-Xun Weng, Bing-Hong Li, Jie Gu, Yu-Shuo Lu, Shan Huang, Zeng-Bing Chen, Experimental quantum secure network with digital signatures and encryption, Natl. Sci. Rev. 10 (4) (2023) nwac228.

[16] Bing-Hong Li, Yuan-Mei Xie, Xiao-Yu Cao, Chen-Long Li, Yao Fu, Hua-Lei Yin, Zeng-Bing Chen, One-time universal hashing quantum digital signatures without perfect keys, Phys. Rev. Appl. 20 (4) (2023) 044011.

[17] M. Bellare, P. Rogaway, Introduction to modern cryptography, in: UCSD CSE Course, first ed., 2005, p. 207, http://www.cs.ucsd.edu/mihir/cse207/classnotes.html.

[18] D. Stinson, Cryptography: Theory and Practice, second ed., CRC/C & H, 2002.

[19] Jorge Guajardo, Attila A. Yavuz, Benjamin Glas, Markus Ihle, Hamit Hacioglu, Karsten Wehefrit, System and method for counter mode encrypted communication with reduced bandwidth, 2014, US 20140270163 A1, Filed: March 14 2013, Issued: September 18 2014.

[20] Daniel Halperin, Thomas S. Heydt-Benjamin, Kevin Fu, Tadayoshi Kohno, William H. Maisel, Security and privacy for implantable medical devices, in: IEEE Pervasive Computing, vol. 7, no. 1, 2008, pp. 30–39.

[21] Carmen Camara, Pedro Peris-Lopez, Juan E. Tapiador, Security and privacy issues in implantable medical devices: A comprehensive survey, J. Biomed. Inform. 55 (2015) 272–289.

[22] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, Bo-Yin Yang, High-speed high-security signatures, J. Cryptogr. Eng. 2 (2) (2012) 77–89.

[23] Gorjan Alagic, Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, et al., Status report on the third round of the nist post-quantum cryptography standardization process, 2022.

[24] Saleh Darzi, Kasra Ahmadi, Saeed Aghapour, Attila Altay Yavuz, Mehran Mozaffari Kermani, Envisioning the future of cyber security in post-quantum era: A survey on pq standardization, applications, challenges and opportunities, 2023, arXiv preprint arXiv:2310.12037.

[25] National Institute of Standards, Technology (NIST), Thinh Dang, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Module-lattice-based digital signature standard, 2024, 2024-08-13 04:08:00.

[26] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al., Falcon: Fast-fourier lattice-based compact signatures over ntru, Submiss. NIST's Post-Quantum Cryptogr. Stand. Process. 36 (5) (2018) 1–75.

[27] National Institute of Standards, Technology (NIST), David Cooper, Stateless hash-based digital signature standard, 2024, 2024-08-13 04:08:00.

[28] Vadim Lyubashevsky, Fiat-shamir with aborts: Applications to lattice and factoring-based signatures, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2009, pp. 598–616.

[29] Léo Ducas, Vadim Lyubashevsky, Thomas Prest, Efficient identity-based encryption over ntru lattices, in: Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December (2014) 7-11, Proceedings, Part II 20, Springer, 2014, pp. 22–41.

[30] Swati Kumari, Maninder Singh, Raman Singh, Hitesh Tewari, Post-quantum cryptography techniques for secure communication in resource-constrained internet of things devices: A comprehensive survey, Softw.: Pr. Exp. 52 (10) (2022).

[31] Swati Kumari, Maninder Singh, Raman Singh, Hitesh Tewari, A post-quantum lattice based lightweight authentication and code-based hybrid encryption scheme for iot devices, Comput. Netw. 217 (2022) 109327.

[32] Prithwi Bagchi, Basudeb Bera, Ashok Kumar Das, Sachin Shetty, Pandi Vijayakumar, Marimuthu Karuppiah, Post quantum lattice-based secure framework using aggregate signature for ambient intelligence assisted blockchain-based iot applications, IEEE Internet Things Mag. 6 (1) (2023) 52–58.

