
Huseyin A. Inan*, Surin Ahn*, Peter Kairouz†, Ayfer Ozgur*
*Department of Electrical Engineering, Stanford University. Emails: {hinan1, surinahn, aozgur}@stanford.edu
†Google AI, 1600 Amphitheatre Parkway, Mountain View, CA 94043. Email: kairouz@google.com

Abstract—We propose a grant-free random access scheme for short-packet communication on a collision channel without feedback, where user identities are conveyed through their activity patterns. We show that this problem is inherently related to the non-adaptive group testing problem, where the goal is to identify a small subset of defective items within a larger population, using as few (pre-determined) tests as possible. In the frame-synchronous case, where users’ transmissions are aligned, we find that any solution to the non-adaptive group testing problem is also a solution to the random access problem. Similar connections to group testing are identified in the asynchronous variant of the problem, in which case users’ transmissions within a frame are received with arbitrary and unknown delays at the receiver. We show that such delays can be accommodated without any additional penalty with respect to the scaling of the transmission length, and that in the regime where the data payload is small, the performance of the proposed random access scheme comes close to that of fully coordinated access schemes.

I. INTRODUCTION

An important challenge for upcoming wireless systems, notably 5G, is to support traffic generated by sensors and other intelligent devices, often called machine-to-machine (M2M) communications. There are a few key characteristics that distinguish M2M communications from conventional human-initiated transmissions: (i) the number of devices associated with a single access point can be massive (on the order of 100,000 nodes per cell), while at any given time only a small fraction of these devices are expected to be active; (ii) packets have small data payloads, i.e., when active, each device may have only a few bits to communicate, e.g., to signal the occurrence of an event of interest; (iii) devices can be limited in terms of their energy resources and processing capabilities, e.g., they may be optimized for uplink transmission with limited capabilities for downlink, feedback and coordination.

Traditionally, random access schemes such as ALOHA (or slotted ALOHA) [1], [2] allow nodes to send their packets without any prior resource requests to the base station (“grantless”) and resolve packet collisions by requiring users to retransmit their packets after waiting a (random) period of time. However, in the absence of a feedback channel, a retransmission-based approach may be ineffective because devices are unable to detect whether collisions have occurred. Recently introduced random access schemes, including irregular repetition slotted ALOHA [3] and coded slotted ALOHA [4], take a “forward error correction” approach in which the active users’ packets are judiciously encoded so that the receiver can decode their information contents with high probability, even if collisions have occurred on route. In these schemes, the ID of the transmitting device is explicitly stored within each transmitted segment. This approach, however, can be suboptimal in M2M applications, where the population size can be very large, and the number of bits required to encode the device ID may be comparable to, or even larger than, that of the actual packet payload.

In this paper, by building on ideas from group testing, we propose and analyze a random access scheme where the device IDs are implicitly encoded in the transmission patterns of the data packets. The related idea of conveying messages (without the device IDs) via the transmission patterns themselves was considered under a different setup in [5], [6]. Assuming that the population size is \( N \) and at most \( d \) devices can be active at any given time, we show that group testing codes can be used to design random access protocol sequences with minimal length. We consider both the frame-synchronous and asynchronous cases, where the former assumes that the transmissions of the active devices are synchronized, while the latter assumes that transmissions can have arbitrary offsets within a frame. To develop order-optimal protocol sequences for the frame-asynchronous case, we build on the binary constant-weight cyclic codes developed in [7], and through a novel analysis we characterize the performance of these combinatorial codes in a probabilistic setting. A similar approach has been recently used in [8] (by a subset of the authors) to develop the first order-optimal, strongly explicit construction for probabilistic group testing. Our results show that this approach can achieve an efficiency close to that of fully coordinated protocols when the data payload is small.

