On Multiuser Write-Efficient Memories

Rudolf Ahlswede and Zhen Zhang

Abstract—Continuing our earlier work on write-efficient memories (WEM), we introduce new models, where several persons use the same storage device. At any time instant, exactly one of a prescribed set of users has access to the memory, but there is no protocol which determines the moving order. Among the constraints we analyze, the most interesting one is a complete privacy protection. While a user stores new data, he has to guarantee that those of the others do not get distorted. This leads to fascinating new coding problems. We provide several code constructions, as well as abstract performance bounds.

Index Terms—Multiuser memories, WEM, WUM, privacy constraints, code constructions, identification.

I. INTRODUCTION: MULTIUSER MEMORIES WITH
CONSTRAINTS ON PRIVACY, HIERARCHY,
AND TECHNOLOGY

Write-efficient memories (WEM) were introduced in [2] as a model for storing and updating information on a rewritable medium. There is a cost \( \varphi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \) assigned to changes of letters. A collection of subsets \( \mathcal{S} = \{ C_i : 1 \leq i \leq M \} \) of \( \mathcal{X}^n \), called clouds, is an \((n, M, D)\) WEM code, if \( C_i \cap C_j = \emptyset \) for all \( i \neq j \) and if

\[
D_{\text{max}} = \max_{1 \leq i, j \leq M} \min_{x^i \in C_i, y^j \in C_j} n \varphi(x_i, y_j) \leq D. \quad (1.1)
\]

\( D_{\text{max}} \) is called the maximal correction cost with respect to the given cost function. The performance of a code \( \mathcal{S} \) can also be measured by two parameters, namely, the maximal per-letter cost \( d_{\varphi} = n^{-1}D_{\text{max}} \) and the rate of the size \( r_{\varphi} = n^{-1} \log M \). The rate achievable with a maximal per-letter cost \( d \) is thus

\[
R(d) = \sup_{\mathcal{S} : d_{\varphi} \leq d} r_{\varphi} \quad (1.2)
\]

This is the most basic quantity (the storage capacity) of a WEM \((\mathcal{X}^n, \varphi)\).

For a WEM code \( \mathcal{S} \), the average correction cost \( D_{\text{ave}} \) can be defined as

\[
D_{\text{ave}} = \frac{1}{M^2} \sum_{1 \leq i, j \leq M} \frac{1}{|C_i|} \sum_{x^i \in C_i} \sum_{y^j \in C_j} \min_n \varphi(x_i, y_j) \quad (1.3)
\]

The rate achievable with an average per-letter cost \( d \) is thus

\[
R(d) = \sup_{\mathcal{S} : d_{\varphi} \leq d} \frac{1}{M^2} \sum_{1 \leq i, j \leq M} \frac{1}{|C_i|} \sum_{x^i \in C_i} \sum_{y^j \in C_j} \min_n \varphi(x_i, y_j) \quad (1.4)
\]

and the average per-letter cost can be defined as

\[
d_{\varphi} = n^{-1}D_{\text{ave}}.
\]

The rate achievable with an average per-letter cost \( d \) is thus

\[
R(d) = \sup_{\mathcal{S} : d_{\varphi} \leq d} \frac{1}{M^2} \sum_{1 \leq i, j \leq M} \frac{1}{|C_i|} \sum_{x^i \in C_i} \sum_{y^j \in C_j} \min_n \varphi(x_i, y_j) \quad (1.5)
\]

The main result of [2] is

\[
R(d) = R(d) = \sup_{P_{XY} \in \mathcal{P}_d} H(Y|X) \quad (1.6)
\]

where \( \mathcal{P}_d \) is a set of bivariate distributions

\[
\mathcal{P}_d = \{ P_{XY} \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) : P_Y = P_X, \mathbb{E}_X(X, Y) \leq d \}, \quad (1.7)
\]

\( P_{XY} \) is the distribution of a pair of random variables \((X, Y)\) with values in \( \mathcal{X} \times \mathcal{X} \), \( P_X \) (resp., \( P_Y \)) is the distribution of \( X \) (resp., \( Y \)), \( H(Y|X) \) is the conditional entropy of \( Y \) given \( X \), and \( \mathbb{E} \) denotes the expected value.

An essential ingredient of this result is a combinatorial result. We say that the hypergraph \((\Omega, \mathcal{S})\) carries \( M \) colors if there is a vertex coloring with \( M \) colors such that all of these colors occur in every edge. Let \( M(\Omega, \mathcal{S}) \) be the maximal number of colors carried by \((\Omega, \mathcal{S})\).

Coloring Lemma: The hypergraph \((\Omega, \mathcal{S})\) carries \( M \) colors if \( M \leq (\ln|\Omega|)^{-1} \min_{\mathcal{E} \in \mathcal{S}} |\mathcal{E}| \) and \( M \geq 2 \).

Since, in typical applications, the quantities \(|\Omega|\) and \(|\mathcal{E}|\) grow exponentially in the block length \( n \), we have \( M(\Omega, \mathcal{S}) \sim \min_{\mathcal{E} \in \mathcal{S}} |\mathcal{E}| \). We remark that determining \( M(\Omega, \mathcal{S}) \) for any hypergraph \((\Omega, \mathcal{S})\) induces problems of the Ramsey type.

More generally, one can consider more general cost functions

\[
\varphi^n : \mathcal{X}^n \times \mathcal{X}^n \to \mathbb{R}_+, \quad n \in \mathbb{N},
\]

and, even more generally, one can combine this WEM model with the stochastic model of [12].

However, for the sake of simplicity, in this paper we stick to our original definition of a WEM. Genuine examples are the Hamming WEM, specified by \( \mathcal{S} = \{0, 1\} \), and the cost function \( \varphi = d_H \), the Hamming distance, and the (symmetric alternating) WUM (see [3]–[5], [9], [10], [12]) specified by

\[
\varphi(x, y) = \begin{cases} 0, & \text{if } (x, y) \neq (1, 1), x, y \in \{0, 1\} \\ \infty, & \text{if } (x, y) = (1, 1). \end{cases}
\]

(1.9)
The connection of this definition to the original definition—which comes directly from the coding problem—is clarified in the passage before Theorem K in Section IV.

Here, we follow the natural idea of letting several users operate with the same memory, that is, the space \( \mathcal{P} \). Several constraints on the mode of these operations are considered and lead, perhaps surprisingly, to some seemingly original problems. One assumption is basic and will always be made:

The users, say \( U_1, U_2, \cdots, U_s \), follow no protocol, which determines in which order they store messages. At any time instant, exactly one user has access to the memory. We address questions, which are meaningful, if the user at any time instant is any member of the prescribed set of users.

We speak of a multiuser WEM and discuss possible constraints.

**Constraints on Technology:** It was laser technology which led to the WUM model and thus to a specific cost function in the WEM model. It is reasonable to assume that different users have different cost functions and different thresholds for the total costs.

**Common Messages:** All users are supposed to write and read the same messages at any time instant.

**Separate Messages Without Protection:** All users always have their own messages to store. However, a user can only read what he wrote formerly, if this was not distorted by another user in the meantime. If storage is a very rare event, this still may be practical to a certain extent.

**Separate Messages with Protection—Privacy:** Suppose that there are two users, \( U_1 \) and \( U_2 \). At any time, where one user updates his memory with a new message, he has to guarantee that the message stored by the other user does not get distorted. In this sense, users respect each other's privacy.

**Separate Messages with One-Sided Protection—Hierarchy:** Here, user \( U_2 \) has to respect the privacy of user \( U_1 \), but not vice versa.

**The Hierarchy Graph:** It is conceivable that any two users, say \( U_i \) and \( U_j \), do not mutually respect their privacy, that they both respect the privacy of the other, and finally, that \( U_1 \) respects the privacy of \( U_j \) but not vice versa. This can be symbolized by an undirected edge, an edge with two arrows, and a directed edge, respectively, in a graph with vertex set \( \{U_1, \cdots, U_s\} \). Thus, the three foregoing cases all become special cases of this general model.

**Side Information:** The issue of side information has played an important role in multiuser source and channel coding. We use the notation \( E_+ \) (resp., \( D_+ \)), if the encoder (resp., decoder) has side information, and the notation \( E_- \) (resp., \( D_- \)), if the encoder (resp., decoder) does not have side information. For memory cells, the side information refers to the knowledge of the contents of the memory before a new action (encoding or decoding) is taken. The results for WEM codes stated above concern the case \( (E_+, D_-) \), which seems to be the most natural. Of course, all kinds of other situations with side information can be imagined—for instance, those where the encoder knows time, that is, he knows how many updates have been made.

**Identification:** We address here for the first time storage for the purpose of identification. The theory of identification via channels [18] is paralleled by a theory of identifications via memories. It should be mentioned that there are also multimeasure problems. Some are formulated in Section IX. Finally, we mention that the area of combinatorial extremal problems already has received a significant impetus from the theory of memories (cf. [6]–[8], [17]). In Section II, further questions are asked which are linked to multiuser memories.

The paper is organized as follows.

In Section II, we present a capacity formula for the WEM with several users having common messages. It is in the spirit of the coding theorem for compound channels.