[33] Ruhui Ma, Jin Cao, Dengguo Feng, Hui Li, Laa: lattice-based access authentication scheme for iot in space information networks, IEEE Internet Things J. 7 (4) (2019) 2791–2805.

[34] Rouzbeh Behnia, Attilla Altay Yavuz, Towards practical post-quantum signatures for resource-limited internet of things, in: Annual Computer Security Applications Conference, ACSAC, Association for Computing Machinery, New York, NY, USA, 2021, pp. 119–130.

[35] Johannes Buchmann Erik Dahmen, Andreas Hülsing, Xmss - a practical forward secure signature scheme based on minimal security assumptions, in: Bo-Yin Yang (Ed.), Post-Quantum Cryptography, Springer, Berlin Heidelberg, 2011, pp. 117–129.

[36] Leonid Reyzin, Natan Reyzin, Better than Biba: Short One-Time Signatures with Fast Signing and Verifying, Springer, 2002.

[37] J. Lee, S. Kim, Y. Cho, Y. Chung, Y. Park, HORSIC: An efficient one-time signature scheme for wireless sensor networks, Inform. Process. Lett. 112 (20) (2012) 783–787.

[38] Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing, Markus Rückert, On the security of the winternitz one-time signature scheme, in: International Conference on Cryptology in Africa, Springer, 2011, pp. 363–378.

[39] Purvi Tandel, Jitendra Nasriwala, Secure authentication framework for iot applications using a hash-based post-quantum signature scheme, Serv. Oriented Comput. Appl. (2024) 1–12.

[40] Joppe W. Bos, Andreas Hülsing, Joost Renes, Christine van Vredendaal, Rapidly verifiable xmss signatures, IACR Trans. Cryptogr. Hardw. Embed. Syst. (2021) 137–168.

[41] Andreas Hülsing, Christoph Busold, Johannes Buchmann, Forward secure signatures on smart cards: Preliminary version, in: International Conference on Selected Areas in Cryptography, Springer, 2012, pp. 66–80.

[42] Wen Wang, Bernhard Jungk, Julian Wälde, Shuwen Deng, Naina Gupta, Jakub Szefer, Ruben Niederhagen, Xmss and embedded systems: Xmss hardware accelerators for risc-v, in: International Conference on Selected Areas in Cryptography, Springer, 2019, pp. 523–550.

[43] Saif E. Nouma, Attila A. Yavuz, Post-quantum forward-secure signatures with hardware-support for internet of things, in: ICC 2023-IEEE International Conference on Communications, IEEE, 2023, pp. 4540–4545.

[44] Saif E. Nouma, Attila A. Yavuz, Trustworthy and efficient digital twins in post-quantum era with hybrid hardware-assisted signatures, ACM Trans. Multimed. Comput. Commun. Appl. (2023).

[45] Frank McKeen, Ilya Alexandrovich, Ittai Anati, Dror Caspi, Simon Johnson, Rebekah Leslie-Hurd, Carlos Rozas, Intel®software guard extensions (intel®sgx) support for dynamic memory management inside an enclave, in: Proceedings of the Hardware and Architectural Support for Security and Privacy 2016, HASP 2016, New York, NY, USA, Association for Computing Machinery, 2016.

[46] Jo Van Bulck, David Oswald, Eduard Marin, Abdulla Aldoseri, Flavio D. Garcia, Frank Piessens, A tale of two worlds: Assessing the vulnerability of enclave shielding runtimes, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1741–1758.

[47] A.A. Yavuz, M.O. Ozmen, Ultra lightweight multiple-time digital signature for the Internet of Things devices, IEEE Trans. Serv. Comput. (2019) 1–1.

[48] Attila A. Yavuz, Peng Ning, Michael K. Reiter, BAF and FI-BAF: Efficient and publicly verifiable cryptographic schemes for secure logging in resource-constrained systems, ACM Trans. Inf. Syst. Secur. 15 (2) (2012).

