Unsourced Massive Random Access Scheme Exploiting Reed-Muller Sequences

Jue Wang, *Student Member, IEEE,* Zhaoyang Zhang, *Senior Member, IEEE,*
Xiaoming Chen, *Senior Member, IEEE,* Caijun Zhong, *Senior Member, IEEE,* and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—The challenge in massive Machine Type Communication (mMTC) is to support reliable and instant access for an enormous number of machine-type devices (MTDs). In some particular applications of mMTC, the access point (AP) only has to know the messages received, but not where they source from, thus giving rise to the concept of unsourced random access (URA). In this paper, we propose a novel URA scheme exploiting the elegant properties of Reed-Muller (RM) sequences. Specifically, after dividing the message of an active user into several information chunks, RM sequences are used to carry those chunks, for exploiting the vast sequence space to improve the spectral efficiency, and their nested structure to enable reliable and efficient sequence detection. Next, we further explore a novel structural property of RM sequences for designing sparse patterns which carry part of the information and serve as the hints of coupling the information chunks of a single user. The factors affecting the performance of our slot-based RM detection are characterized. Besides, the complexity of the proposed message stitching method is analyzed and compared to the commonly used tree coding approach. Our simulation results verify the enhanced performance of the proposed URA scheme in error probability and computational complexity compared to the existing counterpart.**

***Index Terms*—massive Machine Type Communication, unsourced random access, Reed-Muller sequences**

I. Introduction

Massive Machine Type Communication (mMTC) aims for providing instant and reliable access for a large number of machine-type devices (MTDs) [1]. Compared to human-centric communication, the mMTC scenario has a pair of striking features, which constitute immense challenges in designing random access (RA) procedures. On the one hand, despite having an enormous number of MTDs in mMTC, the proportion of active ones in a specific communication round is usually small because of the sporadic activity of MTDs [2]. Hence, it poses the challenge of providing massive connectivity while efficiently detecting the messages sent by a tiny but unknown subset of users. Furthermore, the teletraffic generated by MTDs is typically in the form of short packets [3], which makes the traditional grant-based scheme unacceptable because of its low spectral efficiency (SE) and high latency. Thus, designing reliable and efficient grant-free RA schemes is regarded as a breakthrough in solving this problem.

In most grant-free RA schemes, known preambles are sent to the access point (AP) before the payload messages in order to identify the active users and to estimate their channel conditions [4] [5]. However, in some application scenarios like the cyber-physical systems (CPSs) [6], the AP is only interested in the transmitted messages, regardless of their sources, thus leading to the concept of unsourced random access (URA) that proposed by Polyanskiy in [7]. In URA, all users share a common codebook, and the active ones select codewords from it to carry their messages. The AP aims for capturing the content of the messages without having to know the active users' identities (IDs). Polyanskiy also derived an achievability bound for the real-addition Gaussian multiple access channel (GMAC) using random Gaussian codebooks and maximum-likelihood (ML) decoding [7]. Ngo *et al.* further extend the achievability bound to the case where the number of active users are unknown [8]. But the setting of random coding with ML decoding is computationally intractable in practice. The sporadic nature of the users' activity in mMTC, on the other hand, becomes the inspiration for formulating the message recovery at the AP as a compressive sensing (CS) problem. However, the complexity of CS-based algorithms escalates with the size of the common codebook, which grows exponentially with the number of information bits. Thus, the CS-based algorithms may become computationally intractable in realistic URA scenarios where the size of messages is typically on the order of 100 bits. In this context, Amalladinne *et al.* [9] propose a packetized and slotted transmission framework, and further design a coded/coupled CS (CCS) URA scheme. Specifically, the message of each active user is partitioned into several information chunks and then tree encoding is executed to couple them in some way. Next, each chunk is subjected to the standard CS-based encoding/decoding procedures within a slot. After recovering the slot-based transmitted codewords, tree decoding is performed for stitching together the chunks belonging to the same message. As a further development, Fengler *et al.* [10] propose a URA scheme that combines inner sparse regression coding (SPARC) with outer tree coding,

This work was supported in part by National Natural Science Foundation of China under Grant 61725104 and U20A20158, National Key R&D Program of China under Grant 2018YFB1801104 and 2020YFB1807101.

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/P034284/1 and EP/P003990/1 (COALESCE) as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028).

J. Wang (e-mail: juew@zju.edu.cn), Z. Zhang (Corresponding Author, e-mail: ning_ming@zju.edu.cn), X. Chen (e-mail: chen_xiaoming@zju.edu.cn) and C. Zhong (e-mail: caijunzhong@zju.edu.cn) are with College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, and with International Joint Innovation Center, Zhejiang University, Haining 314400, China, and also with Zhejiang Provincial Key Laboratory of Info. Proc., Commun. & Netw. (IPCAN), Hangzhou 310027, China.

L. Hanzo (e-mail: lh@ecs.soton.ac.uk) is with the Department of Electronics and Computer Science, University of Southampton, UK.

which is then enhanced exploiting belief propagation in [11]. However, numerous parity bits are required for implementing tree-based message stitching, thus reducing both the code rate and SE. Furthermore, the complexity of tree decoding exhibits an exponential increase with the number of slots. As a design alternative, Shyianov *et al.* [12] propose an uncoupled CS-based URA scheme. It exploits the correlation between the slot-by-slot estimated channel coefficients for clustering the information chunks across different slots, thus avoiding the overhead of parity checks. However, channels have to remain constant over the entire transmission period, which is hard to hold in reality. Besides, a tensor-based URA scheme is proposed in [13], where messages are modulated into rank-1 tensors and then tensor decomposition is utilized to separate messages at the AP.

The family of second-order Reed-Muller (RM) sequences has wide-ranging applications in CS [14]–[17]. For example, Howard *et al.* [14] employ RM sequences for constructing deterministic measurement matrices and conceive an efficient chirp reconstruction algorithm, whose complexity depends on the number of measurements rather than on the signal dimension. In [17], on-off RM signatures are used for supporting full-duplex compressed neighbor discovery. Furthermore, as a benefit of their vast sequence space, RM sequences are also regarded as promising candidates for massive access in mMTC [18] because they readily lend themselves to low-complexity RA. By exploiting these compelling properties, a joint active user detection and channel estimation algorithm is proposed in [19] based on the nested structure of RM sequences, while an incremental massive RA scheme is put forward in [20]. Additionally, a URA scheme exploiting RM sequences is proposed in [21]. It is under the framework of packetized and slotted transmission and utilizes tree coding for realizing message stitching.

In this paper, we propose a novel RM-based URA scheme based on the typical packetized and slotted transmission framework. Its main difference from the one in [21] is that both the slot-based information transmission and the message stitching procedures exploit the beneficial properties of RM sequences. Specifically, every active user first splits its message into two parts. One of them is further partitioned into small chunks, and each chunk is mapped to a single RM sequence (i.e., a codeword) of our specifically constructed common codebook. By contrast, the other one is used to select the Sparse Shift Pattern (SSP), which is designed based on the so-called shift property of RM sequences we explored. Then, the active users transmit their RM sequences following their own SSPs. At the AP, the RM detection algorithm proposed in [19] (referred to as Alg. $\mathbf{\Phi}$ in the rest of the paper with the sketch given in Appendix C), which exploits the nested structure of RM sequences, is employed for detecting the RM sequences received slot by slot. Finally, the AP utilizes the specific clues provided by the SSPs for accomplishing the message stitching. To summarize, our contributions are listed as follows:

- The shift property of RM sequences is explored, which reveals the change rule of the element positions of a single RM sequence when the matrix-vector pair that generates it is shifted in a given way.
- Based on the shift property, we design a pool of SSPs according to our specific rules. These SSPs convey information bits in their indices and represent the transmission patterns of the RM sequences to serve as guidance during the message stitching.
- A common RM codebook is designed for facilitating SSP-based message stitching, with the RM sequences therein generated under several specific constraints.
- The factors affecting the reliability of our slot-based RM detection scheme are discussed, and the computational complexity of SSP-based message stitching is analyzed in detail. Our simulation results validate the advantages of the proposed RM-based URA system in terms of its error probability and computational complexity.

The rest of the paper is organized as follows. Section II describes the system model, while Section III elaborates on the shift property of RM sequences. The proposed RM-based URA scheme is introduced in Section IV, followed by Section V on the factors influencing the RM detection capability and on the complexity of the proposed SSP-based message stitching. Our simulation results are provided in Section VI and finally Section VII concludes the paper.

Notations: Throughout this paper, scalars are represented in lowercase letters. Boldface lowercase and uppercase letters denote column vectors and matrices, respectively. The length-N column vectors of all zeros and all ones are denoted as $\mathbf{0}_N$ and $\mathbf{1}_N$, respectively. The notation $(\cdot)^T$ means the transpose of a matrix or a vector, while $(\cdot)^*$ denotes the complex conjugate. We use $\mathbf{X} = [x_{i,j}]_{i=1,j=1}^{I,J}$ to represent the $(I \times J)$-element matrix made up of the elements $x_{i,j}$, where $i = 1, \cdots, I$ and $j = 1, \cdots, J$. Similarly, the column vector $\mathbf{y} = [y_1, \cdots, y_N]^T$ is written as $\mathbf{y} = [y_j]_{j=1}^{N}$ or $\mathbf{y} = [y(j)]_{j=1}^{N}$ for short. The result of appending the column vector $\mathbf{y}_2 \in \mathbb{C}^{N_2 \times 1}$ to $\mathbf{y}_1 \in \mathbb{C}^{N_1 \times 1}$ has the form of $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2] \in \mathbb{C}^{(N_1+N_2)\times 1}$. The symbol $|\cdot|$ is the cardinality of a set; $\oplus$ represents the modulo-2 addition while $\odot$ is the element-wise multiplication. $\mathcal{O}(\cdot)$ represents the complexity order. The vector $\mathbf{a}_{j-1}^m$ is the m-bit binary representation of the decimal number $(j-1)$, while conversely we have that $j = \langle \mathbf{a}_{j-1}^m \rangle$, i.e., the operation $\langle \cdot \rangle$ converts the binary vector to the corresponding decimal number and then increases the result by one. The notation $x \sim \mathcal{CN}(\mu, \sigma^2)$ indicates that the random variable (r.v.) x follows the circular symmetric complex Gaussian distribution with its probability density function (PDF) being

$$f_{\mathcal{CN}}(x; \mu, \sigma^2) = \frac{1}{\pi\sigma^2} \exp\left(-\frac{|x-\mu|^2}{\sigma^2}\right).$$

II. System Model

A massive uplink connectivity scenario is considered, where a single AP serves a huge pool of K potential users. The users keep silent until they have information to send. Assume that K_a users turn active in a specific communication round, and they are synchronized according to the reference signal broadcast by the AP. The active user set is denoted as $\mathcal{K}_a$ and we have $K_a = |\mathcal{K}_a| \ll K$ because of the sporadic user activity of mMTC. Each active user expects to transmit

B bits of information, denoted as $\mathbf{u}_k \in \{0,1\}^B$ with the aid of N channel uses. The AP has to estimate a set of transmitted messages based on the received signal $\mathbf{y}$, which is represented as $\hat{\mathcal{U}}(\mathbf{y})$. Different from the original URA framework proposed by Polyanskiy, the quasi-static Rayleigh fading channel is considered here, i.e., the channel remains constant within a single slot but varies independently among slots. Besides, since the content of messages rather than their sources is the main focus in the URA system, we measure the performance from the point of messages and adopt the miss-detection rate (MDR) and the false-alarm rate (FAR) as the pivotal metrics, which are respectively specified as

$$\begin{aligned} P_{\text{md}} &\triangleq \frac{1}{K_a}\sum_{k\in\mathcal{K}_a}\Pr\left(\mathbf{u}_k \notin \hat{\mathcal{U}}(\mathbf{y})\right), \\ P_{\text{fa}} &\triangleq \frac{\left|\hat{\mathcal{U}}(\mathbf{y}) \setminus \{\mathbf{u}_k | k\in\mathcal{K}_a\}\right|}{\left|\hat{\mathcal{U}}(\mathbf{y})\right|}. \end{aligned} \tag{1}$$

The error probability of the system is defined as $P_{\text{err}} \triangleq P_{\text{md}} + P_{\text{fa}}$.

The basic system settings of our work mainly follow existing URA proposals [9] [21]. Specifically, under the typical packetized and slotted transmission framework, the N channel uses are equally divided into T slots. The active users split their messages into several information chunks and map each one to a codeword in the common codebook before transmitting every codeword in one of the T slots. The AP carries out its task in two steps: 1) recovers the information chunks after detecting the transmitted codewords in each slot; 2) stitches the chunks of the same message together across different slots.

Since the huge RM sequence space can improve SE and the elegant nested structure of RM sequences facilitates reliable RM detection at a low complexity, we adopt RM sequences for carrying information chunks in slots. Besides, we explore the shift property of RM sequences and exploit it to realize message stitching in a novel and efficient way.