Our frame-asynchronous setting subsumes the asynchronous node identification problem considered in [9]. The solution in [9] leads to protocol sequences of length \( t = \Delta + \Theta(d \log d \log N) \), where \( \Delta \) is the maximum offset between the transmissions of the devices. In contrast, our approach leads to protocol sequences of length \( t = \Theta(d \log N) \) for the same problem, providing a \( \log d \) improvement and matching the synchronous case. Note that in our solution \( t \) does not depend on \( \Delta \), and thus the protocol sequences can be designed independently of \( \Delta \) which may change over time due to network dynamics. Furthermore, our construction is explicit whereas the construction in [9] is randomized. On the other hand, the decoding complexity in [9] is linear in the length of the codewords whereas in our case it is \( O(\Delta + t^2 N) \).
Assume we have a total population of $N$ users connected to a central node, and that each user occasionally has a data packet to transmit over a common communication channel. In a similar flavor to slotted ALOHA protocols [11]–[4], [10], we consider a slotted random access setting where each slot duration is assumed to be the transmission time for one packet and transmissions are slot-synchronized, i.e., the transmission of a packet occurs exactly within a slot. We adopt a simple collision model for the transmissions. In a particular slot, the receiver detects no transmission if there is no user sending a packet, detects a single packet if there is exactly one user sending a packet, or detects a collision if two or more users are sending packets in that slot, in which case the colliding packets are lost. There is no feedback available to inform the users of the channel outputs in previous slots. This model is similar to the collision channel without feedback introduced in [11]. The goal is to convey the packets of the transmitting users, as well as their identities, to the receiver. In traditional random access protocols (e.g., ALOHA), the identity of the transmitting node is included in the transmitted packet. This can be suboptimal when the data payload is small, given that each packet may need to be retransmitted multiple times due to collisions. Here, we explore an alternative scheme where the user IDs are conveyed through the transmission patterns of the users, rather than stored in the packets.

As a starting point, we first consider the synchronous case where the slots are grouped in frames, all having the same length $t$ (in slots), and all the users are frame-synchronous. We assume that in each frame at most $d$ of the $N$ users attempt a packet transmission. Users attempting a packet transmission in a particular frame are referred to as the active users for that frame. The active users in each frame want to convey a single data packet along with their identity to the receiver, and consider $A$ to be the protocol sequence of user $i$ for $i = 1, 2, \ldots, N$. For a particular frame, an active user (say $i$) transmits its packet in slots $1 \leq j \leq t$ for which $M_{ij} = 1$, and stays silent otherwise. Note that if user $i$ is active and $M_{ij}$ has Hamming weight $w_i$, then user $i$ will repeat its packet $w_i$ times. The receiver is assumed to know the users’ protocol sequences. Failure occurs if the receiver fails to recover the identities of the active users or decode their corresponding packets. The goal is to design the protocol sequences of the transmitters to have minimal length $t$ while ensuring that the failure probability, denoted by $P_e$, is less than a given fixed $\epsilon$ such that $0 < \epsilon < 1$ when the active set of size $d$ is chosen uniformly at random among the $N$ devices.\footnote{This can be simply extended to a setting with an active set of size at most $d$, and the results are the same for the two cases in terms of scaling. Due to space limitations, we consider the size of the active set as $d$ in what follows.}

The below proposition provides the minimal required value for $t$ in the above problem by establishing its equivalence to the group testing problem.

**Proposition 1:** When $d = O(N^{\alpha})$ for some $\alpha \in (0, 1)$, $t = \Theta(d \log N)$ is necessary and sufficient to achieve $P_e \leq \epsilon$ with the scheme above for the synchronous random access problem.

Note that $d \log N$ bits are needed to uniquely identify $d$ devices in a population of $N$. Hence, if each packet contains $B$ bits, the scheme achieving $t = \Theta(d \log N)$ in Proposition 1 can be thought of as communicating $d(B + \log N)$ bits in $\Theta(d \log N)$ slots. Note that transmitting $d(B + \log N)$ bits on a noiseless channel with capacity $B$ bits per slot would require

$$\frac{d(B + \log N)}{B} = d \left(1 + \frac{\log N}{B}\right)$$

Hence, in the limit when $B$ is very small, i.e., the data payload is small, the information rate of the random access scheme in Proposition 1 comes close order-wise to the capacity of the channel (or equivalently that of a perfectly coordinated access protocol).