Next, we present in Section III a multiuser rate region for separate messages without protection. The hard work starts in Section IV with the analysis of a WEM used by two persons, who respect each other's privacy. In particular, we derive bounds on the rate regions for cost functions satisfying the triangle inequality. We also take a closer look at the WUM. By the probabilistic method, we achieve rate pairs beyond the time-sharing bound. In another direction, we show that the sum of the rates can be made to approach the maximal possible rate \( \log |\mathcal{P}| \) as the number of users increases. Originally, we conjectured this for the WUM, and there it was proved by Koschinck [11]. Serious work remains to be done. Some combinatorial problems are extracted in Section IX.

In Section V, we discuss the issue of hierarchy, and in Section VI, we focus on side information. Whereas all results indicated so far are for the case \( (E_-, D_-) \), we discuss other cases in Section VI. We present several coding theorems. Some of them in the case \( (E_-, D_-) \) are just consequences of the Diatomic Theorem of [16]. Previously for the single-user WUM, this was an outstanding problem (see [9], [12]).

In section VII, we go in a new direction. We take the first steps in a theory of identification and memories. As previously done in [18] and [19], we again establish second-order coding theorems.

In Section VIII, we provide constructions of multiuser WUM codes under the complete privacy constraint. Several open problems are stated in Section IX. Finally, we include in an Appendix two old but unpublished constructions of the second author for single-user WUM codes.

### II. Common Messages

User \( U_k \) \( (k = 1, 2, \cdots, K) \) has his cost function \( \varphi_k : \mathcal{P} \times \mathcal{X} \rightarrow \mathbb{R}_+ \). Furthermore, he has his own cost constraint. A common \( \{M_i, (D_{j, k})_{j=1}^M \} \) WEM code is now a collection of subsets \( \mathcal{V} = \{C_i : 1 \leq i \leq M \} \) of \( \mathcal{P}^n \), called clouds, with \( C_i \cap C_j = \emptyset \) for all \( i \neq j \) and with

\[
D_{k, \text{max}} = \max_{1 \leq i, j \leq M} \min_{x^k \in C_i, y^k \in C_j} \sum_{i=1}^n \varphi_k(x_i, y_i) \leq D_k \tag{2.1}
\]

for \( k = 1, 2, \cdots, K \).
As in Section I, \( r_\varphi = n^{-1} \log M \) is the rate of this code and \( \bar{d}_\varphi = n^{-1} (D_{1, \max}, \ldots, D_{K, \max}) \) is the vector of maximal per-letter costs. The rate achievable with a maximal per-letter cost vector \( \bar{d} = (d_1, \ldots, d_K) \) is thus
\[
R(\bar{d}) = \sup_{\varphi : \bar{d} \leq \bar{d}_\varphi} r_\varphi.
\tag{2.2}
\]

As in (1.7), we now define for every \( k \) a set of bivariate distributions
\[
\mathcal{P}_{d_{i,k}} = \left\{ P_{XY} \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) : P_X = P_{Y}, E_{\varphi}(X, Y) \leq d_k \right\}. \tag{2.3}
\]
Furthermore, we set
\[
\mathcal{P}_d = \bigcap_{k=1}^K \mathcal{P}_{d_{i,k}}.
\tag{2.4}
\]
Here is our first result.

**Theorem I:** The optimal rate for a common code is
\[
R(\bar{d}) = \sup_{P_{XY} \in \mathcal{P}_d} H(Y|X).
\]

**Proof:** Replacing in [2, sect. 4–6] \( S_\varphi(x^n) = \{ y^n \in \Omega : \varphi(x^n, y^n) \leq nd_k \} \) in [2, eq. (4.1)] by
\[
S_\varphi(x^n) = \{ y^n \in \Omega : \varphi_k(x^n, y^n) \leq nd_k \} \quad \text{for } k = 1, \ldots, K
\tag{2.5}
\]
where \( \Omega = \bigcup_{i=1}^M C_i \), the proofs in conjunction with a standard argument of simultaneity also settle the present case. We skip the details because they are routine.

We have to keep in mind that we have just considered the case in which the encoder knows the previous state of the memory and the decoder does not, that is, the case \( (E_n, D_n) \).

Neither the encoder nor the decoder has any knowledge about the user. The code concept allows the user to write on the memory and to read its content in an arbitrary moving order. If, now, the encoder and the decoder know the user at each move, they can achieve as a "one-shot" rate
\[
\min_k \sup_{P_{XY} \in \mathcal{P}_{d_{i,k}}} H(Y|X) \geq R(\bar{d}).
\]

We come across the following question.

**Problem 1:** What is the optimal rate in this situation for any number of moves?

We also would like to see the next question answered.

**Problem 2:** The users want to store their own messages without protection. They cannot operate beyond \( R(d_i)(d_i) \) if \( R(d_i) \) is the optimal rate for a single user \( U_i \) at threshold \( d_i \). These optima are achieved at different letter frequencies for the words used. What now is the achievable rate region?

### III. SEPARATE MESSAGES WITH PROTECTION

**Privacy**

This is perhaps the most basic of our new models. Most of our results concern the case of two users. Since the extensions of our definitions to any number of users are straightforward and so are some of the results, we give the definitions now for two users in order not to be burdened with a heavy formalism. When needed, the reader will be able to add the necessary indexes. A family \( \mathcal{P} = (C_{ij} : 1 \leq i \leq M_1, 1 \leq j \leq M_2) \) of disjoint subsets of \( \mathcal{X}^n \) is an \((n, M_1, M_2, d_1, d_2) \) privacy code for the users \( U_1 \) and \( U_2 \) with cost functions \( \varphi_1 \) and \( \varphi_2 \) if, for all pairs \((i, j)\) and \((i', j')\) [resp., \((i, j')\)] for every \( x^n \in C_{ij} \), \( y^n \in C_{ij'} \) (resp., \( C_{ij'} \)) exists with
\[
\left( \frac{1}{n} \sum_{i=1}^n \varphi_1^*(x^n, y^n) \right) \leq d_1 \quad \text{resp.,} \quad \left( \frac{1}{n} \sum_{i=1}^n \varphi_2^*(x^n, y^n) \right) \leq d_2.
\]

The pair \((R_1, R_2)\) of nonnegative numbers is called an achievable pair of rates if, for every \( \eta \) there are privacy codes of rates \( r_i = (1/n) \log M_i > R_i - \eta \) \((i = 1, 2)\) for all large \( n \). \( \mathcal{P} = (d_1, d_2) = (d_1 + d_2, \varphi_1, \varphi_2) \) is the set of all achievable pairs of rates.

Clearly, this code can be used by any one of the users to store a new message without distorting the message stored by the other user. In the sequel, we assume that
\[
\varphi_1 \equiv \varphi_2 \equiv \varphi
\]
and we let \( M(n, d) \) stand for the maximal length of an \((n, d)\) WEM code with cost function \( \varphi \).

We formulate our first observation.

**Lemma 1:** When the cost function \( \varphi \) satisfies the triangle inequality, then for any \((n, M_1, M_2, d_1, d_2) \) privacy code,
\[
M_1 M_2 \leq M(n, d_1 + d_2).
\]

Thus, \( \mathcal{R}(d_1, d_2) \subseteq \{(R_1, R_2) : 0 \leq R_i, R_1 + R_2 \leq R(d_1 + d_2)\} \).

**Proof:** The privacy code for two users also can be viewed as a one-user code with \( M_1 M_2 \) messages. For any pairs \((i, j)\) and \((i', j')\), we can make the transitions from \((i, j)\) to \((i', j')\) and then to \((i', j')\), for instance. Therefore, by the triangle inequality, for any \( x^n \in C_{ij} \), \( y^n \in C_{ij'} \) and \( z^n \in C_{ij'} \) can be found with
\[
\varphi(x^n, z^n) \leq \varphi(x^n, y^n) + \varphi(y^n, z^n).
\]

The per-letter cost is at most \( d_1 + d_2 \).

**Remark:** The cost function of the symmetric WUM does not satisfy the triangle inequality. Of course, the Hamming distance does.

Now, vice versa, we can also view an \((n, M_1, M_2, d) \) one-user code as an \((n, M_1, M_2, d) \) privacy code by choosing \( \{(1, \ldots, M_1) \times \{(1, \ldots, M_2) \} \) as the index set of the clouds. We express this in terms of rates:
\[
\mathcal{R}(d, d) \subseteq \{(R_1, R_2) : 0 \leq R_1, R_1 + R_2 \leq R(d)\}.
\]

A third simple fact is that, by space sharing, a single-user \((n_1, M_1, \tau_1) \) code \( (C) \) and a single-user \((n_2, M_2, \tau_2) \) code \( (D) \) can be combined to an \((n_1 + n_2, M_1, M_2, n_1 \tau_1 / (n_1 + n_2), n_2 \tau_2 / (n_1 + n_2)) \) privacy code \( (C \times D) \). We rewrite
It is a collection \( \{ C_i : 1 \leq i \leq M \} \) of disjoint subsets of \( S^n = \{0,1\}^n \) with the following properties:

For all \( i \) and \( j \), for any \( x^a \in C_i \), \( y^b \in C_j \) exists with

\[
x^a \wedge y^b = (x_1 \wedge y_1, \ldots, x_n \wedge y_n) = (0, \ldots, 0).
\]

We remind the reader again about the passage before Theorem K in Section IV.