III. The Shift Property of Reed-Muller Sequences

Given an $(m\times m)$-element symmetric binary matrix $\mathbf{P}^m \in \mathbb{Z}_2^{m\times m}$ and an $(m\times 1)$-element binary vector $\mathbf{b}^m \in \mathbb{Z}_2^{m\times 1}$, the length-$2^m$ RM sequence $\mathbf{c}^m$ is generated according to the following generation function [14]:

$$c_j^m = \iota^{(2\mathbf{b}^m + \mathbf{P}^m\mathbf{a}_{j-1}^m)^T\mathbf{a}_{j-1}^m},\ j=1,\ldots,2^m, \tag{2}$$

where ι is the imaginary unit, and $\mathbf{a}_{j-1}^m$ is the m-bit binary representation of $(j-1)$.

To proceed, we present the matrix-vector pair $\{\mathbf{P}^m,\mathbf{b}^m\}$ in a recursive way. Specifically, the lower right $(m-1)\times(m-1)$ sub-matrix of $\mathbf{P}^m$ is denoted as $\mathbf{P}^{m-1}$, while its first column $[p_{m,m},\cdots,p_{m,1}]^T$ or $[p_{m,j}]_{j=m}^1$ for short, is split into $\left[\rho_m;\boldsymbol{\alpha}^{m-1}\right]$ with $\rho_m \triangleq p_{m,m}$ and $\boldsymbol{\alpha}^{m-1} \triangleq [p_{m,j}]_{j=m-1}^1$. Furthermore, we let the lower $(m-1)$ bits of $\mathbf{b}^m = [b_j]_{j=m}^1$ constitute the sub-vector $\mathbf{b}^{m-1}$. To conclude, the recursive representation of $\{\mathbf{P}^m,\mathbf{b}^m\}$ is

$$\mathbf{P}^m = \begin{bmatrix} \rho_m & \left(\boldsymbol{\alpha}^{m-1}\right)^T \\ \boldsymbol{\alpha}^{m-1} & \mathbf{P}^{m-1} \end{bmatrix},\quad \mathbf{b}^m = \begin{bmatrix} b_m \\ \mathbf{b}^{m-1} \end{bmatrix}. \tag{3}$$

Given the pair in (3), we denote the result of shifting it s ($s \in \{0,\cdots,m-1\}$) times as $\left\{\mathbf{P}^{[m,s]},\mathbf{b}^{[m,s]}\right\} \triangleq \mathbf{SF}\left(\{\mathbf{P}^m,\mathbf{b}^m\},s\right)$, which can further be recursively expressed as

$$\left\{\mathbf{P}^{[m,s]},\mathbf{b}^{[m,s]}\right\} = \left\{\begin{bmatrix} \rho_m^s & \left(\boldsymbol{\alpha}^{[m-1,s]}\right)^T \\ \boldsymbol{\alpha}^{[m-1,s]} & \mathbf{P}^{[m-1,s]} \end{bmatrix}, \begin{bmatrix} b_m^s \\ \mathbf{b}^{[m-1,s]} \end{bmatrix}\right\}$$

the same way as in (3). The shift operation $\mathbf{SF}(\cdot)$ is recursively defined in detail as in **Alg. 1**.

Algorithm 1 : Shift Operation $\mathbf{SF}(\cdot)$

Input: $\{\mathbf{P}^m,\mathbf{b}^m\}$, s

1: **if** $s=0$ **then**
2: $\left\{\mathbf{P}^{[m,0]},\mathbf{b}^{[m,0]}\right\} = \{\mathbf{P}^m,\mathbf{b}^m\}$.
3: **else**
4: Obtain $\left\{\mathbf{P}^{[m,s-1]},\mathbf{b}^{[m,s-1]}\right\} = \mathbf{SF}\left(\{\mathbf{P}^m,\mathbf{b}^m\},s-1\right)$;
5: Shift the sub-matrix $\mathbf{P}^{[m-1,s-1]}$ of $\mathbf{P}^{[m,s-1]}$ to the top left corner along the main diagonal while moving the element ρ_m^{s-1} to the bottom right corner, and shift the vectors $\left(\boldsymbol{\alpha}^{[m-1,s-1]}\right)^T$ and $\boldsymbol{\alpha}^{[m-1,s-1]}$ to the last row and last column, respectively, yielding
$$\mathbf{P}^{[m,s]} = \begin{bmatrix} \mathbf{P}^{[m-1,s-1]} & \boldsymbol{\alpha}^{[m-1,s-1]} \\ \left(\boldsymbol{\alpha}^{[m-1,s-1]}\right)^T & \rho_m^{s-1} \end{bmatrix}.$$
6: Move the highest bit of $\mathbf{b}^{[m,s-1]}$ to the lowest position, yielding $\mathbf{b}^{[m,s]} = \left[\mathbf{b}^{[m-1,s-1]};b_m^{s-1}\right]$.
7: **end if**

On this basis, the shift property of RM sequences is summarized in the following theorem.

Theorem 1: Given the pairs before and after a shift as $\left\{\mathbf{P}^{[m,s-1]},\mathbf{b}^{[m,s-1]}\right\}$ and $\left\{\mathbf{P}^{[m,s]},\mathbf{b}^{[m,s]}\right\}$, the RM sequences $\mathbf{c}^{[m,s-1]} = \left[c_j^{s-1}\right]_{j=1}^{2^m}$ and $\mathbf{c}^{[m,s]} = \left[c_j^s\right]_{j=1}^{2^m}$ generated by them satisfy that

$$c_j^s = \begin{cases} c_{(j+1)/2}^{s-1}, & j=1,3,\cdots,2^m-1, \\ c_{j/2+2^{m-1}}^{s-1}, & j=2,4,\cdots,2^m, \end{cases} \tag{4}$$

i.e., $\mathbf{c}^{[m,s]}$ can be obtained by partitioning $\mathbf{c}^{[m,s-1]}$ in half and alternating the elements in the two subsequences.

Proof: Please refer to Appendix A. ∎

An example of the above shift property is illustrated in Fig. 1. For convenience, we also refer to the transformation from $\mathbf{c}^{[m,s-1]}$ to $\mathbf{c}^{[m,s]}$ as a shift and denote it as $\mathbf{c}^{[m,s]} = \mathbf{SF}\left(\mathbf{c}^{[m,s-1]},1\right)$. More generally, the notation $\mathbf{c}^{[m,s_2]} = \mathbf{SF}\left(\mathbf{c}^{[m,s_1]},s_2-s_1\right), 0 \leqslant s_1 < s_2 \leqslant m-1$, represents that the output $\mathbf{c}^{[m,s_2]}$ is obtained by repetitively shifting $\mathbf{c}^{[m,s_1]}$ (s_2-s_1) times.

To proceed, we impose the following constraints on the matrix-vector pairs:

1) The j-th entry of the vector $\mathbf{b}^{[m,0]}$ equals the modulo-2 sum of the elements in the j-th row of the upper triangular

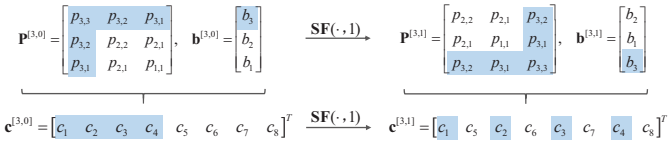


Fig. 1: *An example of the shift property of RM sequences.*

portion of the matrix $\mathbf{P}^{[m,0]}$, yielding:

$$\forall j \in \{1, \cdots, m\},\ b_j^{[m,0]} = \rho_j^0 \oplus \alpha_1^{[j-1,0]} \oplus \cdots \oplus \alpha_{j-1}^{[j-1,0]}. \tag{5}$$

2) For any one of the shifted versions of $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\}$, there exists at least one element in $\mathbf{b}^{[m,s]}$ not satisfying the above equation, which is formulated as:

$$\begin{aligned} &\forall s \in \{1, \cdots, m-1\}, \quad \exists j \in \{1, \cdots, m\}, \\ &b_j^{[m,s]} \neq \rho_j^s \oplus \alpha_1^{[j-1,s]} \oplus \cdots \oplus \alpha_{j-1}^{[j-1,s]}. \end{aligned} \tag{6}$$

The number of such pairs is given in **Theorem 2**.

Theorem 2: When the length of RM sequences is 2^m and the matrix-vector pairs satisfying the constraints in (5-6) constitute the set $\mathcal{P}$, the cardinality of $\mathcal{P}$ equals

$$|\mathcal{P}| = 2^{\frac{m(m+1)}{2}} + \sum_{n=1}^{m-1} (-1)^n \binom{m-1}{n} \cdot 2^{\frac{m(m+1)-n(2m-n-1)}{2}}. \tag{7}$$

Proof: Please refer to Appendix B. ∎

On this basis, we intend to construct a common RM codebook using the pairs in $\mathcal{P}$. Then, if a shifted version of the RM sequence in this codebook is transmitted, say $\mathbf{c}^{[m,s]}$, the AP can readily recover $\left\{\mathbf{P}^{[m,s]}, \mathbf{b}^{[m,s]}\right\}$ by performing Alg. $\mathbf{\Phi}$, and can also estimate the number of shifts s and restore the original pair $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\}$ by checking how many reverse shifts (denoted as $\mathbf{SF}^{-1}(\cdot)$) are required for meeting the constraint in (5). Furthermore, we design a set of sparse patterns called sparse shift patterns (SSPs) and the active users have to transmit their RM sequences following their chosen SSPs, including the specific numbers of shifts before transmission and the specific slots used for transmission. In this way, once the active users' SSPs are recovered at the AP, the message stitching operation may be accomplished exploiting the clues embedded in the SSPs. This is exactly the main philosophy of our proposed RM-based URA scheme, which will be further detailed in the next section.

IV. Unsourced Massive Random Access Scheme using Reed-Muller Sequences

The structure of the transmitters in our proposed URA scheme is depicted in Fig. 2. The active user k first splits its information bits $\mathbf{u}_k$ into two segments, i.e., $\mathbf{u}_k^{(\mathrm{RM})}$ for generating RM sequences and $\mathbf{u}_k^{(\mathrm{SP})}$ for determining the SSP. Then, $\mathbf{u}_k^{(\mathrm{RM})}$ is further divided into N_c chunks and each chunk is mapped to an RM sequence in our common RM codebook $\mathcal{C}$. By contrast, $\mathbf{u}_k^{(\mathrm{SP})}$ is used for selecting an SSP from the common pattern set $\mathcal{D}$. After that, the active user k carries out sparse shift mapping (SSM) according to the chosen SSP.

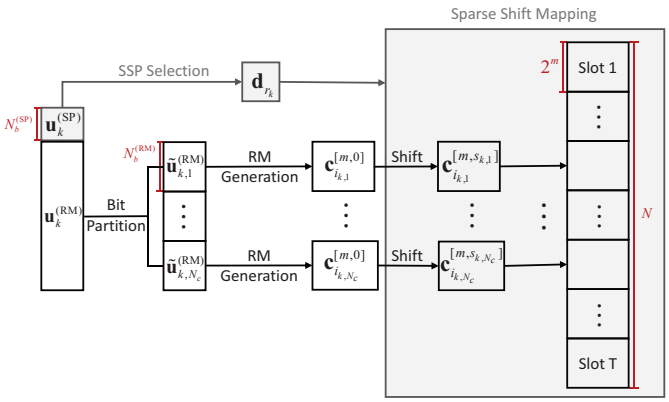


Fig. 2: *The structure of the transmitters in our proposed URA scheme.*

In the sequel, we introduce every module of the transmitter in further detail.

A. Bit Partitioning and RM Generation

Based on the value of $|\mathcal{P}|$ in (7), the number of information bits that a single length-2^m RM sequence can carry equals $N_b^{(\mathrm{RM})} = \lfloor \log_2(|\mathcal{P}|) \rfloor$. Hence, we randomly choose $2^{N_b^{(\mathrm{RM})}}$ matrix-vector pairs from $\mathcal{P}$ and substitute them into the RM generation function in (2). The resultant RM sequences are stored in the common RM codebook $\mathcal{C} \triangleq \left[\mathbf{c}_i^{[m,0]}\right]_{i=1}^{2^{N_b^{(\mathrm{RM})}}}$.

To match its information message $\mathbf{u}_k^{(\mathrm{RM})}$ to the RM sequences in $\mathcal{C}$, the active user k first splits $\mathbf{u}_k^{(\mathrm{RM})}$ into N_c chunks, each having the length of $N_b^{(\mathrm{RM})}$ bits. Next, the active user k maps the n_c-th chunk $\tilde{\mathbf{u}}_{k,n_c}^{(\mathrm{RM})}$ to the index i_{k,n_c} following $\tilde{\mathbf{u}}_{k,n_c}^{(\mathrm{RM})} = \mathbf{a}_{i_{k,n_c}-1}^{N_b^{(\mathrm{RM})}}$, which is denoted as $i_{k,n_c} = \langle \tilde{\mathbf{u}}_{k,n_c}^{(\mathrm{RM})} \rangle$ for simplicity. Thus, the information chunk $\tilde{\mathbf{u}}_{k,n_c}^{(\mathrm{RM})}$ corresponds to the RM sequence $\mathbf{c}_{i_{k,n_c}}^{[m,0]}$. Given the N_c mapped RM sequences, the active user k proceeds to perform SSM.

