Coded Acknowledgement with Random Subspaces

Nicholas Kwan*, Ryan Song†, and Wei Yu*
*Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Canada
†School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Emails: nick.kwan@mail.utoronto.ca, ryan.song@epfl.ch, weiyu@ece.utoronto.ca

***Abstract*—This paper studies efficient acknowledgement strategies for massive random access, where a random subset of ℓ users from a large pool of n potential users seek connectivity from the base-station (BS); the BS detects $k \leq \ell$ of these users and wishes to communicate a positive acknowledgement message ACK to the k detected users and a negative acknowledgement NACK to the $\ell - k$ remaining users whose identities are not known to the BS. Assuming that the BS communicates to the ℓ users by broadcasting a common message, what is the minimum length of the common message that can communicate the ACK and NACK messages up to some allowable per-user error probabilities λ_1 and λ_2 for the k detected and $\ell - k$ undetected users, respectively? Assuming that the BS and users have shared common randomness, we use bounds on Gaussian coefficients to show that a common message length of $\lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2$ bits is sufficient, independent of n. For large n, the achievability is tight to within $O(k)$. We propose a low-complexity coding strategy based on communicating the subspace spanned by the random binary codewords assigned to the users as the acknowledgement message, and show that such a scheme is order optimal in term of the common message length, and has a complexity equivalent to row reducing a $k \times (k + \lceil \log(1/\lambda_2) \rceil)$ binary matrix.**

I. Introduction

Consider a massive access communication scenario in which a central base-station (BS) wishes to provide connectivity to a large pool of n potential users. Due to the sparsity and randomness of the user activities, at any given time, only a random subset $\mathcal{L} \subset [n]$ of $|\mathcal{L}| = \ell$ users are requesting connectivity, of which only a subset $\mathcal{A} \subset \mathcal{L}$ of $|\mathcal{A}| = k$ users have been detected by the BS. The remaining users $\mathcal{L} \setminus \mathcal{A}$ have not been detected, and are thus unknown to the BS. Assuming fixed values of n, ℓ, and k, we ask the question of how to efficiently communicate a positive acknowledgement message ACK to all the *active* users in $\mathcal{A}$ and a negative acknowledgement message NACK to all the remaining users in $\mathcal{L} \setminus \mathcal{A}$, so that all users in $\mathcal{L}$ know whether service is available to them. More specifically, suppose that the BS performs acknowledgement via a common message broadcasted over a noiseless downlink channel. What is the minimum common message length required to communicate the ACK and the NACK messages?

One immediate strategy for performing such a task is to assign a priori a unique $\log(n)$-bit identifier for each of the n potential users, then to transmit a list of the identities of the active users in $\mathcal{A}$. This naive strategy communicates the acknowledgement messages without error, and requires a common message length of $k \log(n)$ bits. This common message length can be improved to $\log \binom{n}{k}$ bits by recognizing that the order of the users does not matter and therefore one can employ enumerative source coding [**?**], [**?**] or similar methods [**?**], [**?**] to encode k-subsets of n. However, $\log \binom{n}{k}$ still scales as $O(\log(n))$. For instance, for $n = 10^6$ and $k = 10^2$, the aforementioned encoding requires a common message length of at least 1300 bits. Since each active user receives only one bit of information, the overhead of this strategy is high. Furthermore, these schemes also require that all users are assigned a unique index in $[n]$, which incurs an additional management cost for the BS, especially if the set of users in the network changes over time.