From here, the definition of a privacy code \((n,M,N)\) for the WUM reads as follows:

It is a collection \( \{ C_{ij} : 1 \leq i \leq M, 1 \leq j \leq N \} \) of disjoint subsets of \( S^n \), called clouds, with the properties that for all \((i,j) \) and \((i',j') \) [resp., \((i,j) \) and \((i',j') \)] for every \( x^a \in C_{ij} \), \( y^b \in C_{ij'} \) [resp., \( C_{ij} \) and \( C_{ij'} \)] exists so that (4.3) holds.

We denote the set of achievable pairs of rates for these codes by \( \mathcal{R}_{wum}^2 \).

Throughout this paper, we use the "sequence" and "subset" notation interchangeably. In "subset" notation, the \( C_i \)'s are disjoint families of subsets of \( \{1,2,\ldots,n\} \).

**Theorem 3:** \( \mathcal{R}_{wum} \supset \{ (R_1, R_2) : 0 \leq R_1 \leq \frac{1}{4} \text{ for } i = 1, 2; R_1 + R_2 \leq h(\frac{1}{4}) = 0.813 \ldots \} \).

**Remark:** Since \( \mathcal{R}_{wum} = 0.694 \ldots \), the new region exceeds \( \mathcal{R}_5 \) not everywhere.

Of course, the convex hull of the union of both regions is again achievable.

**Proof:** It may be helpful for grasping the idea of the following proof to first take a look at the code construction in Section VIII. We describe our random selection of the clouds. They will have two elements. It suffices to consider even \( n \). Set \( n = 2k \) and \( \mathcal{A} = \{1,2,\ldots,n\} \). Choose independently according to the uniform distribution on \( \binom{k}{n} \) \( M+N \) sets. Call the first \( M \) sets \( S_1, \ldots, S_M \) and the last \( N \) sets \( T_1, \ldots, T_N \). Define, now, for any set \( A \subset \mathcal{A}, \)

\[
A^t = A \quad \text{and} \quad A^c = A^c
\]

where "c" denotes complementation in \( \mathcal{A} \). With this convention, we introduce the sets

\[
D_{ij}^a = S_i \cup T_j^b \quad \text{for } \alpha, \beta \in \{0,1\}.
\]

Clearly, \( \{|D_{ij}^a| = |D_{ij}^{00}|, |D_{ij}^{01}| = |D_{ij}^{01}| \). If, now, \( |D_{ij}| \geq \frac{1}{4}n \), then define

\[
D_{ij}^{11} = D_{ij}^{11} \quad \text{and} \quad D_{ij}^{21} = D_{ij}^{00}.
\]

Otherwise, we have \( |D_{ij}^{10}| \geq \frac{1}{4}n \), and we define

\[
D_{ij}^{10} = D_{ij}^{10} \quad \text{and} \quad D_{ij}^{21} = D_{ij}^{11}.
\]

For \( C \subset \mathcal{A}, \frac{1}{2}n \geq |C| \geq \frac{1}{4}n \), define

\[
L(C) = \{ (i,j) \in [1,\ldots,M] \times [1,\ldots,N] : \}

\[
D_{ij}^{(1)} = C \quad \text{or} \quad D_{ij}^{(2)} = C \}.
\]

Clearly, by this definition, for all \( (i,j) \in [1,\ldots,M] \times [1,\ldots,N], \)

\[
(i,j) \in L(D_{ij}^{(1)}) \cap L(D_{ij}^{(2)}).
\]
If, now,
\[ L(D_{ij}^{(1)}) = L(D_{ij}^{(2)}) = \{i, j\}, \]
then define
\[ C_{ij} = \{D_{ij}^{(1)}, D_{ij}^{(2)}\}. \tag{4.11} \]
Let us pause a moment and realize that, by our definitions, in case (4.11) holds for all \((i, j) \in \{1, \ldots, M\} \times \{1, \ldots, N\}, \) then these \(C_{ij}\)'s define a privacy WUM code as introduced in (4.4). However, this need not be the case or, in other words, for some \(C\), we may have \(|L(C)| > 1\). For some \(C\), called regular, we can remove this ambiguity, and for other \(C\), called irregular, we cannot. If irregular \(C\) are there, then the chosen collection \(\{S_1, \ldots, S_M; T_1, \ldots, T_N\} \) is just bad.

We now describe the regular \(C\)’s. For them, we can remove ambiguity by using suitable neighbors. Define
\[ \mathcal{L}(C) = \{C': |C' \Delta C| = 1, C' \subseteq C, \exists C'' \neq C' \text{ with } |C'' \Delta C'| \subseteq \{0, 1\}, C'' \supseteq C' \text{ and } L(C'') \neq \emptyset \}. \tag{4.12} \]
If, now, \(|L(C)| \leq |\mathcal{L}(C) \cup \{C\}| = |\mathcal{L}(C)| + 1, \) then choose any injective map \(\Psi_C: L(C) \rightarrow \mathcal{L}(C) \cup \{C\} \). Notice that the members of \(\mathcal{L}(C)\) have the following properties:
(a) they are contained in \(C\), but are different from \(C\) by one element exactly,     
(b) the do not equal any \(D_{ij}^{(1)}\) or \(D_{ij}^{(2)}, \)
(c) the supersets different from \(C\), which differ by exactly one element from it, also do not equal any \(D_{ij}^{(1)}\) or \(D_{ij}^{(2)}\).

Since \(C\) occurs \(|L(C)|\) times, we can resolve this conflict by using it once and replacing it otherwise by the members of \(\mathcal{L}(C)\). If both \(D_{ij}^{(1)}\) and \(D_{ij}^{(2)}\) are regular, then we can therefore define
\[ C_{ij} = \{\Psi_{D_{ij}^{(1)}}(i, j), \Psi_{D_{ij}^{(2)}}(i, j)\}. \tag{4.13} \]
The privacy WUM code is thus well defined if there are no irregular \(C\)’s. We now estimate \(\text{Prob}(\text{there are no irregular } C \in \mathcal{M})\) from below. A key role in this analysis is played by the condition
\[ \max \left( M2^{-\frac{1}{2n}}, N2^{-\frac{1}{2n}}, MN2^{-2n} \right) \left( \frac{n - |C|}{\left| C \right|} \right) < 2^{-e_n}. \tag{\star} \]

Claim 1: Prob \(|L(C)| \geq \varepsilon n\) \(\leq 2^{-e_n}n^{3/2}\) for some constant \(\varepsilon > 0\) if (\(\star\)) holds.

Let us write \(L(C) = \{(i, j), \ldots, (i, j)\}\), and let us consider the bipartite graph \((\{1, 2, \ldots, M\}, \{1, 2, \ldots, N\}, L(C))\). Recall König’s Theorem, which says that the maximal number \(\eta\) of independent edges in a bipartite graph equals the minimum number \(\mu + \nu\) of points covering all edges, that is, \(\eta = \mu + \nu.\)  \tag{4.14} 

The total number of edges is \(l.\) If, now, \(\eta < \sqrt{l}\), then by (4.14),
\[ \mu + \nu < \sqrt{l} \quad \text{or} \quad \frac{l}{\mu + \nu} \geq \sqrt{l}. \]
Hence, either there is a vertex in \(\{1, 2, \ldots, M\}\) covering at least \(\sqrt{l}\) edges, or there are \(\sqrt{l}\) independent edges, or there is a vertex in \(\{1, \ldots, N\}\) covering at least \(\sqrt{l}\) edges. If, now, \(l \geq \varepsilon n\), then choose
\[ s = \lceil \sqrt{\varepsilon n} \rceil. \tag{4.15} \]
Consequently, either we have event \(E_1\): for some \(i\) and \(j^{(1)}, \ldots, j^{(s)}\)
\[ \{(i, j^{(t)}): 1 \leq t \leq s\} \subseteq L(C) \]
or we have event \(E_2\): there are sets of indexes \(I = \{i^{(1)}, \ldots, i^{(s)}\}, J = \{j^{(1)}, \ldots, j^{(s)}\}\) with \((i^{(t)}, j^{(t)}): 1 \leq t \leq s\) \(\subseteq L(C),\) or we have event \(E_3\): for some \(j\) and \(i^{(1)}, \ldots, i^{(s)}\)
\[ \{(i^{(t)}, j): 1 \leq t \leq s\} \subseteq L(C). \]
Therefore, we have
\[ \text{Prob} \left( |L(C)| \geq \varepsilon n \right) \leq \sum_{i=1}^{3} \text{Prob} \left( E_i \right). \tag{4.16} \]
One readily calculates that
\[ \text{Prob}(E_1) \leq \left( \frac{M}{1} \right) \left( \frac{N}{s} \right) \left( 2^{-n + 0 \log n} \left( \frac{n}{|C|} \right) \right)^s, \]
\[ \text{Prob}(E_2) \leq \left( \frac{N}{1} \right) \left( \frac{M}{s} \right) \left( 2^{-n + 0 \log n} \left( \frac{n}{|C|} \right) \right)^s, \]
\[ \text{Prob}(E_3) \leq \left( \frac{MN}{s} \right) \left( 2^{-2n + 0 \log n} \left( \frac{n - |C|}{|C|} \left( \frac{n}{|C|} \right) \right)^s. \]
This, together with (\(\star\)), implies
\[ \text{Prob} \left( |L(C)| \geq \varepsilon n \right) \leq 3 \cdot 2^{-e_n + s + 0 \log n} \leq 2^{-e_n}. \]

Claim 2: Prob \(|\mathcal{L}(C) \leq \varepsilon n| \leq 2^{-e_n}n^{3/2}\) for some constant \(\delta > 0\) if (\(\star\)) holds.