B. Sparse Shift Pattern Design and Sparse Shift Mapping

During SSM, the active user has to shift every RM sequence several times before transmitting it over one of the T available slots, where the specific number of shifts and the specific slot for transmission are determined by the SSP selected. Specifically, each SSP $\mathbf{d}_r = [d_r(t)]_{t=1}^T, r = 1, \cdots, N_{\mathrm{SP}}$, is a length-$T$ vector with its entries drawn from the set $\{-1, 0, 1, \cdots, m-1\}$, and N_{SP} represents the total number of valid SSPs. Here, $d_r(t) = -1$ represents that the RM sequence is not sent in slot t, while $d_r(t) = 0$ indicates that the RM sequence is transmitted without any shift in slot t, and finally $d_r(t) = s, s = 1, \cdots, m-1$, requires the RM sequence to be shifted s times before being transmitted in slot t.

The SSPs here serve as the guidance to identify the RM sequences transmitted by the same active user in different slots. To this end, the SSPs have to meet the following conditions:

C1: Every active user only transmits N_c RM sequences during a total of T slots, and hence there are exactly N_c entries having values different from minus one in each SSP, i.e., we have $|\mathcal{T}_r| = N_c$, where $\mathcal{T}_r \triangleq \{t | d_r(t) \neq -1\}$.

C2: To recognize the SSPs of active users, the RM sequences transmitted in the same slot by the active users having different SSPs have to be shifted by a different number of times, i.e., $|\mathcal{T}_{t,s}| \leq 1$, where $\mathcal{T}_{t,s} \triangleq \{r|d_r(t) = s\}, s = 0, \cdots, m-1$.

Besides, since all the active users select their SSPs from a common pattern set, it is expected to maximize the number N_{SP} of available SSPs, thus reducing the pattern collision rate. To conclude, the construction of the SSP set can be formulated as the following optimization problem:

$$\begin{aligned} &\max_{d_{r,t}} N_{\text{SP}} \\ &s.t.\ |\mathcal{T}_r| = N_c, \text{ where } \mathcal{T}_r \triangleq \{t|d_r(t) \neq -1\} \\ &\quad |\mathcal{T}_{t,s}| \leq 1, \text{ where } \mathcal{T}_{t,s} \triangleq \{r|d_r(t) = s\}, s = 0, \cdots, m-1 \end{aligned}.$$

We now proceed by conceiving a heuristic algorithm for constructing the SSP set, which consists of two steps. Firstly, we construct a binary matrix satisfying the following conditions: 1) its number of rows is T; 2) its column weights, i.e., the numbers of ones in the columns, is equal to N_c; 3) its row weights are not larger than m. We aim for constructing such a matrix with the largest number of columns. Next, we replace all the zeros in the matrix by minus ones, and randomly pick elements from the set $\{0, 1, \cdots, m-1\}$ without replacement to substitute the ones in each row of the matrix. The details of the algorithm are provided in **Alg. 2**.

Algorithm 2 : The Construction of the SSP Set

Input: m, T, N_c

1: Initialize the matrix $\mathbf{D} = [\mathbf{1}_{N_c}; \mathbf{0}_{T-N_c}]$.
2: **while** more than N_c rows in the matrix $\mathbf{D}$ have weights smaller than m **do**
3: Initialize that $\mathbf{d} = \mathbf{0}_T$.
4: Randomly select N_c rows with the smallest row weights and denote their indices as $t_1, \cdots, t_{N_c}$.
5: Set $d_{t_{n_c}} = 1$, where $n_c = 1, \cdots, N_c$.
6: Update $\mathbf{D}$ by concatenating $\mathbf{d}$ to its end along the second dimension, i.e., $\mathbf{D} = [\mathbf{D}, \mathbf{d}]$.
7: **end while**
8: Replace all the zeros in the matrix $\mathbf{D}$ with minus ones.
9: **for** $t = 1 : T$ **do**
10: Search through the t-th row of the matrix $\mathbf{D}$ for the elements with a value of one, and let their column indices form the collection $\mathcal{T}_t$, i.e., $\mathcal{T}_t \triangleq \{r|d_r(t) = 1\}$.
11: Randomly sample $|\mathcal{T}_t|$ elements from the collection $\{0, 1, \cdots, m-1\}$ without replacement and then substitute them for $d_r(t), \forall r \in \mathcal{T}_t$ on a one-to-one basis.
12: **end for**
13: Let N_{SP} equal the number of columns of the matrix $\mathbf{D}$.
14: **return** $\mathbf{D}$, N_{SP}.

After executing **Alg. 2**, the size of the information segment $\mathbf{u}_k^{(\text{SP})}$ is fixed to $N_b^{(\text{SP})} = \lfloor \log_2 N_{\text{SP}} \rfloor$, and we have to select $\tilde{N}_{\text{SP}} = 2^{N_b^{(\text{SP})}}$ SSPs from $\mathbf{D}$ for constructing the common pattern set $\mathcal{D}$. Recall that the SSPs determine the distribution of RM sequences across slots, which should be as balanced as possible since we have to minimize the chances of numerous RM sequences gathering together to control the multi-user interference (MUI). Since **Alg. 2** is based on the greedy strategy, and Line 4 therein ensures that the matrix $\mathbf{D}$ obtained at the end of each iteration has the most uniform row weights, we can construct $\mathcal{D}$ by extracting the first $2^{N_b^{(\text{SP})}}$ columns of $\mathbf{D}$, yielding $\mathcal{D} \triangleq [\mathbf{d}_r]_{r=1}^{2^{N_b^{(\text{SP})}}}$. An example of $\mathcal{D}$ is shown in Fig. 3.

$$\mathbf{D} = \underbrace{\begin{bmatrix} 1 & -1 & 0 & -1 \\ 1 & -1 & -1 & 2 \\ -1 & 0 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & 2 & -1 & -1 \end{bmatrix}}_{=\mathcal{D}} \begin{matrix} -1 & 2 & -1 \\ -1 & 0 & -1 \\ 1 & -1 & 2 \\ -1 & -1 & 1 \\ 0 & -1 & -1 \end{matrix}$$

Fig. 3: *An example of the output of **Alg. 2** when $m = 3$, $T = 5$ and $N_c = 2$ is given as $\mathbf{D}$ with $N_{SP} = 7$. Thus, we have $N_b^{(SP)} = 2$ and $\mathcal{D}$ is constructed by the first four columns of $\mathbf{D}$.*

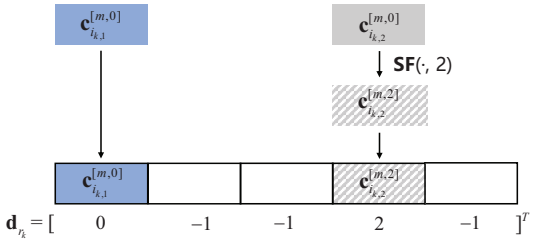


Fig. 4: *The process of SSM when the SSP is chosen as $[0, -1, -1, 2, -1]^T$ from the set $\mathcal{D}$ in Fig. 3.*

On this basis, the active user k maps its information bits $\mathbf{u}_k^{(\text{SP})}$ to the index r_k following $r_k = \langle \mathbf{u}_k^{(\text{SP})} \rangle$, and then performs SSM relying on the SSP $\mathbf{d}_{r_k}$. Fig. 4 illustrates the process of SSM when the SSP is chosen as $[0, -1, -1, 2, -1]^T$ from the set $\mathcal{D}$ in Fig. 3. In general, given the SSP $\mathbf{d}_{r_k}$, the signal to be sent in slot t is formulated as

$$\mathbf{x}_{k,t} = \begin{cases} \mathbf{0}_{2^m}, & \text{if } d_{r_k}(t) = -1, \\ \mathbf{c}_{i_{k,n_c}}^{[m,s]} = \mathbf{SF}\left(\mathbf{c}_{i_{k,n_c}}^{[m,0]}, s\right), & \text{if } d_{r_k}(t) = s \text{ and } \tau_{r_k,n_c} = t, \end{cases}$$

where τ_{r_k,n_c} is the index of the n_c-th element in $\mathbf{d}_{r_k}$ having a value unequal to -1, and $s = 0, \cdots, m-1$. Assuming that the energy-per-bit E_b is normalized to 1, the complete signal sent by the active user k is expressed as

$$\tilde{\mathbf{x}}_k = \sqrt{\frac{B}{2^m N_c}} [\mathbf{x}_{k,1}; \cdots; \mathbf{x}_{k,T}], \tag{8}$$

where B is the size of the message $\mathbf{u}_k$ given by $B = N_b^{(\text{RM})} N_c + N_b^{(\text{SP})}$. In conjunction with the number of channel uses being $N = 2^m T$, the SE of the system equals

$$\eta \triangleq \frac{B}{N} = \frac{N_b^{(\text{RM})} N_c + N_b^{(\text{SP})}}{2^m T} \text{ bit/c.u./user.} \tag{9}$$

C. Slot-Based RM Detection and Message Stitching

After receiving the signal, the AP first performs RM detection in a slot-by-slot manner. Specifically, the signal received in slot t can be expressed as

$$y_{t,j} = \sum_{k \in \mathcal{K}_a} h_{k,t} \cdot \sqrt{\frac{B}{2^m N_c}} x_{k,t,j} + e_{t,j}, \ j = 1, \cdots, 2^m, \tag{10}$$

where $h_{k,t} \sim \mathcal{CN}(0,1)$ represents the channel coefficient between the active user k and the AP in slot t, while $e_{t,j} \sim \mathcal{CN}(0, N_0)$ is the complex additive white Gaussian noise (AWGN). Here, we adopt Alg. $\mathbf{\Phi}$ proposed in [19] for efficient and reliable RM detection exploiting the nested structure of RM sequences. Among the output pairs of Alg. $\mathbf{\Phi}$, some correspond to the RM sequences that were shifted before transmission. As mentioned in Section III, under the constraints in (5-6), we can restore the pairs before shifts and estimate the corresponding number of shifts. The detailed process is summarized in **Alg. 3**.

Algorithm 3 : RM Detection and Shift Times Estimation

Input: $\mathbf{y}_t$

Input $\mathbf{y}_t$ to Alg. $\mathbf{\Phi}$ for obtaining the estimated pairs $\{\hat{\mathbf{P}}_{t,n}, \hat{\mathbf{b}}_{t,n}\}, n = 1, \cdots, N_t'$.
Initialize that $N_t = 0$.
for $n = 1 : N_t'$ **do**
  Initialize that $s = 0$.
  while $\exists j$, $\hat{b}_{t,n,j} \neq \hat{\rho}_{t,n,j} \oplus \hat{\boldsymbol{\alpha}}_{t,n}^{j-1}$ and $s < m$ **do**
    Let $\{\hat{\mathbf{P}}_{t,n}, \hat{\mathbf{b}}_{t,n}\} = \mathbf{SF}^{-1}(\{\hat{\mathbf{P}}_{t,n}, \hat{\mathbf{b}}_{t,n}\}, 1)$ and $s = s + 1$.
  end while
  if $s < m$ **then**
    $N_t = N_t + 1$, and $\hat{s}_{t,N_t} = s$.
    Substitute $\{\hat{\mathbf{P}}_{t,n}, \hat{\mathbf{b}}_{t,n}\}$ to (2) for generating the RM sequence $\hat{\mathbf{c}}_{t,N_t}$, and then recover $\hat{\mathbf{u}}_{t,N_t}$ according to the index of $\hat{\mathbf{c}}_{t,N_t}$ in the common RM codebook $\mathcal{C}$.
  end if
end for
return All N_t estimated information chunks and the corresponding shift times in slot t, i.e., $\{\hat{\mathbf{u}}_{t,n}, \hat{s}_{t,n}\}, n = 1, \cdots, N_t$.

Next, the AP has to stitch together the chunks of the same messages across different slots. We first consider the case where all the active users selected different SSPs, while the more complex situation where SSP collisions occur will be discussed later.