[49] Jean-Philippe Aumasson, Guillaume Endignoux, Clarifying the subset-resilience problem, 2017, Cryptology ePrint Archive.

[50] Jaeheung Lee, Yongsu Park, Horsic+: An efficient post-quantum few-time signature scheme, Appl. Sci. 11 (16) (2021) 7350.

[51] Jean-Philippe Aumasson, Guillaume Endignoux, Improving stateless hash-based signatures, in: Cryptographers' Track at the RSA Conference, Springer, 2018, pp. 219–242.

[52] Santosh Ghosh, Rafael Misoczki, Manoj R. Sastry, Lightweight post-quantum-secure digital signature approach for iot motes, 2019, Cryptology ePrint Archive.

[53] Simon Josefsson, Ilari Liusvaara, Edwards-Curve Digital Signature Algorithm (Eddsa), Technical Report, 2017.

[54] Craig Costello, Patrick Longa, Schnorrq: Schnorr signatures on fourq. Technical report, MSR tech report, 2016, Available at: https://wwwmicrosoft.com/en-us/research/wp-content/uploads/2016/07/SchnorrQ.pdf.

[55] William D. Neumann, Horse: an extension of an r-time signature scheme with fast signing and verification, in: International Conference on Information Technology: Coding and Computing, 2004.

[56] Sebastian Rohde, Thomas Eisenbarth, Erik Dahmen, Johannes Buchmann, Christof Paar, Fast hash-based signatures on constrained devices, in: Smart Card Research and Advanced Applications: 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008, Springer, 2008.

[57] Andreas Hülsing, Lea Rausch, Johannes Buchmann, Optimal parameters for xmss mt, in: Security Engineering and Intelligence Informatics: CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Regensburg, Germany, September (2013) 2-6. Proceedings 8, Springer, 2013, pp. 194–208.

[58] FiBit, Fitbit official site for activity trackers & more — fitbit.com, 2024, https://www.fitbit.com/global/us/home, (Accessed 21 July 2024).

[59] Apple, Apple watch — apple.com, 2024, https://www.apple.com/watch/. (Accessed 21 July 2024).

[60] Moayad Aloqaily, Ouns Bouachir, Fakhri Karray, Ismaeel Al Ridhawi, Abdulmotaleb El Saddik, Integrating digital twin and advanced intelligent technologies to realize the metaverse, IEEE Consum. Electron. Mag. 12 (6) (2022) 47–55.

[61] Xu Yang, Xun Yi, Surya Nepal, Ibrahim Khalil, Xinyi Huang, Jian Shen, Efficient and anonymous authentication for healthcare service with cloud based wbans, IEEE Trans. Serv. Comput. 15 (5) (2021) 2728–2741.

[62] Mikail Mohammed Salim, Laurence Tianruo Yang, Jong Hyuk Park, Lightweight authentication scheme for iot based e-healthcare service communication, IEEE J. Biomed. Heal. Inform. (2023).

[63] Hakjun Lee, Slars: Secure lightweight authentication for roaming service in smart city, IEICE Trans. Commun. (2024).

[64] S.C. Rajkumar, K.R. Karthick, et al., Secure session key pairing and a lightweight key authentication scheme for liable drone services, Cyber Secur. Appl. 1 (2023) 100012.

[65] David McGrew, Panos Kampanakis, Scott Fluhrer, Stefan-Lukas Gazdag, Denis Butin, Johannes Buchmann, State management for hash-based signatures, in: Security Standardisation Research: Third International Conference, SSR 2016, Gaithersburg, MD, USA, December (2016) 5–6, Proceedings 3, Springer, 2016, pp. 244–260.

[66] Martin Raab, Angelika Steger, Balls into bins—a simple and tight analysis, in: International Workshop on Randomization and Approximation Techniques in Computer Science, Springer, 1998, pp. 159–170.