If \(|\mathcal{L}(C) \leq \varepsilon n,\) then there are at least \(\frac{1}{2n} - \varepsilon n \geq \frac{1}{2n} - \varepsilon n \leq C^* \subseteq C^* \subseteq C^* \subseteq C \neq \emptyset \) \(\subseteq C \neq \emptyset \). Choose, now, \(s = \lceil \sqrt{\varepsilon n} \rceil\)

Considering \(U(C) = \{C^*: |C^* \Delta C| \leq 2, |L(C^*)| \geq 1\},\) we can distinguish again three events.

Event \(F_1\): For some \(i\) and \(j^{(1)}, \ldots, j^{(s)}\)
\[ \{(i, j^{(t)}): 1 \leq t \leq s\} \subseteq L(C^*), \quad C^{t} \subseteq C \quad \text{for } 1 \leq t \leq s. \]

Event \(F_2\): For some \(j\) and \(i^{(1)}, \ldots, i^{(s)}\)
\[ \{(i^{(t)}, j): 1 \leq t \leq s\} \subseteq L(C^*), \quad C^{t} \subseteq U(C) \quad \text{for } 1 \leq t \leq s. \]

Event \(F_3\): There are sets of indexes \(I = \{i^{(1)}, \ldots, i^{(s)}\}, J = \{j^{(1)}, \ldots, j^{(s)}\}\) such that
\[ \{(i^{(t)}, j^{(t)}): 1 \leq t \leq s\} \subseteq L(C^*), \quad C^{t} \subseteq U(C) \quad \text{for } 1 \leq t \leq s. \]
We have, therefore,
\[ \text{Prob}(\mathcal{L}(C) \leq \epsilon n) \leq \sum_{i=1}^{M} \text{Prob}(F_i). \quad (4.17) \]

One calculates that, under (*),
\[ \text{Prob}(F_1) \leq \left( \frac{0(n^2)}{\frac{N}{n}} \right) \left( \frac{M}{1} \right) \left( \frac{1}{\frac{n}{5}} \right) \left( \frac{2^{\frac{n}{5} + 1(\log n)}}{\text{Prob}(F_1) \leq 2^{\frac{\epsilon}{2}} \cdot n^{\frac{3}{2}}(3) \cdot 2^{\frac{1}{2}n} \pm 2} \right). \]

Similarly, \( \text{Prob}(F_2) \leq 2^{\frac{\epsilon}{2} n^{\frac{3}{2}}(3)}, \) and finally,
\[ \text{Prob}(F_3) \leq \left( \frac{0(n^2)}{\frac{N}{n}} \right) \left( \frac{M N}{\frac{n}{5}} \right) \left( \frac{2^{\frac{n}{5} + 1(\log n)}}{\text{Prob}(F_3) \leq 2^{\frac{\epsilon}{2}} \cdot n^{\frac{3}{2}}(3) \cdot 2^{\frac{1}{2}n} \pm 2 \pm 2} \right). \]

and thus Claim 2.

The two claims imply \( \mathbb{E}(\mathcal{C}: |L(C)| \geq \epsilon n) + \mathbb{E}(\mathcal{C}: |\mathcal{L}(C)| \leq \epsilon n) \leq 2^{\frac{\epsilon}{2} n^{\frac{3}{2}}(3) + 2^{\frac{1}{2}n}} \leq 2^{\frac{\epsilon}{2} n^{\frac{3}{2}}(3) + 2^{\frac{1}{2}n}}, \) and therefore the expected values tend to 0 as \( n \) tends to infinity. In particular, it becomes smaller than 1, and thus for large \( n \) with positive probability for all \( C \in \mathcal{L}(C) \neq \epsilon n \neq |L(C)| \) and all \( C \) must be regular. Hence, a privacy code exists if (*) holds. Now, for \( (R_i, R_j) \) with \( R_i \geq \frac{1}{2} - 2 \epsilon \) and \( R_1 + R_2 \leq 2 - \epsilon - (1/\log) \max_{\mathcal{C} \in (1/4)n} \left\{ \begin{array}{ll} n - |C| & \frac{1}{2}n \\ |C| & \frac{1}{2}n \end{array} \right\} \),

(*) is valid.

Observe next that
\[ \max_{|C| \geq n} \left( \frac{n - |C|}{\frac{1}{2}n} \right) \left( \frac{n}{|C|} \right) = \frac{1}{2}n \left( \frac{1}{2}n \right) = 2 - h(\frac{1}{2}). \]

and that \( \lim_{n \to \infty} (1/n) \log \left( \frac{1}{2}n \right) \left( \frac{1}{2}n \right) = 2 - h(\frac{1}{2}). \) The proof is complete.

Originally, WUM codes dealt with a situation arising in laser technology, where during one updating, only 0's or only 1's can be printed into the cells of the memory. In particular, this has led to the concept of an alternating WUM code:

A family \( \mathcal{C} = \{C_1, \cdots, C_M\} \) of disjoint subsets of \( \{0, 1\}^n \) is an alternating WUM code if
(i) \( C_i = T_{i0} \cup T_{ij} \) for \( 1 \leq i \leq M \),
(ii) for all \( (i, j) \) and any \( x^* \in T_{i0} \) (resp., \( T_{ij} \)) there is a \( y^* \in T_{ij} \) (resp., \( T_{i0} \)) with \( y^* \geq x^* \) (resp., \( y^* \leq x^* \)).

An alternating WUM code is called symmetric if, for \( i = 1, \cdots, M \),
\[ T_{i1} = T_{i0} = (\bar{x}^* : x^* \in T_{i0}). \]

Here, \( \bar{x}^* \) is obtained from \( x^* \) by exchanging 0's and 1's.

Obviously, a symmetric code is specified solely by the collection of subsets \( \{T_{i0}\} \). \( T_{i0} \)'s are used when we write 1's and \( T_{i1} \)'s are used when we write 0's. We can always use one additional bit of information to indicate which letter we are writing. In this sense, it does not matter whether \( T_{i0} \) and \( T_{i1} \) intersect or not. We need only the disjointness among \( T_{i0} \)'s and the disjointness among \( T_{i1} \)'s. Now, for \( x^* \in T_{i0} \) and \( y^* \in T_{j0} \) with \( x^* \leq y^* \), we have \( \bar{y}^* \in T_{j0} \) and \( \bar{x}^* \wedge \bar{y}^* = (0, 0, \cdots, 0) \). Therefore, we can use the alternative definition for an alternating symmetric WUM code given in (1.9) or (4.3). Since it is often easier to handle, we use in this paper the definition (1.9), bearing in mind that we can always easily construct a code in the sense of the former definition from a code in our new sense.

If \( C_i \cap \bar{C}_j = \emptyset \) for all \( i \) and \( j \), which is the case for the codes of Willems, Vinck, and Simonyi, we do not even need the additional bit of information.

We now present the positive answer to our conjecture that (3.7) also holds for WUM.

Theorem K [11]: We can achieve with a K user privacy WUM code the rate vector \( \left( \frac{1}{K+1}, \cdots, \frac{1}{K+1} \right) \), and thus total rate \( \frac{1}{K+1} \).

Proof: We describe in the language of our codes Koschinck's construction [11] of an \( (n, M_1, \cdots, M_K) \) code with
\[ n = (K + 1)s + 2 \cdot \lceil \log_2 (K + 1) \rceil \]
and
\[ M_i = 2^i \quad \text{for } i = 1, 2, \cdots, K. \]

Divide the memory of size \( n \) into \( K + 1 \) parts \( A_1, \cdots, A_{K+1} \), each of size \( s \), and one part \( B \) of size \( 2 \cdot \lceil \log_2 (K + 1) \rceil \). At any time, the messages of the K users are represented by \( K + 1 \) parts \( A_i \) of the memory. The remaining \( A \) block contains no information; it has only cells with 0 entries. In the \( B \) block, with a single-user WUM it is stored, which \( A \) block is free and which \( A \) block contains the message of which user (note that \( 2 \cdot \lceil \log_2 (K + 1) \rceil \) bits are sufficient for this).

Updatings are performed as follows. If user \( U_i \) wants to update his message stored by the memory, he first reads the content of block \( B \) and finds which \( A \) block contains his currently stored message—say, this block \( A_{r} \), and which one is free. After storing his new \( s \) bits in the free block and printing 0's only in the block \( A_{r} \), he stores in block \( B \) again which \( A \) block relates to which user. The updating is complete.

Finally, by (4.18) and (4.19) for fixed \( K \),
\[ \lim_{n \to \infty} \log M_i = \frac{1}{K + 1}. \]
V. A Hierarchy Code for the WEM

We now come to two users, $U_1$ and $U_2$, who both store messages in the memory, but only $U_1$ enjoys protection of his messages by $U_2$. To simplify matters, we assume that both users have the same cost function $\varphi$.

An $(n, M_1, M_2, d_1, d_2)$ hierarchy code is a collection $\{C_i; 1 \leq i \leq M_1, 1 \leq j \leq M_2\}$ of disjoint subsets of $\mathcal{Z}$ with the following properties:

(a) the collection $\{C_i; 1 \leq i \leq M_1\}$, where $C_i = \bigcup_{j=1}^{M_2} C_{ij}$, is an $(n, M_2, d_1)$ WEM code for the single-user $U_1$;

(b) for every $i, 1 \leq i \leq M_1$, the collection $\{C_{ij}; 1 \leq j \leq M_2\}$ is an $(n, M_2, d_2)$ WEM code for the user $U_2$.