The common message length described above is the minimum possible, if no acknowledgement error is allowed. Recently, Kalør et al. [**?**] showed that it is possible to significantly reduce the common message length by allowing some nonzero error probability in acknowledgement. Although the control channel in most communications systems is often heavily protected, errors can still occur, so it is important to take the probability of erroneous acknowledgement into consideration when designing a system. For massive random access applications, the consequences of receiving an ACK or a NACK erroneously are often asymmetric. Thus, we distinguish between the probability of missed detection λ_1 and the probability of false alarm λ_2. The former occurs when a user who should receive an ACK instead receives a NACK, while the latter occurs when a user who should receive a NACK instead receives an ACK. The main result of [**?**] is that for the case of $\lambda_1 = \lambda_2 = \lambda$, the common message length can be reduced to $k\lceil \log(1/\lambda) \rceil$ bits in theory. In the aforementioned example with $n = 10^6$ and $k = 10^2$, if the number of contending users $\ell = 200$ and if we desire a constant number of errors ϵ in expectation, then it is reasonable to set $\lambda = \epsilon/100$. In this case, for $\epsilon = 1$, the common message length can be reduced to 665 bits, which is a significant saving as compared to 1300 bits in the error-free case. To achieve this, [**?**] proposes a practical coding scheme based on solving a system of k random linear equations over a field $\mathbb{F}_q$ of size $q \geq 1/\lambda_2$. However, the complexity of this scheme can be high, because q can be large.

This paper defines the problem of acknowledgement more formally by assuming the availability of common randomness and by distinguishing the per-user probabilities of error (λ_1, λ_2). We propose a novel coding strategy of assigning each user with a uniformly random binary vector and losslessly compressing the subspace spanned by the vectors assigned to the active users. We show that the proposed scheme achieves a

common message length of $\lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2$ bits. This achievability bound is tight for large n up to an $O(k)$ term which is independent of the error probabilities. This is similar to the theoretical result in [**?**], but does not require $\lambda_1 = \lambda_2$ as in the scheme in [**?**].

Moreover, this paper proposes a low-complexity coding scheme that uses reduced row echelon form (RREF) of binary matrices to represent subspaces of $\mathbb{F}_2$. The proposed scheme achieves a minimum common message length of $k\lceil \log(1/\lambda_2) \rceil + k + \lceil \log(1/\lambda_2) \rceil$ bits in practice for the case of $\lambda_1 = 0$, which is close to the theoretical bound. The common message length achieved by the proposed scheme is only slightly larger than the scheme in [**?**], but the proposed scheme is much less complex, because it operates in finite-field size of 2 instead of $q \geq 1/\lambda_2$. Specifically, the encoding complexity of the proposed scheme is equal to the complexity of row reducing a $k \times (k + \lceil \log(1/\lambda_2) \rceil)$ binary matrix. This represents a significant saving in complexity as compared to the scheme presented in [**?**] by a factor $\log(1/\lambda_2)$.

The problem of coded acknowledgement is closely related to that of identification via channels, studied by Ahlswede and Dueck [**?**]. In identification, the receiver has a fixed message in a codebook, and aims to determine whether or not its message is the same as the sender's message by observing the output of a noisy channel. Ahlswede and Dueck showed that, for vanishing error probabilities, the number of messages that can be identified (i.e., the codebook size) can grow *double-exponentially* with blocklength. Despite their promise, the encoding complexity of explicit constructions of identification codes has proven far too high to be useful in practice [**?**].

The identification problem can be seen as a special case of the acknowledgement problem with $k = 1$, but is also more restrictive in its setup in that no common randomness is allowed in the setting of [**?**]. In practice, common randomness can be implemented using a pseudorandom number generator and a common seed. If common randomness is allowed, then the identification problem actually becomes easy in the sense that it would be possible to construct simple schemes for single-user identification with vanishing error such that the codebook size is an arbitrary function of the blocklength!

One such construction in the context of the acknowledgement problem over a noiseless channel is as follows. For any fixed $\lambda_2 > 0$, assign each user u a uniformly random number $v(u) \in [\lceil 1/\lambda_2 \rceil]$. Then to identify user i, we simply transmit $v(i)$. This requires $\lceil \log(1/\lambda_2) \rceil$ bits. By symmetry, the probability that any user (other than the intended one) is falsely identified is then bounded above by λ_2. To achieve asymptotically vanishing error, we can simply take λ_2 to be any function of n that vanishes as n grows large. Taking $\lambda_2 = 1/\log(n)$ recovers the doubly-exponential scaling of [**?**], but we can beat this scaling by taking λ_2 to be the reciprocal of any function that grows slower than $\log(n)$. The role of common randomness is to average the error probability across all the users to ensure a per-user probability of error bound.