**Proof:** We first briefly describe the non-adaptive group testing problem and then prove the proposition by building on ideas from this framework. Consider a $t \times N$ binary matrix $M = [M_1, \ldots, M_N]$ and define the random set $A$ consisting of $d$ random columns of $M$, i.e., $A$ is uniformly distributed over all possible sets of size $d$ among $N$ columns. Let $Y = \bigvee_{i \in A} w_i$. In words, $Y$ is the Boolean OR combination of $d$ random columns of $M$. The non-adaptive group testing problem focuses on designing $M$ with a decoding rule such that estimating the set $A$ by observing $Y$ (along with the knowledge of $M$) can be done with a small but fixed error probability, $\epsilon$. It is shown in [8], [12]–[14] that $t = \Theta(d \log N)$ is necessary and sufficient for this problem.

We will next show that both the identification and decoding problems in the random access setting can be solved by designing the protocol sequences according to a group testing matrix $M$. We take $M_i$ to be the protocol sequence of user $i$ and consider $A$ to be the random active set of size $d$. Consider the following strategy for the receiver. For any particular frame, we construct the binary sequence $Y$ of length $t$ such that $Y_i = 0$ if there is no transmission in slot $i$ and $Y_i = 1$ otherwise, for $1 \leq i \leq t$. The receiver will determine the active users by looking at $Y$ and using its knowledge of the protocol sequences $M_i$ for $i \in [N]$, where $[N] \triangleq \{1, 2, \ldots, N\}$. Note
that \( Y \) corresponds to the Boolean OR combination of the protocol sequences corresponding to the active users. We will employ the **cover decoder**, a decoding rule which is widely used in the group testing literature. We say that a set of binary columns \( M_1, \ldots, M_d \) covers a binary column \( M \) if \( \bigvee_{j \in [d]} M_j \) or \( M_i = \bigvee_{j \in [d]} M_j \). The cover decoder simply scans through the columns of \( M \) and checks whether or not \( Y \) covers a particular column. If column \( i \) is covered by \( Y \), then user \( i \) is declared active. We note that \( Y \) covers all protocol sequences of the active users. Therefore, failure happens if \( Y \) additionally covers the protocol sequence of any non-active user. We further note that if the identification is done correctly, the packet of an active user \( i \in \mathcal{A} \) can be decoded if its protocol sequence \( M_i \) is not covered by the protocol sequences of the rest of the active users, i.e., \( \mathcal{A} \setminus \{i\} \). Hence, by the union bound, the probability of error can be bounded as

\[
P_e \leq \sum_{i \in \mathcal{A}} \Pr (\mathcal{A} \text{ covers } M_i) + \sum_{i \in \mathcal{A}} \Pr (\mathcal{A} \setminus \{i\} \text{ covers } M_i)
\]

\[
\leq \sum_{i \in [N]} \Pr (\mathcal{A} \setminus \{i\} \text{ covers } M_i)
\]

where \( \mathcal{A} \setminus \{i\} \) is a set of \( d \) random columns in \([N] \setminus \{i\}\). The inequality (1) holds since the probability of being covered increases as the size of the random set increases, and since \( |\mathcal{A} \setminus \{i\}| = d - 1 \) while \( |\mathcal{A} \setminus \{i\}| = d \). It is shown in [8], [12]–[14] that (1) can be bounded by an arbitrary but fixed \( \epsilon \) when \( t = \Theta(d \log N) \). This completes the sufficiency.