At any time instant, any one of the users can operate on the memory. User $U_1$ updates his messages via his code. The message stored by $U_2$ is of no concern for him. On the other hand, user $U_2$ has to protect the message, say $i$, of user 1. This means he can only make changes within $C_i$, where he stores his new messages by the rules of the local code $\{C_{ij}; 1 \leq j \leq M_2\}$. Let us denote the rate regions by $\mathcal{R}^*(d_1, d_2)$. Obviously, $\mathcal{R}^*(d_1, d_2) \supseteq \mathcal{R}(d_1, d_2)$, and we derive the following from Lemma 2.

Lemma 2*: If $\varphi$ is not degenerate, that is, $\varphi(x, x) = 0$ for $x \in \mathcal{Z}$, $\mathcal{R}^*(d_1, d_2) \supseteq \{(R_1, R_2); 0 \leq R_1 \leq \lambda, R(d_1/\lambda), 0 < \lambda, \text{ for } i = 1, 2 \text{ and } \sum_{j=1}^{M_2} \lambda^j = 1\}$.

However, perhaps surprisingly, the upper bound of Lemma 1 also extends.

Lemma 1*: When the cost function $\varphi$ satisfies the triangle inequality, then for any $(n, M_1, M_2, d_1, d_2)$ hierarchy code

$M_1 M_2 \leq M(n, d_1 + d_2)$.

Thus, $\mathcal{R}^*(d_1, d_2) \subset \{(R_1, R_2); 0 \leq R_1, R_1 \leq R(d_1 + d_2)\}$.

Proof: The hierarchy code also can be viewed as a one-user code with $M_1 M_2$ messages. The verification differs slightly from the one for the privacy code. For any pairs $(i, j)$ and $(i', j')$, we can make a transition from $C_i \subset C_{i'}$ to $C_j$ and thus to $C_{j'}$ for some $j'$ (possibly different from $j$). But now, we still can make the transition via the local code to $C_{j'}$.

The argument can be applied to any number of users with a linear hierarchy structure, that is, messages of users with a lower number are to be protected by users with higher numbers.

Theorem 2*: In case $K$ users have a linear hierarchy structure and the same cost function satisfying the triangle inequality, then the optimal common rate for per-letter cost $d$ equals $(1/K)R(Kd)$.

VI. NO SIDE INFORMATION AT THE ENCODER AND THE DECODER, BUT THE ENCODER CAN USE SPACE BARS

We consider now the case $(E_-, D_-)$ described in the Introduction. All users have the same cost function $\varphi$. We denote by $R_-(d)$ [resp., $\bar{R}_-(d)$] the optimal rate for a single user and threshold $d$, in case the maximal (resp., average) costs are considered. Here, $C = \{c_i; 1 \leq i \leq M\}$ with $c_i \in \mathcal{Z}$ is an $(n, M, d)$ WEM code if

$$D_{oc}(C) = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \varphi(c_i, c_j) \leq nd,$$

and it is an $(n, M, d)$ code if

$$\varphi(c_i, c_j) \leq nd$$

for $i = 1, 2, \cdots, M$.

In recent work [16] and as a result of an entirely different direction of research, $\bar{R}_-(d)$ was characterized for all cost functions $\varphi: \mathcal{Z} \times \mathcal{Z} \rightarrow R$ as follows.

Consider a constant $\nu, 0 \leq \nu \leq 1$ and two probability distributions $P$ and $P'$ on $\mathcal{Z}$, $(x_1, x_2, \cdots, x_N) \in \mathcal{Z}$ is said to be of the mixed type $(\nu P, (1 - \nu) P')$ if $(x_1, \cdots, x_N)$ is of type $P$ and $(x_{[N]} + 1, \cdots, x_N)$ is of type $P'$ (that is, has relative frequencies specified by $P'$).

The sets $T_\nu(\nu, P, P')$

$= \{x^N \in \mathcal{Z}; x^N \text{ is of mixed type } (\nu P, (1 - \nu) P')\}$

have a rate

$$R(\nu, P, P') = \nu H(P) + (1 - \nu) H(P')$$

and an average cost

$$D_{oc}(\nu, P, P') = \nu \sum_{x \in \mathcal{Z}} P(x)P(y)\varphi(x, y) + (1 - \nu) \sum_{x \in \mathcal{Z}} P'(x)P(y)\varphi(x, y) + o(n).$$

Theorem AA: $\bar{R}_-(d)$ is max $\{R(\nu, P, P'): (\nu, P, P') \text{ with } \sum_{x, y \in \mathcal{Z}} P(x)P(y) + (1 - \nu)P'(x)P(y) \cdot \varphi(x, y) \leq d\}$.

Remarks: We stated in [2, eq. (9.1)] a simpler formula for $\bar{R}_-(d)$. It holds for special cost functions such as the Hamming distance in the binary case, but not in general. $R_-(d)$ is known in some metric cases [7], for instance in the binary Hamming case.

Recently, a solution of the diadic problem in the average has also been found for several distance and cross-distance constraints [20], that is, a multiuser model.

Inspection of Lemmas 1, 2 and Lemmas 1*, 2* shows that they can be extended much further. One observation is that, for them to hold, no formula for the underlying single user rate function is needed, and another observation is that space sharing must be possible. In case $E_+$, this is obviously always possible because the encoder writes in one part, say $B_1$, of the memory, and leaves the imprints in the other part, say $B_2$, invariant. He can do the latter at no cost if $\varphi$ is not degenerate. In case $E_-$, however, he does not know the imprints, especially not in $B_2$. Therefore, he does not know what to write in order to leave them invariant. If we allow the encoder to use space bars, then this simply means that we allow him to print nothing in certain cells such that their content, unknown to him, does not change. We symbolize this option by $E_0$. Clearly, in this case, space sharing is possible. The single-user rate functions are denoted by $R_-(d)$ and $\bar{R}_-(d)$.

In case $E_-$, there is no way in which the users keep their privacy. The hierarchy constraint can be met if the "slave" is in case $E_+$. (The boss cannot read, but the slave
has to be able to read in order to respect the wishes of the
boss.) This leads to a funny code concept.

We concentrate now on the side information \((E_\circ, D_\circ)\).
In case of privacy, we denote the rate regions by \(\mathcal{R}_\circ(d_1, d_2)\) and \(\mathcal{R}_\circ^*(d_1, d_2)\). We add a star if we have
hierarchy. It saves notation if we introduce the region
\[
\mathcal{R}(R) = \{ (R_1, R_2) : 0 \leq R_1, R_2 \leq R \}.
\]

**Lemma 3:** If \(\varphi\) satisfies the triangle inequality, then
\[
\begin{align*}
(\text{a}) & \quad \mathcal{R}_\circ(d_1, d_2) \subset \mathcal{R}_\circ^*(d_1, d_2) \\
(\text{b}) & \quad \mathcal{R}_\circ^*(d_1, d_2) \subset \mathcal{R}(R)(d_1 + d_2) \\
(\text{c}) & \quad \mathcal{R}_\circ^*(d_1, d_2) \subset \mathcal{R}(R)(d_1 + d_2) \\
(\text{d}) & \quad \mathcal{R}_\circ^*(d_1, d_2) \subset \mathcal{R}(R)(d_1 + d_2).
\end{align*}
\]

**Proof:** Since \(\mathcal{R}_\circ^*(d_1, d_2) \subset \mathcal{R}_\circ^*(d_1, d_2)\) and \(\mathcal{R}_\circ^*(d_1, d_2) \subset \mathcal{R}_\circ^*(d_1, d_2)\), it suffices to establish the last
two inclusions.
(c) From the point of view of the decoder, there is a
collection \(\{C_{ij} : 1 \leq i \leq M_1, 1 \leq j \leq M_2\}\) of disjoint sub-
sets of \(\mathcal{R}\). Elements of \(C_{ij}\) store message \(i\) for \(U_j\)
and message \(j\) for \(U_i\). The elements of \(C_j = \bigcup_{j<i} C_{ij}\) store
message \(i\) for \(U_j\). If the code has parameters
\((n, M_1, M_2, d_1, d_2)\), then we can build a single-user code
with parameters \((n, M_1 \cdot M_2, d_1 + d_2)\). To see this, let us
suppose that \(c_{ij}\) is the imprim not known to the users. We
want to store \((i', j')\). First, let \(U_i\) store \(i\) according to the
rules of the code. This results in some imprim \(c_{i'j}\). Now,
let \(U_i\) store \(j\). This results in some imprim \(c_{ij'}\). Further-
more,
\[
\varphi(c_{ij}, c_{i'j'}) \leq \varphi(c_{ij}, c_{ij'}) + \varphi(c_{i'j'}, c_{ij'}) \leq nd_1 + nd_2.
\]

\[\text{(6.4)}\]

(d) If the average per-letter costs are \(d_1\) and \(d_2\), then
\[
\frac{1}{n} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \sum_{i'=1}^{M_1} \sum_{j'=1}^{M_2} \varphi(c_{ij}, c_{i'j'}) \leq nd_1 + nd_2.
\]

Let \(E_\circ\) be any side information including \(E_\circ\), that is, the
side information which makes space sharing possible, and
let us indicate this information with a circle in the nota-
tions for regions.