The AP checks the estimated results one by one. When checking the n-th estimated result in slot t, i.e., $\{\hat{\mathbf{u}}_{t,n}, \hat{s}_{t,n}\}$, the AP executes the following steps:

Step 1: Under the condition **C2** stipulated for designing SSPs, there can be at most one SSP whose t-th element equals $\hat{s}_{t,n}$. If no such SSP exists in $\mathcal{D}$, then the result $\{\hat{\mathbf{u}}_{t,n}, \hat{s}_{t,n}\}$ must be erroneous, and the AP just removes it from the result pool before proceeding to check the next estimated result. But if the SSP with the t-th element being $\hat{s}_{t,n}$, denoted as $\mathbf{d}_{\hat{r}_{t,n}}$, is contained in $\mathcal{D}$, then it is the one selected by the active user $\hat{k}_{t,n}$ who sent the chunk $\hat{\mathbf{u}}_{t,n}$. Note that the symbol $\hat{k}_{t,n}$ is introduced here for ease of expression, and the specific identity of the active user who sent $\hat{\mathbf{u}}_{t,n}$ is of no interest in the URA scenario. On this basis, the AP can recover the information segment used for choosing this SSP as $\mathbf{a}_{\hat{r}_{t,n}-1}^{N_b^{(\mathrm{SP})}}$.

Step 2: The AP searches the SSP $\mathbf{d}_{\hat{r}_{t,n}}$ for the second element with the value unequal to minus one and denotes its index as $\tau_{t,n}(2)$. If there is one estimated result in slot $\tau_{t,n}(2)$ whose shift times equals $d_{\hat{r}_{t,n}}(\tau_{t,n}(2))$, then it is the second chunk sent by the active user $\hat{k}_{t,n}$. In the other case where no such results exist in slot $\tau_{t,n}(2)$, the AP deletes $\{\hat{\mathbf{u}}_{t,n}, \hat{s}_{t,n}\}$ and heads for checking the next estimated result. In this way, as long as the information chunks of the active user $\hat{k}_{t,n}$ is recovered successfully, the AP can identify them easily.

Step 3: By stitching $\mathbf{a}_{\hat{r}_{t,n}-1}^{N_b^{(\mathrm{SP})}}$, $\hat{\mathbf{u}}_{t,n}$ and the $(N_c - 1)$ chunks found in Step 2 in order, the complete message of the active user $\hat{k}_{t,n}$ is recovered. After that, the AP removes all the estimated results belonging to this message from the result pool and continues to check the next estimated result.

The AP repeats the above steps until the result pool is empty, thus completing all the message stitching tasks. Fig. 5 illustrates an example of the SSP-based message stitching.

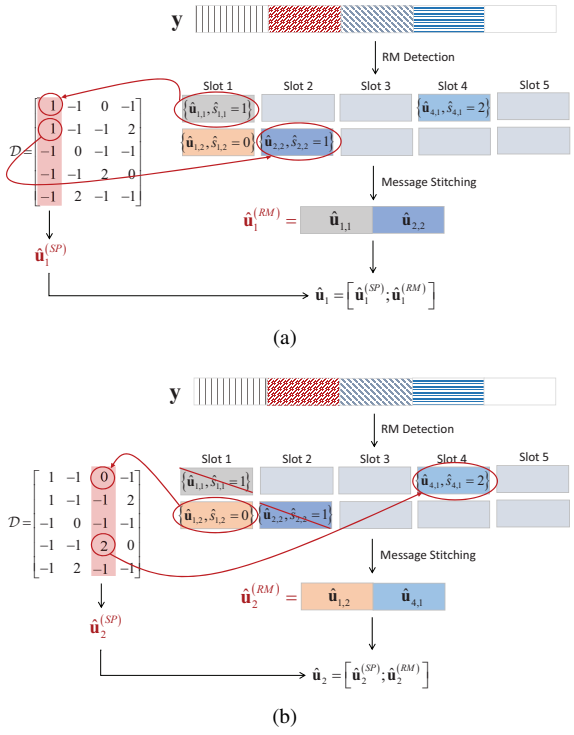


Fig. 5: *The illustration of the SSP-based message stitching. We set that $m = 3$, $T = 5$ and $N_c = 2$ and use the common pattern set $\mathcal{D}$ in Fig. 3.*

It is inevitable that several users may choose the same SSP, especially when the number of active users is high. Once SSP collisions occur, it is hard to distinguish these users' information chunks solely depending on their SSPs and thus we resort to parity checks for help. Specifically, when the parity check allocation is preset as $\mathbf{l} = [l_{n_c}]_{n_c=2}^{N_c}$, the active user first splits its message into N_c chunks with the first and the n_c-th chunks respectively having $N_b^{(\mathrm{RM})}$ and $(N_b^{(\mathrm{RM})} - l_{n_c})$ information bits. Next, it generates l_{n_c} parity check bits by entering the first to the $(n_c - 1)$st chunks into a linear block encoder. The resultant parity check bits are appended to the

n_c-th chunk to form the n_c-th sub-block. Then, the active user maps each sub-block to an RM sequence in $\mathcal{C}$ and proceeds to carry out the SSM process. In the process of message stitching, if two or more estimated results in a single slot have the same number of shifts, the AP will determine the desired one by verifying parity checks. More details on the parity checks can be found in [9]. Since the value of $N_b^{(\mathrm{RM})}$ is fixed, introducing parity checks would reduce the number of information bits, hence degrading the SE of the system. Given $\mathbf{l} = [l_{n_c}]_{n_c=2}^{N_c}$, the SE is given by

$$\eta(\mathbf{l}) = \frac{N_b^{(\mathrm{RM})} N_c + N_b^{(\mathrm{SP})} - \sum_{n_c=2}^{N_c} l_{n_c}}{2^m T} \text{ bit/c.u./user}. \quad (11)$$

Note that the parity checks here are used for identifying the messages of the active users suffering from SSP collisions, while others can be stitched together directly relying on the SSPs. Hence, compared to the schemes that only utilize tree coding for message stitching, combining SSPs and parity checks requires much fewer parity bits and has much lower complexity.

V. Performance Analysis

The performance of our proposed URA system depends both on the reliability of RM detection in each slot and on the efficiency of message stitching across different slots. In this section, we first discuss the factors affecting the slot-based RM detection and then analyze the computational complexity of the SSP-based message stitching in detail.

A. The Performance of Slot-Based RM Detection

The capability of slot-based RM detection is affected by two factors: the MUI in each slot and the signal-to-noise ratio (SNR) of the received signal. We now discuss them separately.

1) MUI: The amount of MUI in a slot depends on the number of superimposed RM sequences. Hence, it is necessary to analyze the distribution of RM sequences over slots.

Assume that each SSP in the common pattern set $\mathcal{D}$ is chosen with the same chance of $1/\tilde{N}_{\mathrm{SP}}$. The condition **C2** ensures that at most m SSPs might assign RM sequences to the same slot. Hence, when the active users have different SSPs, the number of the RM sequences transmitted in the same slot, denoted by the r.v. Θ, obeys the hypergeometric distribution having the probability mass function (PMF) of

$$P_{\mathrm{s,noc}}(\Theta = \theta) = \frac{\binom{m}{\theta}\binom{\tilde{N}_{\mathrm{SP}}-m}{K_a-\theta}}{\binom{\tilde{N}_{\mathrm{SP}}}{K_a}}, \quad (12)$$

where $\theta = 0, \cdots, m$ and $K_a = 0, \cdots, \tilde{N}_{\mathrm{SP}}$. Its expectation and variance have the form of

$$\begin{aligned} \mathbb{E}_{\mathrm{s,noc}}(\Theta) &= \frac{K_a m}{\tilde{N}_{\mathrm{SP}}}, \\ \mathbb{D}_{\mathrm{s,noc}}(\Theta) &= \frac{K_a m}{\tilde{N}_{\mathrm{SP}}}\big(1 - \frac{m}{\tilde{N}_{\mathrm{SP}}}\big)\frac{\tilde{N}_{\mathrm{SP}} - K_a}{\tilde{N}_{\mathrm{SP}} - 1}. \end{aligned} \quad (13)$$

When SSP collisions might occur, the r.v. Θ obeys the binomial distribution with the PMF being

$$P_{\mathrm{s,c}}(\Theta = \theta) = \binom{K_a}{\theta}\left(\frac{m}{\tilde{N}_{\mathrm{SP}}}\right)^{\theta}\left(1 - \frac{m}{\tilde{N}_{\mathrm{SP}}}\right)^{K_a-\theta}, \quad (14)$$

and its expectation and variance are expressed as

$$\mathbb{E}_{\mathrm{s,c}}(\Theta) = \frac{K_a m}{\tilde{N}_{\mathrm{SP}}}, \quad \mathbb{D}_{\mathrm{s,c}}(\Theta) = \frac{K_a m}{\tilde{N}_{\mathrm{SP}}}\left(1 - \frac{m}{\tilde{N}_{\mathrm{SP}}}\right). \quad (15)$$

Upon comparing (13) and (15), we find that $\mathbb{E}_{\mathrm{s,noc}}(\Theta) = \mathbb{E}_{\mathrm{s,c}}(\Theta)$, while $\mathbb{D}_{\mathrm{s,noc}}(\Theta) \leq \mathbb{D}_{\mathrm{s,c}}(\Theta)$. This reveals that the average number of RM sequences in a slot is the same but RM sequences are distributed more evenly over the slots when the active users choose different SSPs. If SSP collisions occur, the chance of a large number of RM sequences overlapping in the same slot grows, thus increasing the MUI in these slots.

To enhance the RM detection capability by reducing MUI, we have to consider the elements affecting the distribution of RM sequences. Eq. (12) and (14) reveal that the following two factors are critical:

(i) The Number of Active Users K_a: The effect of K_a is intuitive. More active users require more sequences to be sent, and SSP collisions would occur more frequently.

(ii) The Value of $m/\tilde{N}_{\mathrm{SP}}$: In the sequel, we focus on the situation where Θ obeys the PMF in (14), and Eq. (12) can be closely approximated by (14) when $\tilde{N}_{\mathrm{SP}} \gg K_a$. It is plausible that the performance of Alg. $\mathbf{\Phi}$ would degrade dramatically if more than θ^* RM sequences overlap in the input signal. According to Hoeffding's inequality [23], when $\theta^* \geq K_a m/\tilde{N}_{\mathrm{SP}}$, we have

$$P_{\mathrm{s,c}}(\Theta \geq \theta^*) \leq \exp\Big(-2K_a\big(\frac{\theta^*}{K_a} - \frac{m}{\tilde{N}_{\mathrm{SP}}}\big)^2\Big). \quad (16)$$

This reveals that the upper bound of $P_{\mathrm{s,c}}(\Theta \geq \theta^*)$ grows monotonically upon increasing $m/\tilde{N}_{\mathrm{SP}}$. Hence, to guarantee RM detection capability, the value of $m/\tilde{N}_{\mathrm{SP}}$ should be as low as possible.

The impact of $m/\tilde{N}_{\mathrm{SP}}$ is hard to tell at first sight and hence we link it to the number of chunks and slots, i.e., with N_c and T. When T is divisible by N_c, the number of columns in $\mathbf{D}$, i.e., N_{SP}, is equal to mT/N_c. The set $\mathcal{D}$ is obtained by drawing the first $\tilde{N}_{\mathrm{SP}}$ columns from $\mathbf{D}$, and hence $m/\tilde{N}_{\mathrm{SP}}$ is proportional to N_c/T. By contrast, if T is not divisible by N_c, it is difficult to give an explicit expression for N_{SP}, but it is plausible that N_{SP} is still proportional to T/N_c, and $m/\tilde{N}_{\mathrm{SP}}$ is monotonically non-decreasing upon increasing N_c/T. Therefore, the value of $m/\tilde{N}_{\mathrm{SP}}$ is closely related to N_c/T. The impact of N_c/T on the distribution of RM sequences is evident. When N_c is given, RM sequences tend to be more widely dispersed over slots upon increasing T, thus mitigating MUI. However, we cannot reduce the value of N_c/T without limitation, since there is a tradeoff with SE according to (9). This suggests that both the RM detection capability and SE have to be considered when choosing the values of N_c and T.

2) SNR: Since we normalize the energy-per-bit E_b to 1, the transmit power is proportional to the number of information bits, as shown in (8). Therefore, the SNR of the received signal depends both on the noise power density (N_0) and on the size of messages. When we introduce parity checks for dealing with SSP collisions, a tradeoff between the performance of RM detection and message stitching arises. Specifically, more

parity check bits are more likely to distinguish the information chunks of the messages suffering from SSP collisions. However, the value of $N_b^{(\mathrm{RM})}$ is fixed, and hence fewer information bits can be transmitted as the number of parity check bits increases, which reduces both the RM detection capability and SE.

In summary, we can draw the following conclusions from the above analysis:

- The parameters affecting the slot-based RM detection capability include the number of active users K_a, the ratio N_c/T, and the noise power density N_0.
- The tradeoff between the RM detection performance and SE has to be considered when determining the values of N_c and T.
- When parity check bits are introduced for resolving SSP collisions, their number has to be optimized for attaining both the reliable parity checks and the satisfactory RM detection capability.