The scheme proposed in this paper can be thought of as a generalization of the simple scheme described above, since transmitting $d = \lceil \log(1/\lambda_2) \rceil$ bits can be thought of as transmitting a one-dimensional subspace in $\mathbb{F}_2^d$.

II. Problem Formulation

We consider the following coded acknowledgement problem consisting of a central BS and n users, where each user is assigned a unique index $u \in [n]$. Given a set of l contending users, k of which should be acknowledged, the BS wishes to communicate an ACK message to all the active users in $\mathcal{A}$, while the remaining $l-k$ users (whose identities are not known to the BS) should receive a NACK message.

Given a set $\mathcal{A}$, the BS uses the encoder

$$f : \binom{[n]}{k} \times \mathcal{Z} \to \{0,1\}^R, \tag{1}$$

to map the set $\mathcal{A} \in \binom{[n]}{k}$ as well as the shared common randomness $z \in \mathcal{Z}$ to a R-bit common message to be broadcast to the users. We assume that the common randomness z is independent of $\mathcal{A}$.

The BS then broadcasts the encoder output $f(\mathcal{A}, z)$ as a common message to all the users, each of which then uses its own decoder

$$g_u : \{0,1\}^R \times \mathcal{Z} \to \{0,1\}, \tag{2}$$

alongside the common randomness to decide whether or not it has been acknowledged, i.e., whether or not it is in $\mathcal{A}$. A zero-error scheme would be equivalent to ensuring that for all $\mathcal{A} \in \binom{[n]}{k}$ and $u \in [n]$,

$$g_u(f(\mathcal{A}, z), z) = 1 \iff u \in \mathcal{A}. \tag{3}$$

It is easy to show that to satisfy the above condition, we would require $R = \Theta(k \log(n/k))$ bits, which corresponds to the number of bits required to encode a uniformly random k subset of $[n]$. To reduce the common message length, this paper focuses on the setting where some errors can be tolerated (which as discussed in [**?**] is also the more practical case).

We say that a coded acknowledgement scheme consisting of an encoder f and decoders $g_1, \cdots, g_n$ achieves missed detection and false alarm error probabilities $0 \leq \lambda_1, \lambda_2 \leq 1$ respectively if, for all $\mathcal{A} \in \binom{[n]}{k}$ and any $u \in [n]$,

$$\Pr\{g_u(f(\mathcal{A}, Z), Z) = 1\} \geq 1 - \lambda_1, \text{ if } u \in \mathcal{A}, \tag{4}$$

and

$$\Pr\{g_u(f(\mathcal{A}, Z), Z) = 0\} \geq 1 - \lambda_2, \text{ if } u \notin \mathcal{A}. \tag{5}$$

Note that these guarantees on the probabilities of error must hold on a per-user basis for all users simultaneously.

We use $(f^*, g_1^*, \cdots, g_n^*, Z)$ to denote the optimal coded acknowledgement scheme which achieves error probabilities λ_1, λ_2 and minimizes the common message length R. We use $R^*(n, k, \lambda_1, \lambda_2)$ to denote the common message length of such an optimal scheme, and we seek to bound $R^*(n, k, \lambda_1, \lambda_2)$ from above (achievability) and from below (converse). For the converse, we assume the worst case where $\mathcal{A}$ is distributed uniformly over all k-subsets $\binom{[n]}{k}$. The achievability results, however, make no assumption on the distribution of $\mathcal{A}$.

III. Achievable Common Message Length

In this section, we propose the following coding strategy for the acknowledgement problem. The basic idea is to assign a binary vector to each user in the network, then to communicate the subspace spanned by the vectors assigned to the users we wish to acknowledge as the common message. Concretely, let $h : [n] \to \mathbb{F}_2^d$ be a uniformly random hash function generated at the encoder that maps the index of each user to a binary vector of length d. A new hash function is generated randomly in each round of acknowledgements. Due to the availability of common randomness, each user $u \in [n]$ has knowledge of its corresponding $h(u)$ in each round. Let $\mathcal{A} \in \binom{[n]}{k}$ be any set of active users, and $\mathcal{A}'$ be a subset of $\mathcal{A}$ containing the users we wish to acknowledge. The encoder communicates the subspace spanned by $h(\mathcal{A}')$ as the common message. At the decoders, each user decides whether they are acknowledged or not by checking if their assigned binary vector is in the subspace.