**Lemma 4:** If \(\varphi\) is not degenerate, then
\[
\mathcal{R}_\circ(d_1, d_2) \supset \{ (R_1, R_2) : 0 \leq R_i \leq \lambda_i R_\circ \left( \frac{d_i}{\lambda_i} \right), 0 < \lambda_i \leq 1, \Sigma \lambda_i = 1 \},
\]
\[
\mathcal{R}_\circ^*(d_1, d_2) \supset \{ (R_1, R_2) : 0 \leq R_i \leq \lambda_i R_\circ \left( \frac{d_i}{\lambda_i} \right), 0 < \lambda_i \leq 1, \Sigma \lambda_i = 1 \}.
\]

The same relations hold for average costs.

**Proof:** As usual, by time sharing.

It is now clear how to produce from these two lemmas
results like Theorems 2, 2'. We address one striking case
here.

**Theorem 4:** For \(K\) users with complete privacy con-
straints and a cost function, which is a metric, we have in
case \((E_\circ, D_\circ)\) as the optimal common rate for per-letter
cost \(d\)
\[
\frac{1}{K} R_\circ(Kd) \left( \lim_{K \to \infty} R_\circ(Kd) = \log |\mathcal{R}| \right).
\]

**Problem 4:** Consider coding problems for common mes-
gages (see Section II) in case \((E_\circ, D_\circ)\) and average costs
constraint.

**Problem 5:** How does \(R_\circ(d)\) relate to \(\mathcal{R}_\circ(d)\)? Are there
\(\varphi\) for which these quantities are different?

**Problem 6:** Suppose in case \((E_\circ, D_\circ)\) the user is in-
formed whether an intended writing is possible, given the
cost constraint and the imprim (unknown to him). Are there
meaningful results?

**VII. IDENTIFICATION ON THE WEM**

We assume here familiarity with the notions and results of
[18].

The memory is now not used for the storage of mes-
gages, but for the identification of any one of \(M\) objects.

Now, just observe that in the case of the maximal per-
letter cost criterion, all storage problems (single- and
multiuser, privacy, hierarchy, common messages, various
kinds of side information) can all be viewed as error-free
transmission problems over a noiseless channel with
senders \(U_1, \ldots, U_k\) and the decoder as receiver.

Let us assume that the respective codes are specified.
To fix ideas, let us address the single-user case now. Let
its message set be \(\mathcal{M} = \{1, \ldots, M\}\). Then there are
\(N = 2^{h(e)}\) subsets \(\{P_j : 1 \leq j \leq N\}\) of cardinality \(M/2\)
with symmetric difference of cardinality at least \(eM\). Let \(D =
\{(D_j, \mathcal{Q}_j) : 1 \leq j \leq N\}\), where \(\mathcal{Q}_j\) is the uniform distribution
on \(D_j\), be the random encodings for the objects \(1, 2, \ldots, N\). We know from [18] that this gives a “good”
randomized identification code for the noiseless channel
with second-order rate \(1 \log(\log(2^{h(e)})) \sim 1/n \log M =
R(d) + o(n)\). We also know by the converse result of [18]
that this is the best possible. With every updating, these
randomized encodings are employed.

The same can be done with several users in any case
\(E_\circ\). Thus, the regions \(\mathcal{R}_\circ(d_1, d_2)\) are achievable as regions
for second-order rate pairs. Also, Lemma 3 (a), (b) and
Lemma 4 extend to second-order regions. We summarize
this.

**Theorem 5:** If \(\varphi\) is a metric, then the second-order
identification regions satisfy
\[
\{ (R_1, R_2) : 0 \leq R_i \leq \lambda_i R_\circ \left( \frac{d_i}{\lambda_i} \right), 0 < \lambda_i \leq 1, \Sigma \lambda_i = 1 \} 
\subset \mathcal{R}_\circ^2(d_1, d_2) \subset \mathcal{R}_\circ^2(d_1, d_2) \subset \mathcal{R}_\circ(d_1 + d_2).
\]

In particular, for \(\lambda_i = \frac{1}{2}, d_1 = d_2 = d(\frac{1}{2} R_\circ(2d), \frac{1}{2} R_\circ(2d))\)
is an optimal identification pair.
VIII. CONSTRUCTIONS OF MULTIUSER WUM CODES UNDER PRIVACY CONSTRAINTS

We now consider symmetric alternating WUM codes as defined in Section IV. For the history of this concept, the reader is also advised to consult the Appendix. Since throughout this section all users keep their privacy, we simply speak of WUM codes.

We begin with $(n, M, N)$, that is, two user WUM codes of blocklength $n$ and message sets of sizes $M$ and $N$.

We use both the subset and the sequence notation. It may be helpful to start with a simple example.

(6, 3, 4) Two-User WUM Code: The entries in the following table are the sets $C_{ij}$ ($1 \leq i \leq 4$, $1 \leq j \leq 3$).

<table>
<thead>
<tr>
<th>$j \backslash i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1, 2, 3}</td>
<td>{1, 2}</td>
<td>{1, 3}</td>
<td>{2, 3}</td>
</tr>
<tr>
<td></td>
<td>{4, 5, 6}</td>
<td>{4, 5}</td>
<td>{4, 6}</td>
<td>{5, 6}</td>
</tr>
<tr>
<td>2</td>
<td>{3}</td>
<td>{1, 2, 6}</td>
<td>{3, 5}</td>
<td>{1, 6}</td>
</tr>
<tr>
<td></td>
<td>{6}</td>
<td>{2, 4, 5}</td>
<td>{2, 6}</td>
<td>{3, 4}</td>
</tr>
<tr>
<td>3</td>
<td>{5}</td>
<td>{1}</td>
<td>{1, 3, 5}</td>
<td>{1, 5}</td>
</tr>
<tr>
<td></td>
<td>{2}</td>
<td>{4}</td>
<td>{2, 4, 6}</td>
<td>{2, 4}</td>
</tr>
</tbody>
</table>

The total rate is $\frac{1}{6} \log 3 \cdot 4 = 0.5975 \ldots$.

The idea used to construct this code is also useful for the construction of the two-user codes below, as well as for the proof of a random coding theorem for two-user WUM in Section IV. This basic idea can be described for the (6, 3, 4) code as follows. Suppose that the first user has four messages. For each message of the first user, we choose a subset of $\{1, \ldots, 6\}$ and denote it by $X_i$ ($1 \leq i \leq 4$). For each message of the second user, we also choose a subset of $\{1, \ldots, 6\}$ and denote it by $Y_j$ ($1 \leq j \leq 3$). We call these two collections of subsets the basis of the code.

Then we define either $C_{ij} = \{X_i \cap Y_j, X_i \setminus Y_j \}$ or $C_{ij} = \{X_i \setminus Y_j, X_i \cap Y_j \}$. Obviously, if all of the sets $C_{ij}$ so defined are disjoint then $|C_{ij}| = 1 \leq i \leq 4 \leq 1 \leq j \leq 3$ forms a two-user WUM code. For our (6, 3, 4) code, we use $X_1 = \{1, 2, 3\}$, $X_2 = \{1, 2, 6\}$, $X_3 = \{1, 3, 5\}$, $X_4 = \{2, 3, 4\}$ and $Y_1 = \{1, 2, 3\}$, $Y_2 = \{1, 2, 6\}$, $Y_3 = \{1, 3, 5\}$. When we construct the code, all we need to take care of is the right choice for $C_{ij}$ between the two possible methods for each pair $(i, j)$.

The construction of the next two codes is quite similar, even with respect to the choices of the $X_i$'s and the $Y_j$'s.

### (8, 6, 6) Two-User WUM Code: We take

$X_1 = \{1, 2, 3, 4, 5\}$, $Y_1 = \{1, 2, 3, 4, 5\}$, $Y_9 = \{1, 2, 5, 8, 9\}$,

$X_2 = \{1, 2, 3, 4, 10\}$, $Y_2 = \{1, 2, 3, 4, 10\}$, $Y_{10} = \{1, 3, 4, 7, 10\}$,

$X_3 = \{1, 2, 4, 5, 9\}$, $Y_3 = \{1, 2, 3, 5, 9\}$, $Y_{11} = \{1, 3, 5, 7, 9\}$,

$X_4 = \{1, 2, 4, 5, 8\}$, $Y_4 = \{1, 2, 4, 5, 8\}$, $Y_{12} = \{1, 4, 5, 7, 8\}$,

$X_5 = \{1, 3, 4, 5, 7\}$, $Y_5 = \{1, 3, 4, 5, 7\}$, $Y_{13} = \{2, 3, 4, 6, 10\}$,

$X_6 = \{2, 3, 4, 5, 6\}$, $Y_6 = \{2, 3, 4, 5, 6\}$, $Y_{14} = \{2, 3, 5, 6, 9\}$,

$X_7 = \{1, 2, 3, 9, 10\}$, $Y_7 = \{1, 2, 3, 9, 10\}$, $Y_{15} = \{2, 4, 5, 6, 8\}$,

$X_8 = \{1, 2, 4, 8, 10\}$, $Y_8 = \{1, 2, 4, 8, 10\}$, $Y_{16} = \{3, 4, 5, 6, 7\}$.

The code is defined through the following table. The entries of the table are either $\ast$ or $\ast \ast$, indicating which formula is used to form the set $C_{ij}$, except for $C_{62}$, where the entry is the set $\{\emptyset\}$.