For the necessity part, we use \( X \) to denote the binary vector of length \( N \) with \( d \) non-zero positions chosen uniformly at random, indicating the random active users. Let \( \tilde{Y} \) be a vector of length \( t \) denoting the channel output with 3 possible states in each entry and let \( \tilde{X} \) denote our estimate for \( X \) based on \( \tilde{Y} \). Note that \( X \rightarrow \tilde{Y} \rightarrow \tilde{X} \) forms a Markov chain, and by standard information-theoretic definitions we have \( H(Y) = H(X; \tilde{X}) + I(X; \tilde{Y}) \). We have \( H(X) = \log(N) \) and by Fano’s inequality, \( H(X; \tilde{X}) \leq 1 + \epsilon \log \frac{N}{d} \). By the data processing inequality we get \( I(X; \tilde{Y}) \leq I(X; \tilde{Y}) \). Combining these with \( \Pr (\tilde{Y} \rightarrow \tilde{X}) \geq (N/d)^d \), it follows that \( t = \Theta(d \log N) \) is required.

### III. Asynchronous Random Access

We now extend the random access model from the previous section to the asynchronous setting where the users’ transmissions are not necessarily aligned within a frame but are still assumed to be slot-synchronous. As before, an active user transmits its packet by using its assigned protocol sequence. However, we now assume that the transmission of each user is received with an arbitrary delay (in slots), e.g., due to different path delays for different nodes. We assume that the delays of different transmitters can be arbitrary and unknown (both to the transmitters and the receiver), but that the receiver has an upper bound on the maximal possible delay in the system, which we denote by \( \Delta \) slots. This model simplifies communication by not requiring the receiver to estimate the propagation delays of different users or to synchronize the network, which can be impractical. As before, we assume that in each frame at most \( d \) of the \( N \) users attempt a packet transmission, and the goal is to recover the user IDs and their packets at the receiver with \( P_e \leq \epsilon \), assuming that the active set of size \( d \) is chosen uniformly at random among the \( N \) devices. Fig. 2 depicts an example of this asynchronous random access model. The following theorem is our main result for this setting.

**Theorem 1:** Let \( c > 8 \) be a constant. With protocol sequences of length \( t = cd \log N \), a vanishing probability of error can be achieved in the asynchronous random access problem in the regime \( d = \Omega(\log^2 N) \), \( d = o(\sqrt{N}) \).

It is interesting to note that no additional cost is incurred in terms of the scaling of \( t \) when the model from Section II is made asynchronous. Our proof builds on a combinatorial code construction presented in [7]. We analyze its performance in a probabilistic setting when the active set is chosen uniformly at random, and characterize its scaling performance for a small failure probability. Our approach applies to the asynchronous node identification problem considered in [9] as a special case (by focusing only on the identification problem based on \( Y \), as in the proof of Proposition 1, and ignoring the packet decoding problem). The solution developed in [9] for the identification problem leads to protocol sequences of length \( t = \Delta + \Theta(d \log d \log N) \), which depends on \( \Delta \) and incurs an additional \( \log d \) factor. The proof is provided in the following section.

### IV. Proof of the Main Result

We will provide the proof in two steps. We first describe one of the constructions presented in [7] (Construction I) and then building on this construction we characterize its performance in a probabilistic setting.

**A. Construction I of [7]**

In [7], a binary constant-weight cyclic code is constructed by taking a cyclic Reed-Solomon (RS) code [15] and mapping its codewords to constant-weight binary codewords in a way that provides the cyclic property in the corresponding binary code. In the following, we describe this method. We begin with the definition of Reed-Solomon codes.
since they can simply be satisfied by choosing 3. Since the codeword c, first 3 × 2 matrix A is created and then mapped to the final binary codeword b as described in Section IV-A.

Definition 1: Let \( F_q \) be a finite field and \( \alpha_1, \ldots, \alpha_n \) be distinct elements from \( F_q \). Let \( 1 \leq k < n \leq q \). The Reed-Solomon code of dimension \( k \) over \( F_q \), with evaluation points \( \alpha_1, \ldots, \alpha_n \) is defined with the following encoding function. The encoding of a message \( m = (m_0, \ldots, m_{k-1}) \) is the evaluation of the corresponding \( k - 1 \) degree polynomial \( f_m(X) = \sum_{i=0}^{k-1} m_i X^i \) at all the \( \alpha_i \)'s:

\[
RS(m) = (f_m(\alpha_1), \ldots, f_m(\alpha_n)).
\]