[67] Markus Jakobsson, Fractal hash sequence representation and traversal, in: Proceedings IEEE International Symposium on Information Theory, IEEE, 2002, p. 437.

[68] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, Zooko Wilcox-O'Hearn, Sphincs: practical stateless hash-based signatures, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2015, pp. 368–397.

[69] Krzysztof Piotrowski, Peter Langendoerfer, Steffen Peter, How public key cryptography influences wireless sensor node lifetime, in: Proceedings of the Fourth ACM on Security of Ad Hoc and Sensor Networks, 2006.

[70] Lov K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96, ACM, New York, NY, USA, 1996, pp. 212–219.

[71] Luan Cardoso dos Santos, Johann Großschädl, Alex Biryukov, Felics-aead: benchmarking of lightweight authenticated encryption algorithms, in: International Conference on Smart Card Research and Advanced Applications, Springer, 2019, pp. 216–233.

[72] Dan Boneh, Ben Lynn, Hovav Shacham, Short signatures from the weil pairing, in: Colin Boyd (Ed.), Advances in Cryptology — ASIACRYPT 2001, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 514–532.

[73] Renuka Sahebrao Pawar, Dhananjay Ramrao Kalbande, Optimization of quality of service using eceba protocol in wireless body area network, Int. J. Inf. Technol. 15 (2) (2023) 595–610.

[74] M. Grissa, A.A. Yavuz, B. Hamdaoui, Cuckoo filter-based location-privacy preservation in database-driven cognitive radio networks, in: Computer Networks and Information Security (WSCNIS), 2015 World Symposium on, 2015, pp. 1–7.

[75] Syed Hussain Ali Kazmi, Rosilah Hassan, Faizan Qamar, Kashif Nisar, Ag Asri Ag Ibrahim, Security concepts in emerging 6 g communication: Threats, countermeasures, authentication techniques and research directions, Symmetry 15 (6) (2023) 1147.

[76] Daniel J. Bernstein, Nadia Heninger, Paul Lou, Luke Valenta, Post-quantum rsa, in: Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, the Netherlands, June (2017) 26-28, Proceedings 8, Springer, 2017, pp. 311–329.

[77] Thomas Häner, Samuel Jaques, Michael Naehrig, Martin Roetteler, Mathias Soeken, Improved quantum circuits for elliptic curve discrete logarithms, in: Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April (2020) 15–17, Proceedings 11, Springer, 2020, pp. 425–444.

[78] Craig Gidney, Martin Ekerå, How to factor 2048 bit rsa integers in 8 h using 20 million noisy qubits, 2019, arXiv preprint arXiv:, 1905.

[79] John Proos, Christof Zalka, Shor's discrete logarithm quantum algorithm for elliptic curves, 2003, arXiv preprint quant-ph/0301141.

[80] Saif E. Nouma, Attila A. Yavuz, Practical Cryptographic Forensic Tools for Lightweight Internet of Things and Cold Storage Systems, IoTDI '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 340–353.

[81] Adi Shamir, Yael Tauman, Improved online/offline signature schemes, in: Advances in Cryptology—CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August (2001) 19–23 Proceedings 21, Springer, 2001, pp. 355–367.

[82] Attila A. Yavuz, An efficient real-time broadcast authentication scheme for command and control messages, Inf. Forensics Secur. IEEE Trans. on 9 (10) (2014) 1733–1742.

[83] C. Schnorr, Efficient signature generation by smart cards, J. Cryptol. 4 (3) (1991) 161–174.

[84] Yi-Fan Zhang, Wen-Bo Liu, Bing-Hong Li, Hua-Lei Yin, Zeng-Bing Chen, Continuous-variable quantum digital signatures that can withstand coherent attacks, Phys. Rev. A 110 (5) (2024) 052613.

[85] Andrea Coladangelo, Saachi Mutreja, On black-box separations of quantum digital signatures from pseudorandom states, in: Theory of Cryptography Conference, Springer, 2024, pp. 289–317.