### (10, 6, 16) Two-User WUM Code: The construction of this code is similar to the first two two-user codes. To choose the basis, we divide the ten positions into two equal parts, $\{1, 2, 3, 4, 5\}$ and $\{6, 7, 8, 9, 10\}$. For a subset of the first part, say $Z = \{1, 2, 5\}$, we denote $\{3, 4\}$ by $Z^c$ and $\{8, 9\}$ by $5 + Z^c$, and we define $Z = \{1, 2, 5, 8, 9\} = Z \cup (5 + Z^c)$. For this code, we take as $X_i$'s the six $Z$'s such that the cardinalities of the corresponding $Z$'s are greater than or equal to 4, and as $Y_i$'s the 16 $Z$'s such that the cardinalities of the corresponding $Z$'s are greater than or equal to 3.

<table>
<thead>
<tr>
<th>$j \backslash i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>2</td>
<td>$\ast \ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast \ast$</td>
<td>${\emptyset}$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>3</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>4</td>
<td>$\ast \ast$</td>
<td>$\ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>5</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>6</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast \ast$</td>
<td>$\ast$</td>
</tr>
</tbody>
</table>

The total rate is $\frac{1}{6} \log 6 \cdot 6 = 0.6463 \ldots$.
The code is given in the following table by showing which formula is used to form the sets $C_{ij}$.

<table>
<thead>
<tr>
<th>$j \setminus i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>2</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>3</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>4</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>5</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>6</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>7</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>8</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>9</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>10</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>11</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>12</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>13</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>14</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>15</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>16</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
</tbody>
</table>

For this code, the sum of the two rates is 0.6585\ldots. This is the best two-user WUM code we know.

For more than two users, it is convenient to write symmetric alternating WUM codes in the following way.

Let $\mathcal{A} = \{1, 2, \ldots, M\}$, $1 \leq k \leq k$ be finite sets and let $\mathcal{A} = \prod_{i=1}^{k} \mathcal{A}_i$. An $(n, M_1, \ldots, M_k)$ (symmetric alternating) $k$-user WUM code is a collection of sets $\mathcal{C} = \{C_x : x \in \mathcal{A}\}$ with $C_x \subset \{0, 1\}$ such that, for all $u \in \mathcal{A}$ and all $v \in \mathcal{A}$ with $d_H(u, v) \leq 1$ for all $x \in C_u$, $y \in C_v$, exists with $x \land y = 0$.

We consider the case $M_i = 2$ for $i = 1, \ldots, k$, and we address $[n, k]$ one-bit $k$-user WUM codes. Their total rate is obviously $k/n$.

5.3 One-Bit WUM Code:

$C_{000} = \{000000\}$, $C_{010} = \{011000, 00011\}$,
$C_{001} = \{010100, 00110\}$, $C_{011} = \{010010, 10010\}$,
$C_{011} = \{000100, 01001\}$, $C_{111} = \{100000, 01000\}$.

The rest of this section is devoted to a one-bit six-user WUM code. Its rate $\frac{1}{2}$ is surprisingly high for an alternating code.

7.6 One-Bit WUM Code: We first look at the point-line incidence matrix of a projective plane:

\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0
\end{pmatrix}
\]

Deleting the first column results in a $6 \times 7$ matrix:

\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0
\end{pmatrix}
\]

Denote the entries of this matrix by $z_{ij}$; $1 \leq i \leq 6$, $1 \leq j \leq 6$; the rows by $X_1, X_2, \ldots, X_6$; and the columns by $Y_1, Y_2, \ldots, Y_6$. For $x \in \{0, 1\}^6$, define

$\Phi(x) \triangleq y = (y_1, \ldots, y_6)$

where $y_i = 1$ iff $d_H(X_i, x) \leq 2$ and $y_i = 0$ iff $d_H(X_i, x) \geq 3$. Similarly, for $y \in \Omega_3^n = \{z \in \{0, 1\}^6 : |z| \leq 3\}$, define

$\Psi(y) \triangleq x = (x_1, \ldots, x_6)$

where $x_i = 1$ iff $d_H(Y_i, y) \leq 3, x_i = 0$ iff $d_H(Y_i, y) \geq 4$.

Theorem 6:

(i) $\Phi$: $\{0, 1\}^6 \rightarrow \Omega_3^n$ is bijective and $\Psi = \Phi^{-1}$.

(ii) $\mathcal{C} = \{(\Phi(x), \Phi(x)) : x \in \{0, 1\}^6\}$ is a one-bit six-user WUM code.

Ad (ii): We know from (i) that $\Phi$ is the encoding function and $\Psi$ is the decoding function. Both functions are quite simple and easy to implement. Now, the WUM property remains to be verified. It suffices to show that $d_H(x, x) \leq 1$ implies $\Phi(x) \land \Phi(x) = 0$. For this, notice that in case $d_H(x, x) \geq 5$ for any $i$, we have, by the triangle inequality, either $d_H(x, X_i) \geq 3$ or $d_H(X_i, X_i) \geq 3$. This means that

$(\Phi(x) \land \Phi(x)) = 0$ for $i = 1, 2, \ldots, n$

or $\Phi(x) \land \Phi(x) = 0$.

IX. MORE OPEN PROBLEMS

Problem 7 (Carrier Problem): Prescribe cost function $\Phi$, per-letter cost $d$, and rate $R$. Find $S \subset \mathcal{C}$ with minimal cardinality such that

$\forall s \in S : |S \cap S_{\delta}(s)| \geq \exp\{Rn\}$

where $S_{\delta}(s)$ is sphere of cost radius $\delta n$. In particular, we are interested in the Hamming case.

Problem 8: Recall from the Introduction the definition of an $(n, M, d)$ WEM code in case $(E_+, D_-)$, and for any code $\mathcal{C} = \{C_1, \ldots, C_M\}$, the definitions of $d_{\min}, \nu_{\min}$, and $R(d)$. We impose the following condition on $\mathcal{C}$.

$|C_i| \leq \exp\{\rho n\}$ for $i = 1, 2, \ldots, M$ (9.1)

and set $R(d, \rho) = \sup\{\nu_{\min} : d_{\min} \leq d, \mathcal{C} \text{ satisfies } (9.1)\}$. Determine this rate function. There is a relation to Problem 7.

Problem 9: We speak of an automatic privacy guarantee if the code $\{C_{ij} : 1 \leq i \leq M, 1 \leq j \leq N\}$ is such that updatings from $C_{ij}$ to $C_{ij'}$ in one step are impossible (for instance, for the WUM or other cost constraints) if $i \neq i'$ and $j \neq j'$. This means that the users can "move only in rows or columns," and keeping their interests in updating...
never can hurt the other. What are the achievable rates in the various models of side information, etc.?

APPENDIX

TWO SINGLE-USER WUM CODES

We begin with historical remarks.

Write-uni-directional memories (WUM's) have been introduced independently by Willems and Vinck [3] and Borden [4]. A WUM is a binary storage medium that is constrained the following way. During alternate updatings of its contents, the encoder may write either 0's in selected bit locations or 0's in selected bit locations, but is not permitted to write combinations of 0's and 1's. Such a constraint arises when the mechanism that chooses to write 0's or 1's operates much more slowly than that of accessing and scanning the memory.

Corresponding to the given constraint, a WUM code is defined as follows.

A family \( \mathcal{B} = \{C_1, \ldots, C_M\} \) of subsets of \( \{0,1\}^n \) is a WUM code of length \( n \) and size \( M \), if

(i) \( C_i \cap C_j = \emptyset \) for \( i \neq j \),

(ii) for all \( i \in \{1, \ldots, M\} \), all \( x \in C_i \), and all \( j \in \{1, \ldots, M\} \), there exists a vector \( y \in C_j \) such that \( x \leq y \) or \( y \leq x \), where we write \( x = (x_1, \ldots, x_n) \leq y = (y_1, \ldots, y_n) \) if, for all \( i = 1, \ldots, n \), \( x_i = 1 \) implies \( y_i = 1 \).

The sets \( C_i \) are called the WUM sets of the code \( \mathcal{B} \). All codewords of a set \( C_i \) represent the same message, say \( m_i \). We use the abbreviation \( (n, M) \) code for a WUM code of length \( n \) and size \( M \). The rate \( R \) of an \((n, M)\) code is

\[
R = \frac{1}{n} \log_2 M. \tag{A.1}
\]

Let \( M(n) \) be the maximum value of \( M \) such that there exists an \((n, M)\) code, and let \( R(n) \) be the corresponding rate. Borden [4] proved the following result.

Theorem:

(i) \( R(n) < \gamma := \log_2 \left( \frac{1 + \sqrt{5}}{2} \right) = 0.6942 \) if \( n \geq 5 \),

(ii) \( \lim_{n \to \infty} R(n) = \gamma. \tag{A.3} \)

In Borden's model, the encoder can choose whether he will operate in the 0-write state or in the 1-write state when he updates the information stored by the WUM. Willems and Vinck considered a slightly different model. They required that the encoder writes 0's and 1's alternately during successive updatings. The WUM code in the sense of Willems and Vinck has later been called alternating WUM code by Simonly [5]. Its definition is given in Section IV.

Let \( R_2(n) \) be the maximal rate of an alternating WUM code of length \( n \). It follows from Borden's proof of his theorem above that also \( \lim_{n \to \infty} R_2(n) = \gamma \). The subclass of alternating WUM codes is important because any alternating WUM code can be expanded with the same rate to arbitrarily large \( n \) simply by concatenation. Note that this is not true for general WUM codes.