We choose the binary vector length d to be

$$d = \lceil (1-\lambda_1)k \rceil + \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil. \tag{6}$$

The missed detection error probability of at most λ_1 requires us to acknowledge at least $(1-\lambda_1)k$ active users. The first term in (**??**) is related to the fact that the span of the vectors assigned to $\lceil (1-\lambda_1)k \rceil$ users has dimension at most $\lceil (1-\lambda_1)k \rceil$. The second term in (**??**) serves to bound the false alarm probability of error. In particular, d is the smallest integer such that the ratio of the cardinalities between a $\lceil (1-\lambda_1)k \rceil$-dimensional subspace and $\mathbb{F}_2^d$ is at most λ_2. By utilizing bounds on the number of subspaces of fixed dimension in a vector space of larger dimension over a finite field, we arrive at the desired achievable common message length. We make the rationales for choosing d and the computation of common message length precise in the proof of the theorem below.

Theorem 1: Consider a massive random access scenario in which a central BS and users have access to unlimited common randomness and the BS wishes to acknowledge k users out of a total of n users with per-user missed detection and false alarm error probabilities at most $\lambda_1 \geq 0$ and $\lambda_2 > 0$, respectively. The minimum fixed-length common message $R^*(n, k, \lambda_1, \lambda_2)$ is bounded from above as

$$R^*(n, k, \lambda_1, \lambda_2) \leq \lceil (1-\lambda_1)k \rceil \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil + 2. \tag{7}$$

Proof: As mentioned in the paragraphs leading to the theorem statement, the coding strategy is to use the subspace spanned by the binary vectors assigned to the $\lceil (1-\lambda_1)k \rceil$ users that we wish to acknowledge as the common message. The first step is to subsample the set of active users uniformly at random to obtain a set $\mathcal{A}' \sim \mathrm{Unif}\binom{\mathcal{A}}{\lceil (1-\lambda_1)k \rceil}$ of size $\lceil (1-\lambda_1)k \rceil$. If the binary vectors assigned to the users in $\mathcal{A}'$ are linearly independent, then $\mathrm{span}(h(\mathcal{A}'))$ has dimension $\lceil (1-\lambda_1)k \rceil$, and set $W = \mathrm{span}(h(\mathcal{A}'))$. If some of these binary vectors are linearly dependent, so $\mathrm{span}(h(\mathcal{A}'))$ is of lower dimension than $\lceil (1-\lambda_1)k \rceil$, we construct a random subspace W of dimension $\lceil (1-\lambda_1)k \rceil$ that contains $\mathrm{span}(h(\mathcal{A}'))$ by adding random linearly independent vectors to the span (for reasons that will be explained in Remark **??** after the proof). In both cases, we perform lossless compression of the subspace W and transmit the compressed bits as the acknowledgement message.

By construction, the missed detection error probability is bounded above by λ_1. Now for any user $u \notin \mathcal{A}$, $h(u)$ maps u uniformly at random to the space of binary vectors of length d independently of W, which by symmetry is a random $\lceil (1-\lambda_1)k \rceil$-dimensional subspace of $\mathbb{F}_2^d$. Hence, we have

$$\Pr\{u \in W\} = \frac{|W|}{\left|\mathbb{F}_2^d\right|} = \frac{2^{\lceil (1-\lambda_1)k \rceil}}{2^{\lceil (1-\lambda_1)k \rceil + \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil}} \leq \lambda_2, \tag{8}$$

where we have used (**??**). Thus, the per-user false alarm probability of error is bounded above by λ_2.