B. Computational Complexity of Message Stitching

1) SSP-Based Message Stitching: We now derive the computational complexity of the SSP-based message stitching in the case of no SSP collisions. Following the process in Fig. 5, the AP first finds the desired one from a total of $\tilde{N}_{\mathrm{SP}}$ SSPs in $\mathcal{D}$, which is accomplished using the conventional sequential search with an average complexity order of $\mathcal{O}(\frac{\tilde{N}_{\mathrm{SP}}}{2})$. Next, the AP has to find the remaining (N_c-1) information chunks based on the slot indices and on the number of shifts indicated by the SSP. Given the MDR and the FAR of RM detection as $P_{\mathrm{md}}^{(\mathrm{RM})}$ and $P_{\mathrm{fa}}^{(\mathrm{RM})}$, respectively, the average number of the detected results in each slot is equal to

$$\hat{N}^{(\mathrm{RM})} = \frac{N_c K_a (1 - P_{\mathrm{md}}^{(\mathrm{RM})})}{T(1 - P_{\mathrm{fa}}^{(\mathrm{RM})})}. \tag{17}$$

Hence, the AP has to search through $\hat{N}^{(\mathrm{RM})}$ results for finding the one having the specific number of shifts, which introduces the average complexity order of $\mathcal{O}(\frac{\hat{N}^{(\mathrm{RM})}}{2})$ for each chunk. Note that in this step, we neglect that the detected results belonging to the successfully stitched messages are removed from the result pool. To sum up, the complexity of stitching a single message together successfully equals $\mathcal{O}\left[\frac{1}{2}\left(\tilde{N}_{\mathrm{SP}} + \hat{N}^{(\mathrm{RM})}(N_c-1)\right)\right]$. The number of successfully stitched messages, including those sent by the active users and the others causing false alarms, is $\hat{N} = |\hat{\mathcal{U}}(\mathbf{y})| = \frac{K_a(1-P_{\mathrm{md}})}{1-P_{\mathrm{fa}}}$, where P_{md} and P_{fa} are defined in (1). Thus, the complexity of those messages that successfully complete stitching is

$$C_{\mathrm{SSP,comp}} = \mathcal{O}\left[\frac{1}{2}\hat{N}\left(\tilde{N}_{\mathrm{SP}} + \hat{N}^{(\mathrm{RM})}(N_c-1)\right)\right]. \tag{18}$$

The message stitching processes may also be curtailed unexpectedly in the following cases:

(i) The first chunk of an active message is detected successfully, and the corresponding SSP is found. However, at least one of its remaining (N_c-1) chunks fails to be detected.
Provided that the MDR of each chunk equals $P_{\mathrm{md}}^{(\mathrm{RM})}$, the probability that the message stitching ends at the n_c-th chunk is $P_{\mathrm{md}}^{(\mathrm{RM})}(1-P_{\mathrm{md}}^{(\mathrm{RM})})^{n_c-1}$, which has the complexity of $\mathcal{O}\left[\frac{\tilde{N}_{\mathrm{SP}}+\hat{N}^{(\mathrm{RM})}n_c}{2}\right]$ with $n_c = 2,\cdots,N_c$. Hence, the average computational complexity in this case is given by

$$\begin{aligned}
&C_{\mathrm{SSP,uncom}}^{(1)}\\
=&P_{\mathrm{md}}K_a\sum_{n_c=2}^{N_c}\left\{\frac{1}{2}\left(\tilde{N}_{\mathrm{SP}}+\hat{N}^{(\mathrm{RM})}n_c\right)P_{\mathrm{md}}^{(\mathrm{RM})}\left(1-P_{\mathrm{md}}^{(\mathrm{RM})}\right)^{n_c-1}\right\}\\
=&\frac{1}{2}P_{\mathrm{md}}K_a\Bigg\{\left(1-P_{\mathrm{md}}^{(\mathrm{RM})}\right)\left[1-\left(1-P_{\mathrm{md}}^{(\mathrm{RM})}\right)^{N_c-1}\right]\cdot\tilde{N}_{\mathrm{SP}}\\
&+\frac{1-\left(P_{\mathrm{md}}^{(\mathrm{RM})}\right)^2-\left(1+N_cP_{\mathrm{md}}^{(\mathrm{RM})}\right)\left(1-P_{\mathrm{md}}^{(\mathrm{RM})}\right)^{N_c}}{P_{\mathrm{md}}^{(\mathrm{RM})}}\hat{N}^{(\mathrm{RM})}\Bigg\}.
\end{aligned} \tag{19}$$

(ii) The desired SSP is not found since the chunk that trigger the message stitching process belongs to an undetected active message.
Specifically, if an active message fails to complete stitching but its n_c-th chunk is successfully detected, where $n_c = 2,\cdots,N_c$, then this chunk would remain in the result pool until the AP checks it and searches the SSP set according to its position and the number of shifts. However, it will fail to find the desired SSP. Specifically, let us assume that this message corresponds to the SSP $\mathbf{d}$, and its n_c-th chunk is detected in slot t with the number of shifts being $\hat{s}$, i.e., $d(t)=\hat{s}$. Then, owing to the condition **C2** of constructing $\mathcal{D}$, it is impossible to have another SSP $\mathbf{d}'$ satisfying $d'(t)=\hat{s}$ and meanwhile $d'(t)$ is its first entry not equal to minus one. The number of this kind of uncompleted message stitching processes equals the number of successfully detected chunks among the second to the N_c-th chunks of the missed active messages, which is formulated as $P_{\mathrm{md}}K_a(N_c-1)(1-P_{\mathrm{md}}^{(\mathrm{RM})})$. On this basis, the average computational complexity here is

$$C_{\mathrm{SSP,uncom}}^{(2)} = \mathcal{O}\big[P_{\mathrm{md}}K_a(N_c-1)(1-P_{\mathrm{md}}^{(\mathrm{RM})})\tilde{N}_{\mathrm{SP}}\big]. \tag{20}$$

(iii) The desired SSP is not found since the chunks that the AP checks are incorrect.
In this case, each uncompleted message stitching has the complexity order of $\mathcal{O}(\tilde{N}_{\mathrm{SP}})$. Given $P_{\mathrm{fa}}^{(\mathrm{RM})}$, the average number of incorrect detection results in each slot is $P_{\mathrm{fa}}^{(\mathrm{RM})}\hat{N}^{(\mathrm{RM})}$. Note that there is at least one incorrect detection result in each message that belongs to false alarms, and these results are removed after this message completes stitching. Thus, the average number of incorrect detection results that may trigger message stitching processes should be no more than $\left(P_{\mathrm{fa}}^{(\mathrm{RM})}\hat{N}^{(\mathrm{RM})}T - P_{\mathrm{fa}}\hat{N}\right)$. To summarize, the expected complexity here euqals

$$C_{\mathrm{SSP,uncom}}^{(3)} = \mathcal{O}\left[\left(P_{\mathrm{fa}}^{(\mathrm{RM})}\hat{N}^{(\mathrm{RM})}T - P_{\mathrm{fa}}\hat{N}\right)\tilde{N}_{\mathrm{SP}}\right]. \tag{21}$$

Amalgamating all the above results yields **Theorem 3**.

Theorem 3: The average computational complexity of SSP-

based message stitching is given by

$$C_{\text{SSP}} = C_{\text{SSP,com}} + C^{(1)}_{\text{SSP,uncom}} + C^{(2)}_{\text{SSP,uncom}} + C^{(3)}_{\text{SSP,uncom}}, \quad (22)$$

where the detailed expressions of the additive terms are given in (18-21).

2) Tree-Based Message Stitching: The performance of tree coding is discussed in [9]. However, the complexity is only loosely measured by the number of parity check constraints that must be verified. For comparison, we derive the complexity of the tree-based message stitching of [21] quantified in terms of the number of multiplication and search operations.

In [21], a total T slots are equally divided into N_c groups and then one of the slots is chosen from the n_c-th group to transmit the n_c-th RM sequence. Hence, if the average number of the detected results in each slot is also given by (17), then each non-leaf node has $\hat{N}^{(\text{RM})}T/N_c$ children in the tree decoder. Assuming that the codewords sent in the same slot are distinct, the expected number of erroneous paths that survive stage n_c, i.e., have stitched n_c chunks already, equals [9]

$$L_{n_c} = \sum_{q=2}^{n_c} \left\{ \left(\frac{\hat{N}^{(\text{RM})}T}{N_c} \right)^{n_c - q} \left(\frac{\hat{N}^{(\text{RM})}T}{N_c} - 1 \right) \prod_{\ell=q}^{n_c} 2^{-l_\ell} \right\}, \quad (23)$$

where $n_c = 2, \cdots, N_c$, and l_ℓ is the number of parity check bits appended to the ℓ-th information chunk. Then, each non-leaf node that survives stage n_c engenders $\hat{N}^{(\text{RM})}T/N_c$ children. For each child, l_{n_c+1} parity checks have to be verified, with $(n_c N_b^{(\text{RM})} - \sum_{q=2}^{n_c} l_q)$ multiplication operations required for calculating a single parity check. Then we have to search through all the children at the complexity of $\mathcal{O}(\hat{N}^{(\text{RM})}T/N_c)$ for the ones satisfying the parity check constraints. To conclude, at stage $(n_c + 1)$, each non-leaf node that survives stage n_c would impose the complexity of $\mathcal{O}\big[\frac{\hat{N}^{(\text{RM})}T}{N_c}\left(l_{n_c+1}\left(n_c N_b^{(\text{RM})} - \sum_{q=2}^{n_c} l_q\right) + 1\right)\big]$. Hence, the total computational complexity of the tree-based message stitching can be expressed as

$$C_{\text{tree}} = \frac{\hat{N}^{(\text{RM})}T}{N_c}\left(l_2 N_b^{(\text{RM})} + 1\right) + \sum_{n_c=2}^{N_c-1} \left\{ (L_{n_c} + 1)\frac{\hat{N}^{(\text{RM})}T}{N_c} \left[l_{n_c+1}\left(n_c N_b^{(\text{RM})} - \sum_{q=2}^{n_c} l_q \right) + 1 \right] \right\}. \quad (24)$$

VI. Simulation Results

In this section, we evaluate the performance of the proposed URA scheme ("*RM_Shift*"). The simulation parameters are listed in Table I. Substituting these parameters into (7), we have $N_b^{(\text{RM})} = 35$. Meanwhile, the common pattern set $\mathcal{D}$ contains 256 valid SSPs, and thus $N_b^{(\text{SP})} = 8$.

The URA scheme proposed in [21] ("*RM_Tree*") is used for benchmarking. This scheme also adopts the packetized and slotted transmission framework. The information chunks are carried by RM sequences, while the message stitching is accomplished relying on tree coding only. Given the parity check allocation as $\mathbf{l}' = \left[l'_{n_c}\right]_{n_c=2}^{N_c}$, the SE of "*RM_Tree*" is calculated by [21]

TABLE I: *Simulation Parameters*

RM Sequence Length, 2^m	256
The Number of Chunks, N_c	3
The Number of Slots, T	96
The Number of Complex Channel Uses, $N = 2^m T$	24576
Energy Per Bit, E_b	1
Simulation Times	10000

$$\eta'(\mathbf{l}') = \frac{\left(N_b^{(\text{RM})} + \lfloor \log_2(\frac{T}{N_c}) \rfloor\right) N_c - \sum_{n_c=2}^{N_c} l'_{n_c}}{2^m T} \text{ bit/c.u./user.} \quad (25)$$

A. The Performance of Slot-Based RM Detection

We first compare the performance of the slot-based RM detection in "*RM_Shift*" with that in "*RM_Tree*", which is quantified by the error probability defined as $P_{\text{err}}^{(\text{RM})} \triangleq P_{\text{md}}^{(\text{RM})} + P_{\text{fa}}^{(\text{RM})}$.

Based on the analysis of Section V-A, the performance of the slot-based RM detection is closely related to the distribution of RM sequences across slots. Fig. 6 exhibits the probability distribution of the number of RM sequences in the same slot, i.e., $P(\Theta)$, in the case of $K_a = 100$. The simulation results of "*RM_Shift*" are entirely consistent with the theoretical results calculated from (12) and (14), thus validating our theoretical derivation. Furthermore, Fig. 6 reveals that when SSP collisions occur, more RM sequences would overlap in the same slot, thus leading to a higher risk of detection failures. The distribution of RM sequences in "*RM_Tree*" is also depicted in Fig. 6. Recall that in "*RM_Tree*", the T slots are equally divided into N_c groups before a specific slot is chosen from the n_c-th group to send the n_c-th RM sequence. Assuming that each slot has an equal chance of being chosen, we have

$$P_{\text{t}}(\Theta = \theta) = \binom{K_a}{\theta} \left(\frac{N_c}{T}\right)^{\theta} \left(1 - \frac{N_c}{T}\right)^{K_a - \theta}. \quad (26)$$

Under the simulation parameters of Table I, the value of N_c/T happens to be equal to $m/\tilde{N}_{\text{SP}}$. Thus, the distribution of RM sequences in "*RM_Tree*" should be the same as that in "*RM_Shift, with SSP collisions*" according to (26) and (14), which agrees with the results of Fig. 6.