In [7], they start with a cyclic \([n, k]\)q RS code that contains the all-ones codeword \( \mathbf{1} \) where \( q \) is an odd prime, \( n \) is a divisor of \( q - 1 \), and \( k \) satisfies \( 1 \leq k < n \). Each codeword of length \( n \) is then mapped to a binary codeword of length \( nq \) as follows. For each codeword, at first a \( q \times n \) matrix is created where each column is a unit weight binary vector of length \( q \) representing the corresponding symbol in the codeword. The representation is via the “identity mapping” which takes a symbol \( i \in [q] \) and maps it to the vector in \([0, 1]^q \) that has a 1 in the \( i \)th position and zero everywhere else. Denoting such a \( q \times n \) binary matrix as \( A \), a corresponding \( nq \) binary vector \( b \) is constructed as \( b_i = A(i \mod q, i \mod n) \) for \( i \in [nq] \). An example of this procedure is provided in Figure 3. When the positive integers \( q \) and \( n \) are relatively prime, i.e., \( \gcd(q,n) = 1 \), then the Chinese remainder theorem specifies a one-to-one correspondence between such matrices and \( nq \)-dimensional vectors. It is shown in Theorem 1 of [7] that this leads to a binary constant-weight cyclic code with \( q^k \) codewords of blocklength \( nq \).

B. Proof of Theorem 1

We consider the binary constant-weight cyclic code described in Section IV-A. We distribute the cyclically distinct codewords (i.e., no codeword can be obtained by the cyclic shifting, one or more times, of another codeword) of this code to the users. Therefore, this can be considered as assigning a codeword with all its cyclic shifts to a single user. An active user randomly selects a protocol sequence among its codeword and all the cyclic shifts and transmits its packets based on this protocol sequence. Based on the construction described in Section IV-A, the length of the protocol sequences is \( t = nq \). We shall later see that the choice \( q = 4d \) and \( n = \Theta(\log N) \) is appropriate, therefore, leading to \( t = \Theta(d \log N) \) length protocol sequences2. Since a column can have at most \( t \) cyclic shifts and we need to support \( N \) users, \( k \) should be selected such that \( q^k/t \geq N \). It follows that \( k = O(\log N / \log d) \), which satisfies the constraint \( 1 \leq k < n \).

We next describe the decoding process. Similar to the synchronous case, for any particular frame we construct the binary sequence \( Y \) of length \( t + \Delta \) such that \( Y_{i} = 0 \) if there is no transmission in slot \( i \) and \( Y_{i} = 1 \) otherwise for \( 1 \leq i \leq t + \Delta \). We next convert this to the binary sequence \( \tilde{Y} \) of length \( t \) by calculating

\[
\tilde{Y}_i = \bigvee_{j=0}^{\Delta/t} Y_{i+jt},
\]

for \( 1 \leq i \leq t \) and we set \( Y_i = 0 \) if \( i > t + \Delta \). In words, we are taking the Boolean OR combination of the blocks of the frame where the block size is \( t \) consecutive slots. Note that due to the transmission patterns and arbitrary offsets of these transmissions, \( \tilde{Y} \) corresponds to the Boolean OR combination of the protocol sequences (or any of their cyclic shifts) of \( d \) random active users.

We denote \( M \) as the binary constant-weight cyclic code described in Section IV-A and observe that \( \tilde{Y} \) is the Boolean OR combination of \( d \) random columns of \( M \). Note that this may include cases where a column is a cyclic shift of another column which is not possible in our problem formulation for \( \tilde{Y} \) since users have cyclically distinct codewords. However we note that a column can have at most \( t \) cyclic shifts and using union bound, the probability of this event can be bounded by \((\frac{\Delta}{t})^d \frac{n}{N} \) which asymptotically vanishes when \( d = o(\sqrt{N}) \).