It remains to compute the achievable common message length. To this end, we use known upper bounds on the number of subspaces of fixed dimension in a vector space over a finite field. Let $\binom{m}{r}_q$ denote the number of r-dimensional subspaces of $\mathbb{F}_q^m$ (commonly known as the *Gaussian coefficient*, or the q-binomial coefficient). The following bounds hold [**?**]:

$$q^{r(m-r)} \leq \binom{m}{r}_q \leq \frac{q^2}{q^2-q-1} q^{r(m-r)}. \tag{9}$$

Since $q = 2$ and the subspace W is uniform at random, by substituting the values of $m = \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil$ and $r = \lceil (1-\lambda_1)k \rceil$ then taking the logarithm, we have

$$R^*(n, k, \lambda_1, \lambda_2) \leq \lceil (1-\lambda_1)k \rceil \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil + 2, \tag{10}$$

which proves the theorem. ∎

Remark 1: In the case where the binary vectors assigned to the users are not linearly independent, the reason for inserting additional random vectors into $\mathcal{A}'$ to increase the dimension of W up to $\lceil (1-\lambda_1)k \rceil$ is that doing so would reduce the common message length. This is because in the massive random access setting, k is at least several hundreds, so $\lceil (1-\lambda_1)k \rceil \geq \lceil \log(1/\lambda_2) \rceil$. In this case, the Gaussian coefficient is a decreasing function of the dimension of W, because $\lceil (1-\lambda_1)k \rceil > \frac{1}{2}\left(\lceil (1-\lambda_1)k \rceil + \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil\right)$. In other words, increasing the dimension of W reduces the common message length. Note that the dimension of W can be increased up to $\lceil (1-\lambda_1)k \rceil$ while still satisfying the false alarm probability bound in (**??**).

Remark 2: We can tighten the common message length bound in Theorem **??** by using a larger field size. Specifically, for any $2 < q \leq 1/\lambda_2$, we can repeat the previous proof with $d = \lceil (1-\lambda_1)k \rceil + \left\lceil \frac{\log(1/\lambda_2)}{\log(q)} \right\rceil$. The corresponding coefficient $\frac{q^2}{q^2-q-1}$ in the upper bound (**??**) is then bounded above by 2, and the resulting additive constant becomes 1, rather than 2. However, the choice of $q = 2$ has the advantage in that it gives a practical scheme with a lower complexity than if a larger field size is used, as seen in the next section.

IV. Practical Coding for Acknowledgement

The implementation of the acknowledgement scheme described in the previous section hinges upon the lossless compression of random subspaces in $\mathbb{F}_2^d$. In this section, we describe a low-complexity source coding scheme that achieves close to the theoretical bound in Theorem **??**. For notational simplicity, we set $\lambda_1 = 0$, so that $d = k + \lceil \log(1/\lambda_2) \rceil$.

A. Parity-Check Scheme

Subspaces of $\mathbb{F}_2^d$ are exactly length-d linear binary codes. Linear binary codes can be represented by RREF matrices: either in terms of a generator matrix in separable form, or in terms of a parity check matrix in separable form. Moreover, it is easy to obtain one representation from the other. We leverage this relationship to devise a practical acknowledgement scheme by compressing the subspace in $\mathbb{F}_2^d$ spanned by the binary vectors assigned to the acknowledged users in terms of the parity-check matrix of the corresponding binary linear code. At the decoder, each user simply checks whether its assigned binary vector is in the code by multiplying its binary vector with the parity-matrix. We refer to this scheme as the *parity-check scheme*.

Given a set of users $\mathcal{A} = \{a_1, \ldots, a_k\} \in \binom{[n]}{k}$ that we wish to acknowledge and a uniformly random hash function $h : [n] \to \mathbb{F}_2^{k+\lceil \log(1/\lambda_2) \rceil}$, let G be the matrix whose rows are the random vectors assigned to the users in $\mathcal{A}$, i.e.,

$$G = \begin{bmatrix} - & h(a_1) & - \\ & \vdots & \\ - & h(a_k) & - \end{bmatrix} \in \mathbb{F}_2^{k \times (k + \lceil \log(1/\lambda_2) \rceil)}. \tag{11}$$

The subspace spanned by $h(\mathcal{A})$, i.e., the row space of G, can be compressed losslessly by first row-reducing G, then describing the parity-check matrix of G in RREF. Recall from Remark **??** that it is advantageous to add random vectors to the row space of G to make it full row-rank. This turns out to be easier to do in the RREF by replacing the last set of non-pivot columns with indicator vectors corresponding to the all-zero rows of G, as illustrated in the example below.