Fig. 7 depicts the error probability of RM detection versus the number of active users when $E_b/N_0 = 25$dB. We can infer several observations from it as listed below:

1) As expected, the error probability of RM detection increases with the number of active users, which is also in line with our analysis in Section V-A.
2) Since RM sequences are distributed more evenly without SSP collisions, "*RM_Shift*" (shown by the red solid line) shows improved RM detection performance against the increasing user activity.
3) The RM detection capability of "*RM_Shift*" degrades when SSP collisions occur since transmitting more RM

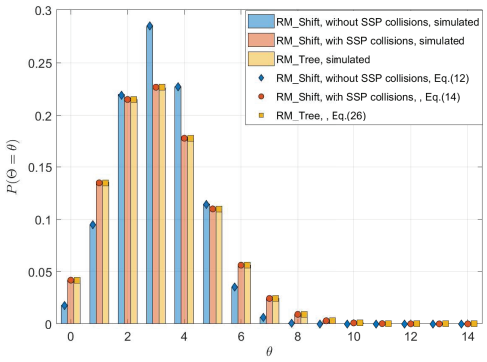


Fig. 6: *The distribution of RM sequences across slots.*

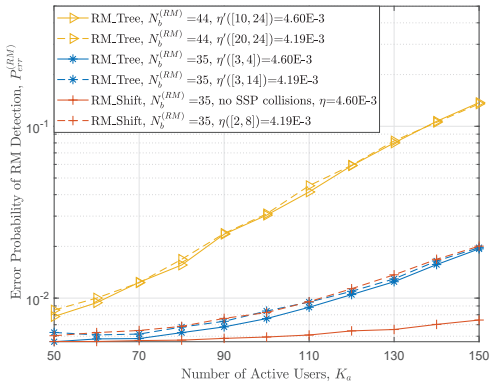


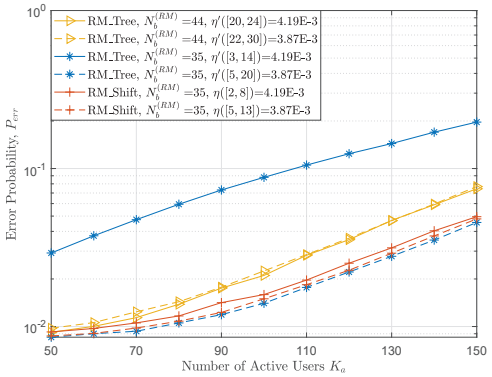
Fig. 7: *The error probability of RM detection versus the number of active users when* $E_b/N_0 = 25dB$.

sequences in the same slot increases MUI and introducing parity check bits reduces the SNR of the received signal.

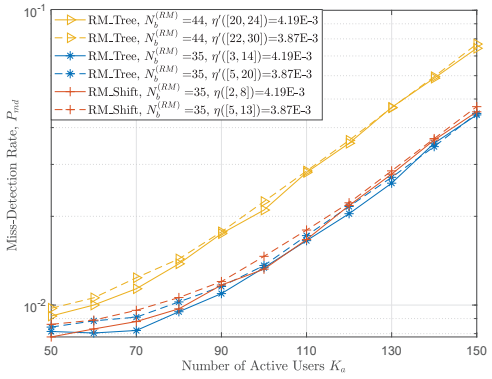
4) Since the constraints in (5-6) are not necessary in "*RM_Tree*", they can be removed and then the value of $N_b^{(\mathrm{RM})}$ increases to $m(m+3)/2 = 44$ [21]. The lines marked by "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$" and "*RM_Tree*, $N_b^{(\mathrm{RM})} = 44$" respectively quantify the RM detection capability with and without these constraints. Their huge disparity reveals that these constraints can significantly reduce detection failures despite the shrinkage of the RM sequence space.

B. The Overall Performance of the URA System

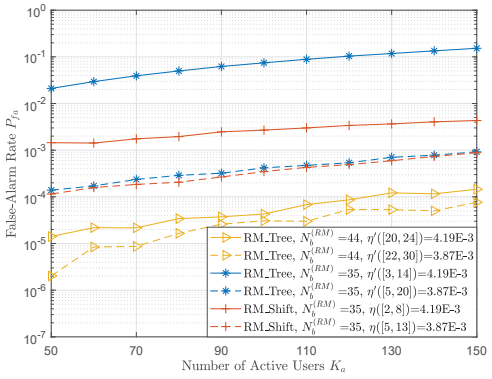
The simulation results of $E_b/N_0 = 25$dB are given in Fig. 8. It shows that in the case of $\eta'([3,14]) = 0.00419$, "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$" does not work well, and the errors mainly belong to the false-alarm category according to Fig. 8(c), which indicates the weak parity checks. Upon increasing the number of parity check bits to 25, the error probability of "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$" decreases dramatically, at the cost of reduced spectral efficiency. Under the same SE, more parity check bits can be used in "*RM_Tree*, $N_b^{(\mathrm{RM})} = 44$" for constructing more powerful parity checks and hence the FAR and the error probability show a significant reduction. However, the performance bottleneck of "*RM_Tree*, $N_b^{(\mathrm{RM})} = 44$" lies in the inferior RM detection capability as shown in Fig. 8(b), and



(a) the error probability



(b) the miss-detection rate



(c) the false-alarm rate

Fig. 8: *The performance of different URA schemes in the case where SSP collisions may occur and* $E_b/N_0 = 25dB$.

thus adding more parity checks plays little role in decreasing the error probability. By contrast, it shows in Fig. 8(a) that when $\eta = 0.00419$, "*RM_Shift*" with merely 10 parity checks achieves much better performance than "*RM_Tree*", and it even outperforms "*RM_Tree*, $N_b^{(\mathrm{RM})} = 44$, $\eta'([22,30])$=3.87E-3". It is because the messages having different SSPs can be readily stitched together following the clues provided by their SSPs in "*RM_Shift*", and we only have to verify the parity checks for identifying those messages suffering from SSP collisions. Besides, despite the similar performance of

"*RM_Shift*, $N_b^{(\mathrm{RM})} = 35$, $\eta'([5,13])$=3.87E-3" and "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$, $\eta'([5,20])$=3.87E-3", much fewer parity bits are needed in "*RM_Shift*", which can significantly ease the computational burden. The simulation results of "*RM_Shift*, $N_b^{(\mathrm{RM})} = 35$, $\eta'([2,8])$=4.19E-3" and "*RM_Shift*, $N_b^{(\mathrm{RM})} = 35$, $\eta'([5,13])$=3.87E-3" in Fig. 8(b) and Fig. 8(c) also validate that increasing the number of parity checks lowers the false-alarm rate by enhancing the performance of message stitching, but raises the miss detection rate because of the degraded RM detection capability.

Fig. 9 shows the performance in the case of no SSP collisions and the message stitching in "*RM_Shift*" relies solely on SSPs. SSP collisions are hard to avoid in practice, and the objective of this simulation setting is to make a clear comparison between the SSP-based and the tree-based message stitching by quantifying the performance of the URA systems adopting them alone. To eliminate the impact of the distribution of RM sequences, we also stipulate that the RM sequences in "*RM_Tree*" follow the same distribution pattern as that in "*RM_Shift*". According to Fig. 7, given the same common RM codebook, the same distribution of RM sequences, and the same SE, the RM detection capabilities of "*RM_Shift*, $N_b^{(\mathrm{RM})} = 35$" and "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$" are almost the same. On this basis, the performance gap between them in Fig. 9 mainly arises from their different message stitching methods, and it validates that the SSP-based message stitching significantly outperforms the tree-based one in terms of its error probability. Furthermore, "*RM_Shift*, $N_b^{(\mathrm{RM})} = 35$" also achieves a lower error probability than "*RM_Tree*, $N_b^{(\mathrm{RM})} = 44$" because of its better RM detection capability, which becomes increasingly obvious upon increasing the number of active users.

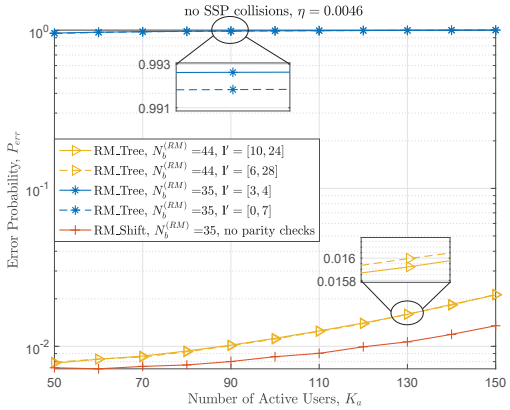


Fig. 9: *The performance of different URA schemes in the case of no SSP collisions and* $E_b/N_0 = 25$*dB.*

The computational complexity of these schemes is calculated according to (22) and (24) and depicted in Fig. 10. The difference in the complexity of "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$" and "*RM_Tree*, $N_b^{(\mathrm{RM})} = 44$" reveals that introducing more parity bits results in more reliable parity checking, but at a much heavier computational burden. Furthermore, it shows that assigning more parity check bits to the last chunk also improves the performance of message stitching at the expense of complexity, which agrees with the analysis in [9]. By

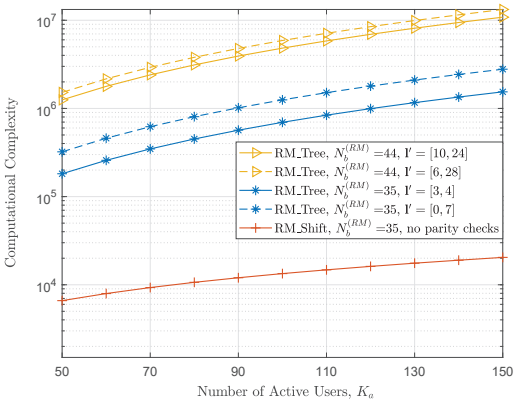


Fig. 10: *The computational complexity versus the number of active users.*

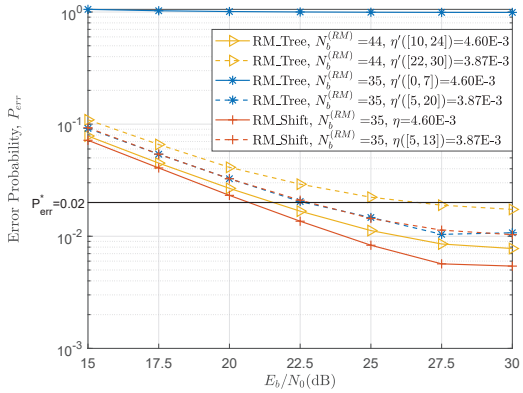


Fig. 11: *The error probability of different URA schemes versus* E_b/N_0.

contrast, the computational complexity of "*RM_Shift*" is orders of magnitudes lower. In conjunction with Fig. 9, it validates the superiority of our proposed scheme in terms of its error probability and computational complexity.

Finally, in Fig. 11, we evaluate the impact of E_b/N_0 on the performance of different URA schemes for $K_a = 100$. The level of E_b/N_0 mainly affects the RM detection capability. However, the performance of "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$, $\eta'([0,7])$=4.60E-3" is limited by the weak parity check constraints, and hence its error probability is not reduced with the increasing E_b/N_0. As the reliability of parity checks improves upon increasing the number of parity check bits, the performance of "*RM_Tree*" exhibits a substantial improvement with growing E_b/N_0 before reaching a plateau at about $E_b/N_0 =$ 27.5dB. As for "*RM_Shift*", increasing E_b/N_0 reduces the error probability, especially when there are no SSP collisions and the RM detection capability dominates the performance of "*RM_Shift*". Given the target error probability P^*_{err}, the required E_b/N_0 of "*RM_Shift*, $N_b^{(\mathrm{RM})} = 35$, $\eta([5,13])$=3.87E-3" is much lower than that of "*RM_Tree*, $N_b^{(\mathrm{RM})} = 44$, $\eta'([22,30])$=3.87E-3", especially when P^*_{err} is relatively small, e.g., $P^*_{\mathrm{err}} = 0.02$ in Fig. 11. Similar to Fig. 8, the performance of "*RM_Shift*, $N_b^{(\mathrm{RM})} = 35$, $\eta([5,13])$=3.87E-3" and "*RM_Tree*, $N_b^{(\mathrm{RM})} = 35$, $\eta'([5,20])$=3.87E-3" is comparable,

but fewer parity checks are needed in "*RM_Shift*", thus validating that combining SSPs with parity checks achieves superior performance at a reduced complexity.