We define the random set \( A \) consisting of \( d \) random columns of \( M \) (corresponding to random active set of size \( d \)). We will similarly employ the cover decoder. Note that \( \tilde{Y} \) covers (potentially cyclic shifts) of the protocol sequences of the active users. Therefore, failure happens if \( \tilde{Y} \) further covers any cyclic shift of the protocol sequence of any non-active user.

Given that the identification is done correctly, the decoding can be performed for each active user if \( i \) cannot be covered by \( A \backslash \{i\} \) for all \( i \in A \). Hence, using the union bound, we can bound the probability of error similar to the proof of Proposition 1 as

\[
P_e \leq \sum_{i \in [q^k]} \Pr \left( \tilde{M}_i \text{ is covered by } A_{[q^k] \backslash \{i\}} \right),
\]

where \( A_{[q^k] \backslash \{i\}} \) is a set of \( d \) random columns in \([q^k] \backslash \{i\}\).

We next consider the condition for which \( \tilde{M}_i \) is covered by \( A_{[q^k] \backslash \{i\}} \). Note that this happens when each non-zero entry of \( \tilde{M}_i \) has a corresponding non-zero entry in one of the columns in \( A_{[q^k] \backslash \{i\}} \) in the same row. We claim that this is equivalent to the condition that the symbols of \( \tilde{M}_i \) are contained in the union of symbols of \( A_{[q^k] \backslash \{i\}} \) in the RS code for all rows in \([n]\). Indeed, due to the identity mapping which creates the \( q \times n \) matrix \( A \) and applying the same vectorization rule \( b_i = A(i \mod q, i \mod n) \) for \( i \in [nq] \), it follows that if there is a symbol of \( \tilde{M}_i \) in the RS code that is not included in the union of symbols of \( A_{[q^k] \backslash \{i\}} \) for some row in \([n]\), the corresponding position of one in the final codeword \( b \) will not be covered by \( A_{[q^k] \backslash \{i\}} \).

\[
c = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\]

Fig. 3. An example of the method introduced in [7] that maps the codewords of the RS code to binary constant-weight codewords. In this example, \( n = 2 \) and \( q = 3 \). From the codeword \( c \), first 3 × 2 matrix \( A \) is created and then mapped to the final binary codeword \( b \) as described in Section IV-A.

\[\alpha \in \mathbb{F}_q \implies \tilde{\alpha} = \alpha \mod q \]

2We can safely ignore the constraints that \( q \) is an odd prime and \( n|q - 1 \) since they can simply be satisfied by choosing \( q \) as the smallest prime greater than \( 4d \) and noting that \( d = \Theta(\log^2 N) \).
We now fix any \( n \) evaluation points \( \alpha_1, \alpha_2, \ldots, \alpha_n \) from \( \mathbb{F}_q \) in the RS code. We denote \( \Psi \triangleq \{ \alpha_1, \alpha_2, \ldots, \alpha_n \} \). Recall that there is a \( k-1 \) degree polynomial \( f_m(x) = \sum_{i=0}^{k-1} m_i x^i \) corresponding to each column in the RS code and the corresponding symbols in the column are the evaluation of \( f_m(x) \) at \( \alpha_1, \alpha_2, \ldots, \alpha_n \). Denoting \( f_{M_i} \) as the polynomial corresponding to the column \( M_i \), we have

\[
\Pr \left( M_i \text{ is covered by } A[q^i \setminus \{i\}] \right) = \Pr \left( f_{M_i}(x) \in \{ f_j(x) : j \in A[q^i \setminus \{i\}] \} \forall \alpha \in \Psi \right) = \Pr \left( 0 \in \{ f_j(x) - f_{M_i}(x) : j \in A[q^i \setminus \{i\}] \} \forall \alpha \in \Psi \right) = \Pr \left( 0 \in \{ f_j(x) : j \in A \} \forall \alpha \in \Psi \right)
\]

where in the last step the random set of polynomials \( \{ f_j(x) : j \in A \} \) is taken as \( d \) nonzero polynomials of degree at most \( k-1 \). We define the random polynomial \( h(x) = \prod_{j \in A} f_j(x) \).