Example 1: Suppose $k = 3$, $\lceil \log(1/\lambda_2) \rceil = 2$, and the active users a_1, a_2, a_3 have the hashes

$$\begin{aligned} h(a_1) &= [1\ 1\ 0\ 1\ 1], \\ h(a_2) &= [1\ 1\ 1\ 1\ 0], \\ h(a_3) &= [0\ 0\ 1\ 0\ 1]. \end{aligned} \tag{12}$$

The matrix of hashes is of dimension $k \times (k + \lceil \log(1/\lambda_2) \rceil)$. In this example, it is

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{13}$$

Row reducing G yields

$$\mathrm{RREF}(G) = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{14}$$

Note that G has rank 2, because the third row of $\mathrm{RREF}(G)$ is all zero. To add a random vector to the row space of G to make it full rank, we form G' by replacing the last non-pivot column (i.e., column 5) with e_3 so that

$$G' = \begin{bmatrix} 1 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{0} & 1 & \mathbf{0} & 0 \\ 0 & \mathbf{0} & 0 & \mathbf{0} & 1 \end{bmatrix}. \tag{15}$$

Next, we compute the parity-check matrix H of G' and communicate H as the acknowledgement message.

In $\mathbb{F}_2$, the parity-check matrix H can be formed from G' as follows. The columns of H corresponding to the pivot columns of G' should be the transpose of the non-pivot columns of G'. The rest of H should be an appropriate identity matrix. In this example, the parity matrix is formed from G' as

$$H = \begin{bmatrix} \mathbf{1} & 1 & \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{1} & 0 & \mathbf{0} & 1 & \mathbf{0} \end{bmatrix}, \tag{16}$$

where columns $1, 3, 5$ of H are the transpose of the non-pivot columns $2, 4$ of G' (shown in bold in (**??**) and (**??**)), and columns $2, 4$ of H are an identity matrix. One can easily check that $h(a_i)H^T = 0$ for each $i = 1, 2, 3$. Once H is communicated to the users, each user i only needs to check whether $h(a_i)H^T = 0$ to decide whether it has been acknowledged. It is easy to see that in this way the users in $\mathcal{A}$ are always correctly acknowledged. The probability that some other $u \notin \mathcal{A}$ with a random $h(u)$ is incorrectly acknowledged, i.e., it happens to have $h(u)H^T = 0$, is at most λ_2.

To communicate H to the users, we only need to specify the locations of the non-pivot columns of G', which can be represented by a binary vector of length $k + \lceil \log(1/\lambda_2) \rceil$, along with the contents of these columns, which take $k\lceil \log(1/\lambda_2) \rceil$ bits to describe. Thus, the total number of bits needed to specify H is $k\lceil \log(1/\lambda_2) \rceil + k + \lceil \log(1/\lambda_2) \rceil$. This has the same order as the common message length characterized in Theorem **??**, because $k \gg 1$ and $\lceil \log(1/\lambda_2) \rceil \gg 1$. We note here that there exist more sophisticated schemes for compressing G' that rely on recursive computation of Gaussian coefficients [**?**], [**?**], [**?**]; they achieve the common message length of Theorem **??**, but also incur additional computational complexity.

We now characterize the complexity of the encoding and decoding operations. The encoding complexity of the parity-check scheme is dominated by the complexity of row-reducing G, which requires $k^2(k + \lceil \log(1/\lambda_2) \rceil)$ bit operations. Since each user u decodes by computing $h(u)H^T$, the complexity of decoding is $\lceil \log(1/\lambda_2) \rceil (k + \lceil \log(1/\lambda_2) \rceil)$ bit operations.