VII. Conclusions

An RM-based URA scheme under the packetized and slotted transmission framework is proposed in this paper. First, we use RM sequences to transmit information chunks in slots since their nested structure facilitates computationally efficient RM detection. Besides, we exploit the shift property of RM sequences and design SSPs for convenient message stitching. The factors influencing the performance of the slot-based RM detection are discussed, and the complexity expressions of the SSP-based message stitching are derived. Our simulation results validate the advantage of our proposed RM-based URA scheme in terms of its error probability and computational complexity.

Appendix A
Proof of Theorem 1

According to (2), it can be derived that if the indices j and j' satisfy $\mathbf{a}_{j'-1}^m = \left[\mathbf{a}_{j-1}^{m-1}; a_{j-1}^m(m)\right]$, then we have $c_{j'}^s = c_j^{s-1}$, where $\mathbf{a}_{j-1}^m = \left[a_{j-1}^m(i)\right]_{i=m}^1 = \left[a_{j-1}^m(m); \mathbf{a}_{j-1}^{m-1}\right]$ and $\mathbf{a}_{j'-1}^m = \left[a_{j'-1}^m(i)\right]_{i=m}^1$ are the m-bit binary expressions of $(j-1)$ and $(j'-1)$, respectively. On this basis, the polynomial representations of the indices j and j' are given by

$$\begin{aligned} j' &= a_{j'-1}^m(m)\cdot 2^{m-1} + \cdots + a_{j'-1}^m(2)\cdot 2 + a_{j'-1}^m(1) + 1, \\ j &= a_{j'-1}^m(1)\cdot 2^{m-1} + a_{j'-1}^m(m)\cdot 2^{m-2} + \cdots + a_{j'-1}^m(2) + 1. \end{aligned}$$

Hence, when $a_{j'-1}^m(1) = 0$, i.e., j' is odd, we have $j'+1 = a_{j'-1}^m(m)\cdot 2^{m-1} + \cdots + (a_{j'-1}^m(2)+1)\cdot 2$, and then $j = (j'+1)/2$. In the other case of $a_{j'-1}^m(1) = 1$, j' can be rewritten as $j' = a_{j'-1}^m(m)\cdot 2^{m-1} + \cdots + (a_{j'-1}^m(2)+1)\cdot 2$, and then we arrive at $j = j'/2 + 2^{m-1}$. When considered together, they yield the conclusion in (4).

Appendix B
Proof of Theorem 2

Let the matrix-vector pairs satisfying the constraint in (5) constitute the set $\mathcal{P}^{[m,0]}$ with $\left|\mathcal{P}^{[m,0]}\right| = 2^{\frac{m(m+1)}{2}}$. Furthermore, if the result of shifting the pair $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\} \in \mathcal{P}^{[m,0]}$ s times fails to meet the requirement in (6), we put the pair $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\}$ in the set $\mathcal{P}^{[m,s]}$, where $s = 1, \cdots, m-1$. According to the inclusion-exclusion principle of [22], the number of matrix-vector pairs that simultaneously satisfy (5-6) is expressed as

$$\left|\mathcal{P}^{[m,0]}\right| + \sum_{n=1}^{m-1}(-1)^n \sum_{\{s_1,\cdots,s_n\}\in\zeta_n} \left|\mathcal{P}^{m/s_1} \cap \cdots \cap \mathcal{P}^{m/s_n}\right|, \quad \text{(B1)}$$

where ζ_n represents the set consisting of all possible combinations of the elements of $\{1, \cdots, m\}$ taken n without repetition. Hence, our main task is to derive the value of $\left|\mathcal{P}^{m/s_1} \cap \cdots \cap \mathcal{P}^{m/s_n}\right|$.

With the matrix-vector pair in $\mathcal{P}^{[m,0]}$ specified as $\mathbf{P}^{[m,0]} = [p_{i,j}]_{i=1,j=1}^{m,m}$ and $\mathbf{b}^{[m,0]} = [b_i]_{i=1}^m$, the constraint in (5) can be rewritten as

$$\forall i = 1, \cdots, m, \quad b_i = p_{i,i} \oplus p_{i,i+1} \oplus \cdots \oplus p_{i,m}. \quad \text{(B2)}$$

If its shifted version $\left\{\mathbf{P}^{[m,s]}, \mathbf{b}^{[m,s]}\right\}$ cannot meet the requirement in (6), based on the correspondence between the elements of $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\}$ and $\left\{\mathbf{P}^{[m,s]}, \mathbf{b}^{[m,s]}\right\}$, we have

$$\begin{cases} b_i = p_{i,i} \oplus \cdots \oplus p_{i,m} \\ \qquad \oplus p_{i,1} \oplus \cdots \oplus p_{i,s}, & \forall\, i = s+1, \cdots, m, \\ b_i = p_{i,i} \oplus \cdots \oplus p_{i,s}, & \forall\, i = 1, \cdots, s. \end{cases} \quad \text{(B3)}$$

According to (B2) and (B3), if $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\}$ belongs to the set $\mathcal{P}^{[m,s]}$, it is satisfied that

$$\begin{cases} p_{i,1} \oplus \cdots \oplus p_{i,s} = 0, & \forall\, i = s+1, \cdots, m \\ p_{i,s+1} \oplus \cdots \oplus p_{i,m} = 0, & \forall\, i = 1, \cdots, s \end{cases}. \quad \text{(B4)}$$

Since $\mathbf{P}^{[m,0]}$ is symmetric, (B4) is equivalent to

$$\begin{cases} p_{1,i} \oplus \cdots \oplus p_{s,i} = 0, & \forall\, i = s+1, \cdots, m \\ p_{i,s+1} \oplus \cdots \oplus p_{i,m} = 0, & \forall\, i = 1, \cdots, s \end{cases}, \quad \text{(B5)}$$

which means that the modulo-2 sums of the elements in each row and each column of the submatrix $\tilde{\mathbf{P}}^s = [p_{i,j}]_{i=1,j=s+1}^{s,m}$ are zeros. The number of such submatrices is in **Lemma B1**.

Lemma B1: The number of $(I \times J)$-element binary matrices satisfying that the modulo-2 sums of the entries in each row and each column are zeros equals $2^{IJ-(I+J-1)}$.

Proof: Given a binary matrix $\mathbf{X} = [x_{i,j}]_{i=1,j=1}^{I,J}$, the requirement that the modulo-2 sums of the elements in each row and each column of it are zeros can be formulated as the following homogeneous linear equations:

$$\begin{cases} \sum_{j=1}^{J} x_{i,j} = 0, & \forall\, i = 1, \cdots, I \\ \sum_{i=1}^{I} x_{i,j} = 0, & \forall\, j = 1, \cdots, J \end{cases}, \quad \text{(B6)}$$

which can be compacted as $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}_{I+J}$, where $\mathbf{A}$ is a $((I+J) \times IJ)$-element binary matrix, and $\mathbf{x} \in \mathbb{Z}_2^{IJ\times 1}$ is obtained by vectorizing $\mathbf{X}$. Since the rank of $\mathbf{A}$ is $(I+J-1)$, the fundamental set of the solutions of (B6) contains $IJ-(I+J-1)$ solution vectors. Bearing in mind that $x_{i,j}$ are binary, the number of the solutions of (B6) equals $2^{IJ-(I+J-1)}$, thus proving **Lemma B1**. ■

On this basis, we provide the following lemma concerning the value of $\left|\mathcal{P}^{m/s_1} \cap \cdots \cap \mathcal{P}^{m/s_n}\right|$.

Lemma B2: The number of matrix-vector pairs $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\}$ that simultaneously belong to the n sets $\mathcal{P}^{s_1}, \cdots, \mathcal{P}^{s_n}$ is given by $\left|\mathcal{P}^{m/s_1} \cap \cdots \cap \mathcal{P}^{m/s_n}\right| = 2^{\frac{m(m+1)-n(2m-n-1)}{2}}$.

Proof: We first look at the case of $n = 1$ and omit the subscript of s for ease of expression. As mentioned above, the pairs in $\mathcal{P}^s$ meet the requirement in (B5), and there are $2^{s(m-s)-(s+m-s-1)}$ such submatrices $\tilde{\mathbf{P}}^s$ according to **Lemma B1**. Meanwhile, the remaining $\frac{m(m+1)}{2} - s(m-s)$ elements of $\mathbf{P}^{[m,0]}$ that are not included in $\tilde{\mathbf{P}}^s$ can take the value of 1 or 0 arbitrarily, and thus we arrive at $|\mathcal{P}^s| = 2^{\frac{m(m+1)}{2}-s(m-s)} \times 2^{s(m-s)-(s+m-s-1)} = 2^{\frac{m(m+1)}{2}-(m-1)}$, which is in accordance with the conclusion of **Lemma B2**.

Next, we consider the case of $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\} \in \mathcal{P}^{m/s_1} \cap \mathcal{P}^{m/s_2}$, and we assume that $s_1 < s_2$ without loss of generality. The submatrices $\tilde{\mathbf{P}}^{s_1} = [p_{i,j}]_{i=1,j=s_1+1}^{s_1,m}$ and

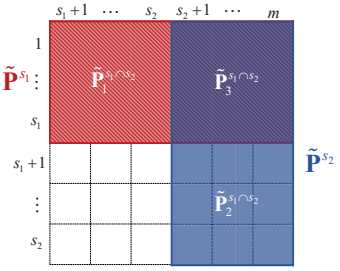


Fig. B1: *The illustration of the submatrices* $\tilde{\mathbf{P}}^{s_1}$ *and* $\tilde{\mathbf{P}}^{s_2}$.

$\tilde{\mathbf{P}}^{s_2} = [p_{i,j}]_{i=1,j=s_2+1}^{s_2,m}$ are depicted in Fig. B1, and they satisfy that

$$\begin{cases} p_{i,s_1+1} \oplus \cdots \oplus p_{i,m} = 0, & \forall i = 1, \cdots, s_1 \\ p_{1,j} \oplus \cdots \oplus p_{s_1,j} = 0, & \forall j = s_1+1, \cdots, m \end{cases}, \quad \begin{cases} p_{i,s_2+1} \oplus \cdots \oplus p_{i,m} = 0, & \forall i = 1, \cdots, s_2 \\ p_{1,j} \oplus \cdots \oplus p_{s_2,j} = 0, & \forall j = s_2+1, \cdots, m \end{cases}. \tag{B7}$$

Then, we can divide the elements in $\tilde{\mathbf{P}}^{s_1}$ and $\tilde{\mathbf{P}}^{s_2}$ into three matrices as shown in Fig. B1, i.e., $\tilde{\mathbf{P}}_1^{s_1\cap s_2} = [p_{i,j}]_{i=1,j=s_1+1}^{s_1,s_2}$, $\tilde{\mathbf{P}}_2^{s_1\cap s_2} = [p_{i,j}]_{i=s_1+1,j=s_2+1}^{s_2,m}$, and $\tilde{\mathbf{P}}_3^{s_1\cap s_2} = [p_{i,j}]_{i=1,j=s_2+1}^{s_1,m}$, and (B7) is equivalent to that these three new matrices all satisfy the requirement that the modulo-2 sums of the elements in their each row and each column are all zeros. Denote the numbers of rows and columns of $\tilde{\mathbf{P}}_\tau^{s_1\cap s_2}$ as I_τ and J_τ, respectively, where $\tau = 1, 2, 3$. Then, the number of different $\tilde{\mathbf{P}}_\tau^{s_1\cap s_2}$ equals $2^{I_\tau J_\tau - (I_\tau + J_\tau - 1)}$ according to **Lemma B1**. Taking the remaining unconstrained elements in $\mathbf{P}^{[m,0]}$ into consideration, we have

$$\begin{aligned} \left|\mathcal{P}^{m/s_1} \cap \mathcal{P}^{m/s_2}\right| &= 2^{\frac{m(m+1)}{2} - \sum_{\tau=1}^{3} I_\tau J_\tau} \cdot \prod_{\tau=1}^{3} 2^{I_\tau J_\tau - (I_\tau + J_\tau - 1)} \\ &= 2^{\frac{m(m+1)}{2} - \sum_{\tau=1}^{3} (I_\tau + J_\tau - 1)} \\ &= 2^{\frac{m(m+1)}{2} - (2m-3)} \end{aligned}.$$

When $n > 2$, the derivation process is similar. To summarize, if $\left\{\mathbf{P}^{[m,0]}, \mathbf{b}^{[m,0]}\right\}$ is contained in the sets $\mathcal{P}^{m/s_1}, \cdots, \mathcal{P}^{m/s_n}$ simultaneously, then the modulo-2 sums of the elements in each row and each column of the matrices $[p_{i,j}]_{i=1+s_{\tau'},j=1+s_{\tau''}}^{s_{\tau'+1},s_{\tau''+1}}$ are zeros, where $\tau' = 0, \cdots, n-1$, $\tau'' = \tau' + 1, \cdots, n$, $s_0 = 0$ and $s_{n+1} = m$. On this basis, we arrive at the conclusion that

$$\begin{aligned} &\left|\mathcal{P}^{m/s_1} \cap \cdots \cap \mathcal{P}^{m/s_n}\right| \\ =&2^{\frac{m(m+1)}{2} - \sum_{\tau'=0}^{n-1} \sum_{\tau''=\tau'+1}^{n} \left[(s_{\tau'+1} - s_{\tau'}) + (s_{\tau''+1} - s_{\tau''}) - 1\right]} \\ =&2^{\frac{m(m+1)}{2} - \frac{n(2m-n-1)}{2}}, \end{aligned}$$

which completes the proof. ■

Finally, substituting the conclusion in **Lemma B2** into (B1) yields **Theorem 2**.