Note that

\[
0 \in \{ f_j(x) : j \in A \} \forall \alpha \in \Psi \Leftrightarrow h(\alpha) = 0 \forall \alpha \in \Psi.
\]

We next bound the number of roots of the polynomial \( h(x) \). Let \( r \) denote the number of roots of a random nonzero polynomial over \( \mathbb{F}_q \) of degree at most \( k-1 \). We note that \( \mathbb{E}[r] \leq 1 \) since there is exactly one value of \( m_0 \) such that \( f_m(x) = 0 \) for any fixed \( X \in \mathbb{F}_q \) and \( m_1, \ldots, m_{k-1} \).

Furthermore, from Lemma 3.9 of [16], the probability that a random nonzero polynomial over \( \mathbb{F}_q \) of degree at most \( k-1 \) has exactly \( l \) distinct roots is bounded by \( 1/l! \) for all primes \( q \) and integers \( l, k \). Therefore \( \mathbb{E}[r^2] \leq \sum_{i=1}^{k-1} i^2 = \frac{k(k-1)}{2} \leq 2e. \) We denote \( r_k \) as the number of roots of the polynomial \( f_l(x) \) and \( r_h \) as the number of roots of the polynomial \( h(x) \). Note that \( r_h \leq \sum_{j \in A'} r_j \). By using the Bernstein concentration bound [17], we get:

\[
\Pr \left( \sum_{j \in A'} r_j > 2d \right) \leq \Pr \left( \frac{1}{d} \sum_{j \in A'} (r_j - \mathbb{E}[r_j]) > 1 \right) \leq \exp \left( -\frac{d}{4e} \right) \cdot \exp \left( -\frac{d}{4e + k(2/3)} \right).
\]

Hence, when \( d = \Omega(\log^2 N) \), the last quantity is bounded by \( N^{-\log^d} \) for some constant \( c > 0 \). Thus the number of roots of \( h(x) \) is bounded by \( 2d \) with high probability. Since we pick the nonzero polynomials \( \{ f_j(x) : j \in A' \} \) randomly, due to the symmetry in the position of the roots, the probability of satisfying \( h(\alpha) = 0 \) for all \( \alpha \in \Psi \) is bounded by the probability of covering \( n \) elements from a field of size \( q \) by picking \( 2d \) elements randomly without replacement. If we ensure \( n \leq 2d \), then it follows that

\[
\Pr \{ h(\alpha) = 0 \forall \alpha \in \Psi \mid r_h \leq 2d \} \leq \max_{n \leq 2d} \Pr \{ h(\alpha) = 0 \forall \alpha \in \Psi \mid r_h = l \} = \max_{n \leq 2d} \left( \frac{(q-n)}{(l-n)} \right)^{\frac{q}{l}}.
\]

Let us fix \( q = 4d \). We then have

\[
\Pr \{ h(\alpha) = 0 \forall \alpha \in \Psi \mid r_h \leq 2d \} \leq \left( \frac{4d-n}{2d-n} \right)^{4d/2d} \leq 2^{-n}.
\]

Therefore, it follows that

\[
\Pr \{ M_i \text{ is covered by } A[q^i \setminus \{i\}] \} \leq \Pr \{ \text{Max} h(\alpha) = 0 \forall \alpha \in \Psi \mid r_h \leq 2d \} + \Pr \{ r_h > 2d \} \leq 2^{-n} + \frac{N^{-\log^d} \cdot 1}{2d}.
\]

Applying the summation over all \( i \in [q^k] \), we obtain \( P_c \leq \Omega(\log^d N)+\frac{q^k}{2d} \). Therefore, under the regime \( d = \Omega(\log^2 N) \), the average probability of error can be bounded as \( P_r \leq \Omega(\log^d N) + \Delta \delta \) by choosing \( n = (2 + \delta) \log N \). The condition \( n \leq 2d \) required in the proof is also satisfied under this regime. Note that this will lead to protocol sequences of length \( t = nq = \Omega(d \log N) \).

REFERENCES