Theorem 2: The proposed parity-check scheme for acknowledgement in massive random access, with missed detection proability of $\lambda_1 = 0$ and false alarm probability of λ_2, achieves a common message length of $k\lceil \log(1/\lambda_2) \rceil + k + \lceil \log(1/\lambda_2) \rceil$ with an encoding complexity of $O(k^3 + k^2 \log(1/\lambda_2))$ bit operations and a decoding complexity of $O(k \log(1/\lambda_2) + \log^2(1/\lambda_2))$ bit operations.

B. Comparison to the Acknowledgement Scheme of [?]

We now contrast the proposed parity-check scheme with the acknowledgement scheme in [**?**], which is tailored for the setting where $\lambda = \lambda_1 = \lambda_2$. The scheme in [**?**] is based on prior work in hashing by Porat [**?**]; we refer to this scheme as *Porat's scheme*. Porat's scheme is based on solving k uniformly random linear equations in k variables over a field of size at least $1/\lambda$, and achieves a common message length of $k\lceil\log(1/\lambda)\rceil$. Concretely, let q be any prime power such that $q \geq 1/\lambda$. The common randomness takes the form of two uniformly random hash functions $h_1 : [n] \to \mathbb{F}_q^k$ and $h_2 : [n] \to \mathbb{F}_q$ that are known to the BS, and such that each user u knows its own values of $h_1(u)$ and $h_2(u)$. Given an active set $\mathcal{A}$, the BS computes and transmits some $x \in \mathbb{F}_q^k$ such that $\langle h_1(u), x\rangle = h_2(u)$ for all $u \in \mathcal{A}$. By bounding the probability that $h_1(\mathcal{A})$ is a set of linearly independent vectors (see Theorem 3.1 in [**?**]) it can be shown that the probability that such an x exists is at least $1 - \lambda$. If such an x exists, it can be found using Gaussian elimination. Since a uniformly random element of $\mathbb{F}_q$ requires $\lceil\log(1/\lambda)\rceil$ bits to encode, the resulting common message length is $k\lceil\log(1/\lambda)\rceil$. If no such x exists, a missed detection error is declared. (This differs slightly from the setting of [**?**], which assumes that $\lambda_1 = 0$ and regenerates hashes until $\langle h_1(u), x\rangle = h_2(u)$ for all $u \in \mathcal{A}$.)

Both the parity-check scheme and Porat's scheme are linear algebraic in nature. Porat's scheme finds a solution to a uniformly random system of linear equations over $\mathbb{F}_q$ for some $q \geq 1/\lambda_2$. In contrast, the parity-check scheme finds a system of equations for which the random length-$(k + \lceil\log(1/\lambda_2)\rceil)$ binary vector assigned to each of the active users is a solution. In this way, the parity-check scheme can be seen as a dual of Porat's scheme. From this perspective, the parity-check scheme has several advantages. For Porat's scheme, a random system of linear equations may not have a solution, and the probability that it does is very close to the probability that the equations are linearly independent. This necessitates a larger field, so that the probability that a system of k linear equations in k variables is linearly independent is high. Since the false alarm probability in Porat's scheme is exactly $1/q$, this introduces a coupling between the two error probabilities λ_1 and λ_2. In contrast, for the parity-check scheme proposed in this paper, for any subset $S \subset \mathbb{F}_q^d$ (where q is a prime power) and any $d > 0$, there exists a system of linear equations in d variables over $\mathbb{F}_q$ for which S is a solution. This allows us to decouple the error probabilities λ_1 and λ_2, while also allowing us to work in $\mathbb{F}_2$, the field over which arithmetic can be performed with the fewest number of bit operations.