APPENDIX C
THE RM DETECTION ALGORITHM USING THE NESTED STRUCTURE OF RM SEQUENCES

Given the recursive representation of the matrix-vector pairs in (3), and denoting the RM sequences generated by the pairs $\left\{\mathbf{P}^m, \mathbf{b}^m\right\}$ and $\left\{\mathbf{P}^{m-1}, \mathbf{b}^{m-1}\right\}$ respectively by $\mathbf{c}^m$ and $\mathbf{c}^{m-1}$, the *nested structure* of RM sequences is expressed as $\mathbf{c}^m = \left[\mathbf{c}^{m-1}; \mathbf{c}^{m-1} \odot \mathbf{v}^{m-1}\right]$, where $\mathbf{v}^{m-1}$ is the length-2^{m-1} Walsh sequence parameterized by $\left\{\boldsymbol{\alpha}^{m-1}, \rho_m, b_m\right\}$, with its j-th entry being

$$v_j^{m-1} = \iota^{(\rho_m + 2b_m)} \cdot (-1)^{(\boldsymbol{\alpha}^{m-1})^T \mathbf{a}_{j-1}^{m-1}}. \tag{C1}$$

Exploiting the above nested structure, the layer-by-layer RM detection algorithm summarized in **Alg. 4** is proposed for detecting a single RM sequence. If multiple sequences are superimposed in the received signal, they are detected iteratively and the details can be found in [19].

Algorithm 4 : Layer-by-Layer RM Detection Algorithm [19]

Input: the received signal $y_j = h \cdot c_j^m + e_j^m$, $1 \le j \le 2^m$.

1: Initialize $\mathbf{y}^m = \mathbf{y}$.
2: **for** $s = m : -1 : 2$ **do**
3: Perform the Walsh-Hadamard Transformation (**WHT**) on the result of $(\mathbf{y}^s)' \odot \left((\mathbf{y}^s)''\right)^*$ to obtain the estimates $\{\hat{\boldsymbol{\alpha}}^{s-1}, \hat{\rho}_s, \hat{b}_s\}$, where $(\mathbf{y}^s)' \triangleq [y_j^s]_{j=1}^{2^{s-1}}$ and $(\mathbf{y}^s)'' \triangleq [y_j^s]_{j=2^{s-1}+1}^{2^s}$.
4: Recover $\hat{\mathbf{v}}^{s-1}$ according to (C1) and calculate that $y_j^{s-1} = \frac{1}{2}\left[\left(y_j^s\right)' + \left(\hat{v}_j^{s-1}\right)^* \cdot \left(y_j^s\right)''\right]$.
5: **end for**
6: Compute that $\mathbf{V} = \left(\mathbf{y}^1\right)^* \cdot \left[\begin{smallmatrix} 1 & 1 & 1 & 1 \\ 1 & \iota & -1 & -\iota \end{smallmatrix}\right]$, and then we have $[\hat{b}_1, \hat{\rho}_1] = \mathbf{a}_{w-1}^2$ with $w \triangleq \arg\max_i \{V_1, V_2, V_3, V_4\}$.
7: Reconstruct $\{\hat{\mathbf{P}}^m, \hat{\mathbf{b}}^m\}$ with $\{\hat{\boldsymbol{\alpha}}^{s-1}, \hat{\rho}_s, \hat{b}_s\}, s = 1, \ldots, m$, and calculate the channel estimate following $\hat{h} = \frac{1}{2}(y_1^1 + (-\iota)^{2\hat{b}_1 + \hat{\rho}_1} \cdot y_2^1)$.
8: **return** $\{\hat{\mathbf{P}}^m, \hat{\mathbf{b}}^m\}, \hat{h}$.

REFERENCES

[1] C. Bockelmann *et al.*, "Massive machine-type communications in 5G: physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59-65, Sept. 2016.

[2] Z. Chen, F. Sohrabi and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1890-1904, Apr., 2018.

[3] G. Durisi, T. Koch and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711-1726, Sept. 2016.

[4] L. Liu and W. Yu, "Massive connectivity with massive MIMO-Part I: Device activity detection and channel estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2933-2946, Jun. 2018.

[5] S. Kim, H. Kim, H. Noh, Y. Kim and D. Hong, "Novel transceiver architecture for an asynchronous grant-free IDMA system," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4491-4504, Sept. 2019.

[6] H. Xu, W. Yu, D. Griffith and N. Golmie, "A survey on industrial internet of things: a cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78238-78259, 2018.

[7] Y. Polyanskiy, "A perspective on massive random-access," in *IEEE Int. Symp. Inform. Theory (ISIT)*, 2017, pp. 2523-2527.

[8] K. -H. Ngo, A. Lancho, G. Durisi and A. G. i. Amat, "Massive uncoordinated access with random user activity," in *IEEE Int. Symp. Inform. Theory (ISIT)*, 2021, pp. 3014-3019.

[9] V. K. Amalladinne, J. F. Chamberland and K. R. Narayanan, "A coded compressed sensing scheme for uncoordinated multiple access," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6509-6533, Oct. 2020.

[10] A. Fengler, P. Jung and G. Caire, "Sparcs for unsourced random access," *IEEE Trans. Inf. Theory*, vol. 67, no. 10, pp. 6894-6915, Oct. 2021.
[11] V. K. Amalladinne, A. K. Pradhan, C. Rush, J.-F. Chamberland and K.R. Narayanan, "Unsourced random access with coded compressed sensing: Integrating AMP and belief propagation," *arXiv preprint*, arXiv:2010.04364, 2020.
[12] V. Shyianov, F. Bellili, A. Mezghani and E. Hossain, "Massive unsourced random access based on uncoupled compressive sensing: another blessing of massive MIMO," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 820-834, March 2021.
[13] A. Decurninge, I. Land, and M. Guillaud, "Tensor-based modulation for unsourced massive random access," *IEEE Wireless Commun. Lett.*, 2020.
[14] S. D. Howard, A. R. Calderbank, and S. J. Searle, "A fast reconstruction algorithm for deterministic compressive sensing using second order Reed-Muller codes," in *42nd Annu. Conf. Inf. Sci. and Sys.*, Princeton, 2008, pp. 11-15.
[15] R. Calderbank and S. Jafarpour, "Reed Muller sensing matrices and the LASSO," in *Int. Conf. Sequences Their Applicat. (SETA)*, Paris, 2010, pp. 442-463.
[16] R. Calderbank, S. Howard and S. Jafarpour, "Sparse reconstruction via the Reed-Muller Sieve," in *IEEE Int. Sym. Inf. Theory (ISIT)*, Austin, 2010, pp. 1973-1977.
[17] L. Zhang, J. Luo and D. Guo, "Neighbor discovery for wireless networks via compressed sensing," *Perform. Eval.*, vol. 70, no. 7, pp. 457-471, 2013.
[18] H. Zhang, R. Li, J. Wang, Y. Chen and Z. Zhang, "Reed-Muller sequences for 5G grant-free massive access," in *IEEE Global Commun. Conf.*, Singapore, 2017, pp. 1-7.
[19] J. Wang, Z. Zhang and L. Hanzo, "Joint active user detection and channel estimation in massive access systems exploiting Reed-Muller sequences," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 739-752, Jun. 2019.
[20] J. Wang, Z. Zhang, C. Zhong and L. Hanzo, "Incremental massive random access exploiting the nested Reed-Muller sequences," *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 2917-2932, May 2021.
[21] R. Calderbank and A. Thompson, "CHIRRUP: a practical algorithm for unsourced multiple access," *Inform. Inference J. IMA*, vol. 9, no. 4, pp. 875–897, 2020.
[22] A. Björklund, T. Husfeldt, and M. Koivisto, "Set partitioning via inclusion-exclusion," *SIAM J. Computing*, vol. 39, no. 2, pp. 546-563, Jul. 2009.
[23] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. American Stat. Assoc.*, vol. 58, no. 301, pp. 13-30, Mar. 1963.

Jue Wang (S'18) received the B.S. degree in communication engineering from the Department of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016. She received the Ph.D. degree in information and communication engineering under the supervision of Prof. Z. Zhang at Zhejiang University in 2021. Her research interests include signal processing, massive random access, massive MIMO and machine learning.

Zhaoyang Zhang (M'02-SM'21) received his Ph.D. degree from Zhejiang University, Hangzhou, China, in 1998, where he is currently a Qiushi Distinguished Professor. His research interests are mainly focused on the fundamental aspects of wireless communications and networking, such as information theory and coding, network signal processing and distributed learning, AI-empowered communications and networking, network intelligence with synergetic sensing, computing and communication, etc. He has co-authored more than 300 peer-reviewed international journal and conference papers, and is a co-recipient of 8 international conference best papers awarded by IEEE ICC 2019 and IEEE GlobeCom 2020, etc. He was granted the National Natural Science Fund for Distinguished Young Scholars by NSFC in 2017.

Dr. Zhang is serving or has served as Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS and IET COMMUNICATIONS, etc, and as General Chair, TPC Co-Chair or Symposium Co-Chair for PIMRC 2021 Workshop Native-AI, VTC-Spring 2017 Workshop HMWC, WCSP 2013/2018, Globecom 2014 Wireless Communications Symposium, etc. He was also a keynote speaker for Globecom 2021 Workshop on Native-AI Wireless Networks, APCC 2018 and VTC-Fall 2017 Workshop NOMA, etc.

Xiaoming Chen (IEEE Senior Member) received the B.Sc. degree from Hohai University in 2005, the M.Sc. degree from Nanjing University of Science and Technology in 2007 and the Ph. D. degree from Zhejiang University in 2011, all in electronic engineering. He is currently a Professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. From March 2011 to October 2016, He was with Nanjing University of Aeronautics and Astronautics, Nanjing, China. From February 2015 to June 2016, he was a Humboldt Research Fellow at the Institute for Digital Communications, Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Germany. His research interests mainly focus on 5G/6G key techniques, Internet of Things, and smart communications.

Dr. Chen is currently serving as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS. He served as the IEEE COMMUNICATIONS LETTERS, and a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS "Massive Access for 5G and Beyond" and the IEEE WIRELESS COMMUNICATIONS "Massive Machine-Type Communications for IoT". He received the Best Paper Awards at the IEEE Global Communications Conference (GLOBECOM) 2020, the International Conference on Wireless Communications and Signal Processing (WCSP) 2020, the IEEE International Conference on Communications (ICC) 2019, and the IEEE/CIC International Conference on Communications in China (ICCC) 2018.

Caijun Zhong (S'07-M'10-SM'14) received the B.S. degree in Information Engineering from the Xi'an Jiaotong University, Xi'an, China, in 2004, and the M.S. degree in Information Security in 2006, Ph.D. degree in Telecommunications in 2010, both from University College London, London, United Kingdom. From September 2009 to September 2011, he was a research fellow at the Institute for Electronics, Communications and Information Technologies (ECIT), Queen's University Belfast, Belfast, UK. Since September 2011, he has been with Zhejiang University, Hangzhou, China, where he is currently a Professor. His current research interests include Reconfigurable intelligent surfaces assisted communications and artificial intelligence based wireless communications.

Dr. Zhong is an Editor of Science China: Information Science and China Communications. He was an editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS LETTERS. He is the recipient of the 2013 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He and his coauthors has been awarded a Best Paper Award at the IEEE GLOBECOM 2020 and IEEE ICC 2019.

Lajos Hanzo (http://www-mobile.ecs.soton.ac.uk, https://en.wikipedia.org/wiki/Lajos_Hanzo) (FIEEE'04) received his Master degree and Doctorate in 1976 and 1983, respectively from the Technical University (TU) of Budapest. He was also awarded the Doctor of Sciences (DSc) degree by the University of Southampton (2004) and Honorary Doctorates by the TU of Budapest (2009) and by the University of Edinburgh (2015). He is a Foreign Member of the Hungarian Academy of Sciences and a former Editor-in-Chief of the IEEE Press. He has served several terms as Governor of both IEEE ComSoc and of VTS. He has published 2000+ contributions at IEEE Xplore, 19 Wiley-IEEE Press books and has helped the fast-track career of 123 PhD students. Over 40 of them are Professors at various stages of their careers in academia and many of them are leading scientists in the wireless industry. He is also a Fellow of the Royal Academy of Engineering (FREng), of the IET and of EURASIP. He is the recipient of the 2022 Eric Sumner Field Award.