[86] Tomoyuki Morimae, Takashi Yamakawa, One-wayness in quantum cryptography, 2022, arXiv preprint arXiv:2210.03394.

[87] Ji-Qian Qin, Zong-Wen Yu, Xiang-Bin Wang, Efficient quantum digital signatures over long distances with likely bit strings, Phys. Rev. Appl. 21 (2) (2024) 024012.

[88] Post-quantum cryptography. Selected algorithms, 2022, 2022, URL: https://csrc.nist.gov/projects/post-quantum-cryptography/selected-algorithms-2022.

[89] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan, (Leveled) fully homomorphic encryption without bootstrapping, ACM Trans. Comput. Theory (TOCT) 6 (3) (2014) 1–36.

[90] Adeline Langlois, Damien Stehlé, Worst-case to average-case reductions for module lattices, Des. Codes Cryptogr. 75 (3) (2015) 565–599.

[91] Léo Ducas, Thomas Prest, Fast fourier orthogonalization, in: Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, 2016, pp. 191–198.

[92] Swati Kumari, Maninder Singh, Raman Singh, Hitesh Tewari, To secure the communication in powerful internet of things using innovative post-quantum cryptographic method, Arab. J. Sci. Eng. 47 (2) (2022).

[93] Andreas Hülsing, Denis Butin, Stefan Gazdag, Joost Rijneveld, Aziz Mohaisen, Xmss: extended merkle signature scheme, 2018.

[94] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, Zooko Wilcox-O'Hearn, SPHINCS: Practical stateless hash-based signatures, in: Advances in Cryptology – EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin Heidelberg, 2015, pp. 368–397.

[95] Andreas Hülsing, Joost Rijneveld, Peter Schwabe, Armed sphincs, in: Proceedings, Part I, of the 19th IACR International Conference on Public-Key Cryptography — PKC 2016 - Volume 9614, Springer-Verlag, Berlin, Heidelberg, 2016, pp. 446–470.
[96] Johannes Buchmann, Erik Dahmen, Andreas Hülsing, Xmss - a practical forward secure signature scheme based on minimal security assumptions, in: Proceedings of the 4th International Conference on Post-Quantum Cryptography, PQCrypto'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 117–129.
[97] Ghazi Muhammad Abdullah, Quzal Mehmood, Chaudry Bilal Ahmad Khan, Adoption of lamport signature scheme to implement digital signatures in iot, in: 2018 International Conference on Computing, Mathematics and Engineering Technologies, ICoMET, IEEE, 2018, pp. 1–4.
[98] Leslie Lamport, Constructing Digital Signatures from a One-Way Function, Technical Report, Technical Report CSL-98, SRI International Palo Alto, 1979.
[99] Ralph Charles Merkle, Secrecy, Authentication, and Public Key Systems, Stanford university, 1979.
[100] A. Perrig, The BiBa: One-time signature and broadcast authentication protocol, in: Proceedings of the ACM Conference on Computer and Communications Security, 2001, pp. 28–37.
[101] Qiyan Wang, Himanshu Khurana, Ying Huang, Klara Nahrstedt, Time valid one-time signature for time-critical multicast data authentication, in: IEEE INFOCOM 2009, IEEE, 2009, pp. 1233–1241.
[102] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, Peter Schwabe, The Sphincs+ Signature Framework, Association for Computing Machinery, 2019.
[103] Emanuele Bellini, Florian Caullery, Alexandros Hasikos, Marc Manzano, Victor Mateu, You shall not pass!(once again) an iot application of post-quantum stateful signature schemes, in: Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop, 2018, pp. 19–24.
[104] Gokay Saldamli, Levent Ertaul, Bharani Kodirangaiah, Post-quantum cryptography on iot: Merkle's tree authentication, in: Proceedings of the International Conference on Wireless Networks, ICWN, The Steering Committee of The World Congress in Computer Science, Computer . . . , 2018, pp. 35–41.