Borden's proof of (A.3) is not constructive. The best explicit construction in [4] for arbitrarily large \( n \) is based on the following one. Let \( n \), the number of binary positions of the memory, be even, and assume that the memory is initially filled with 0's. Divide the WUM into two parts, \( A \) and \( B \), each of size \( n/2 \). When updating the content of the memory for the first time, write 1's in selected bit locations of part \( A \) and in all bit locations of part \( B \) of the memory; during the next updating, write 0's into all bit locations of part \( A \) and into selected bit locations of part \( B \), and so on. Thus, in every updating, \( n/2 \) bits of information can be stored in one of the two parts of the memory, and the other part is prepared for the next updating. This corresponds to a WUM code with the rate \( R_2 = \frac{1}{2} \).

Several WUM codes with higher rates are known. Willems and Vinck [3] gave the construction of a \((5,6)\) code with the rate \( \log_2 6/5 \approx 0.5170 \). Simonly [5] presented an \((11,58)\) code with the rate \( \log_2 58/11 \approx 0.5325 \). Zhang constructed the \((10,41)\) code and the \((15,307)\) code presented below which achieve the rates 0.5376 and 0.5508, respectively. The presently known highest rate is \( \log_2 7767 \) for Koschinke's \([11,17,767]\) code. All of these codes have the "alternating" property, and therefore can be used for arbitrarily large memories.

They are all symmetric in the sense defined in (4.3). To fix ideas, we begin with a simple symmetric alternating WUM code.

Example—(5, 6) WUM code (Willems and Vinck [3]):

\[
\begin{align*}
C_1 & = \{(0000, 01100, 00011), \quad C_2 = \{(01000, 10010, 00110), \quad C_3 = \{(00100, 10001, 01010), \quad C_4 = \{(00010, 10100, 01001), \quad C_5 = \{(00001, 11000, 00110), \quad C_6 = \{(00000).\end{align*}
\]

The rate of this code is \( \frac{1}{5} \log_2 6 \approx 0.5170 \). This code, as well as Simonly's code, have the property that for all \( i \in \{1, \ldots, M\} \), the words of the WUM sets \( C_i \) have disjoint supports. However, one can show that symmetric alternating WUM codes with this property cannot have rates arbitrarily near \( \gamma \). Actually, the best code with this property is Simonly's code having the rate 0.5325.

We now present two single-user WUM codes. One of them is a \((10,41)\) code; the other one is a \((15,307)\) code. We first introduce a new concept—WUM subsets—which will play a key role in their constructions.

Let \( \Omega_n^k = \{x \in \{0,1\}^n : \|x\| \leq k \} \)

where \( \|x\| \) is the weight of \( x \). A subset \( W \) of \( \Omega_n^k \) is called an \((n, k)\) WUM set if

\[
\forall x \in \Omega_n^k \exists y \in W \quad \text{such \ that \ } y \text{ and } x \text{ are disjoint.} \tag{A.5}
\]

Obviously, a set of disjoint \((n, k)\) WUM subsets of \( \Omega_n^k \) for a symmetric alternating WUM code. Our two codes presented here are both of this type.

First, the reader is asked to check that the following sets are \((10,3)\) WUM sets.

\[
\begin{align*}
A: & = \{(0000000000), \quad B: = \{(0000000010, 0000000011)\}, \quad C: = \{(0001110000, 0001100100, 0000000111)\};
1110000000, & = \{(0000000000, 0000000011)\},
1011000000, & = \{(0001110000, 0001100100, 0000000111)\};
0000000011, & = \{(0000000000, 0000000011)\}; \quad 0000000011.
\end{align*}
\]

Not only are these three sets \((10,3)\) WUM sets, but their permutations are as well. In fact, we have the following more general observation:

Let \( W \) be an \((n, k)\) WUM set and \( \pi \) a permutation on the set \( \{1, \ldots, n\} \); then \( \pi W \) is again an \((n, k)\) WUM set.

Now, the idea for our construction is clear. To construct our codes, we first choose a set of basic WUM sets, and then choose a set of permutations for each WUM set to make the results of
the permutations all disjoint. These WUM sets obtained from the permutations form a WUM code. An inspection of Borden’s proof of (A.2) shows that rates arbitrarily close to \( r \) can be obtained with these codes. The task of choosing WUM sets and permutations is usually very difficult. It can be done (as here) artistically or algorithmically (as in [11]).

\((10, 41)\) WUM Code: We use the following permutation in the construction of this code:

\[
s = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 3 & 4 & 5 & 1 & 7 & 8 & 9 & 10 & 6 \end{pmatrix}
\]

This permutation has the property that

\[
s^5 = I,
\]

and for any nonzero codeword \( x \) in \( \Omega_4^{10} \), \( sx \neq x \). To show how this permutation works on the words in \((0, 1)^{10}\), we present the following example.

Let

\[
x = 1101001010;
\]

then

\[
sx = 0110100101, \quad s^2x = 1011001010,
\]

\[
s^3x = 0101101001, \quad s^4x = 1010110100,
\]

and finally,

\[
s^5x = 1101001010 = x.
\]

For a set \( A \) of words, \( s'A \) is defined by \( s'A = \{ s'x : x \in A \} \).

The \((10, 41)\) WUM code consists of the set \((0000000000), 0\) and the following eight basic WUM subsets \( B_i \) of \( \Omega_4^{10} \) and all their permutations \( s'B_i, 0 \leq i \leq 4, 1 \leq j \leq 8 \). There are all together \( 1 + 5 \times 8 = 41 \) distinct subsets in this code. We leave to the reader the task to check that this code satisfies all of the requirements for a symmetric alternating WUM code. This can be done by computer.

\[
B_1: \quad 0000000000 \quad B_2: \quad 1010001000 \quad B_3: \quad 1000001100 \\
1100001000 \quad 1010000100 \quad 0100001100 \\
0000001100 \quad 0000000100 \quad 1100000000 \\
0001000100 \quad 0001001000 \quad 0011000000 \\
0010000011 \quad 0001000011 \quad 0000100010 \\

B_4: \quad 0010000000 \quad B_5: \quad 0001010000 \quad B_6: \quad 1010000010 \\
0010000000 \quad 1000011000 \quad 0100100000 \\
0000000000 \quad 0000000000 \quad 0000000000 \\
0110000000 \quad 0110000000 \quad 0110000000 \\

B_7: \quad 1000010100 \quad B_8: \quad 1100000000 \\
0010010100 \quad 0000001110 \\
0100000000 \quad 00000001001 \\
0010000000 \quad 0001010000 \\

(15, 307) WUM Code: This code uses up all of the words in \( \Omega_4^{15} \). It consists only of \((15, 4)\) WUM sets. To describe the construction of this code, we first introduce a few basic permuta-

\[
S = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 3 & 4 & 5 & 1 & 7 & 8 & 9 & 10 & 6 \end{pmatrix};
\]

\[
S_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 3 & 4 & 5 & 1 & 7 & 8 & 9 & 10 & 6 \end{pmatrix};
\]

\[
S_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix};
\]

\[
S_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix};
\]

\[
T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix};
\]

\[
R = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix};
\]

The code contains the set \((0000000000000000), 0\) and the following 12 basic WUM sets and their various permutations. The 12 basic sets are

\[
A_1: \quad 11000010000100000000 \quad A_2: \quad 11000110000000000000 \\
01000010000100000000 \quad 00001000100000000000 \\
10000001000000000000 \quad 00001000000000000000 \\
01000000100000000000 \quad 00001000000000000000 \\
01000000001000000000 \quad 00001000000000000000 \\
01000000000100000000 \quad 00001000000000000000 \\
01000000000010000000 \quad 00001000000000000000 \\
01000000000001000000 \quad 00001000000000000000 \\
01000000000000100000 \quad 00001000000000000000 \\
01000000000000010000 \quad 00001000000000000000 \\
01000000000000001000 \quad 00001000000000000000 \\
01000000000000000100 \quad 00001000000000000000 \\
01000000000000000010 \quad 00001000000000000000 \\
01000000000000000001 \quad 00001000000000000000 \\
01000000000000000000 \quad 00001000000000000000 \\

(15, 307) WUM Code: This code uses up all of the words in \( \Omega_4^{15} \). It consists only of \((15, 4)\) WUM sets. To describe the construction of this code, we first introduce a few basic permuta-
The (15, 307) WUM code consists of the following subsets of $\Omega_5^{15}$:

a) $\{00000000000000\}$; 
b) $S_i S_j S_k A_l$, $0 \leq i < 4$, $0 \leq j < 4$, $0 \leq k < 4$; 
c) $S_i A_j$, $0 \leq i < 4$, $2 \leq j < 3$; 
d) $T^i R^j A_k$, $0 \leq i < 2$, $0 \leq j < 1$; 
e) $S^i T^j R^k A_l$, $0 \leq i < 4$, $0 \leq j < 2$, $0 \leq k < 1$, $5 \leq l \leq 7$; 
f) $S^i T^j A_l$, $0 \leq i < 4$, $0 \leq j < 2$, $8 \leq l \leq 12$.

**Remark:** For these two codes, only minimal WUM sets and all the elements in $\Omega_5^2$ are used. This leads us to conjecture that these two codes are optimal for their blocklengths.

**REFERENCES**