Porat's scheme requires performing Gaussian elimination on a $k\times(k+1)$ matrix whose entries are in $\mathbb{F}_q$, where $q \geq 1/\lambda_2$ is a prime power. This requires $Ck^2(k+1)$ arithmetic operations in $\mathbb{F}_q$, for some constant C. In contrast, the encoding complexity of the parity-check scheme is dominated by the complexity of Gaussian elimination on a $k \times (k + \lceil\log(1/\lambda_2)\rceil)$ matrix whose entries are in $\mathbb{F}_2$. This requires $Ck^2(k + \lceil\log(1/\lambda_2)\rceil)$ arithmetic operations in $\mathbb{F}_2$, for the same constant C. Hence, under the mild assumption that $\log(1/\lambda_2)$ scales sublinearly in k, both arithmetic complexities have the same leading term. A difference in complexity arises when considering the number of *bit operations* required to perform arithmetic over $\mathbb{F}_q$ and $\mathbb{F}_2$. Since elements of $\mathbb{F}_q$ are stored in memory as length-$\log(q)$ bit sequences, an arithmetic operation in $\mathbb{F}_q$ requires $\log(q) \geq \log(1/\lambda_2)$ bit operations, whereas an arithmetic operation in $\mathbb{F}_2$ requires just one. Therefore the parity-check scheme improves on Porat's scheme complexity-wise by a multiplicative factor of $\log(1/\lambda_2)$.

V. Converse

As the final part of this paper, we present a lower bound on $R^*(n, k, \lambda_1, \lambda_2)$ which shows that the achievability result of Theorem **??** is tight within $O(k)$ for large n. Since we consider coding schemes with common randomness, a volume bound argument that is typically used (e.g., in [**?**], [**?**], [**?**]) is not sufficient. Instead, we make the following observation. Let A be a random variable denoting the set of active users, let Z be the common randomness, which is known to both the encoder and the decoders, and let $\hat{A} = \{u \in [n] : g_u(f(A, Z), Z) = 1\}$ be the set of users who believe they have been acknowledged. When conditioned on Z, $A - f(A, Z) - \hat{A}$ form a Markov chain. Now $H(f(A, Z)) \geq H(f(A, Z)|Z)$, and by the conditional version of the data processing inequality,

$$H(f(A, Z)|Z) \geq I(f(A, Z); \hat{A}|Z) \geq I(A; \hat{A}|Z). \tag{17}$$

We further lower bound $I(A; \hat{A}|Z)$ as

$$\begin{aligned} I(A; \hat{A}|Z) &= H(A|Z) - H(A|\hat{A}, Z) &(18)\\ &= H(A) - H(A|\hat{A}, Z) \geq I(A; \hat{A}). &(19)\end{aligned}$$

Bounding $I(A; \hat{A})$ from below yields the following converse.

Theorem 3: Consider the acknowledgement problem with k active users and n total users. For any error probabilities $0 \leq \lambda_1, \lambda_2 < \frac{1}{2} - \frac{1}{k}$ such that $\lambda_2 n - \lambda_1 k \geq 0$, the minimum common message length $R^*(n, k, \lambda_1, \lambda_2)$ is lower bounded as

$$\begin{aligned} R^*(n, k, \lambda_1, \lambda_2) \geq (1 - \lambda_1)k \log\left(\frac{1}{\lambda_2 + \frac{(1-\lambda_1)k}{n}}\right)\\ - k(1 + 2\log(e)) - \log(\lambda_1 k + 1) - \log(e). \end{aligned} \tag{20}$$

The first term in (**??**) arises from the fact that $H(A|\hat{A})$ is approximately the logarithm of the number of subsets A that satisfy the error probability bounds, given $\hat{A}$. The remaining terms arise from applications of Jensen's inequality and approximations to the binomial coefficient. Note that when n grows superlinearly in k, the converse result coincides asymptotically with the achievability result in Theorem **??**.

VI. Conclusion

Motivated by massive random access applications, this paper studies efficient strategies for acknowledging k users among a large pool of n potential users. We propose a coding scheme for acknowledgement, which is equivalent to specifying a random subspace of $\mathbb{F}_2^d$ of some fixed dimension

determined by k and the probability of error. This coding scheme affords a practical low-complexity implementation through the representation of subspaces as RREF matrices. By compressing the RREF matrix losslessly, it achieves a common message length that is order optimal for large n.