1979-04-17

A LOWER BOUND ON THE KEY EQUIVOCATION FOR THE SIMPLE SUBSTITUTION CIPHER APPLIED ON A BINARY MEMORYLESS SOURCE

Rolf Blom

INTERNAL REPORT
LiTH-ISY-I-0288
ABSTRACT

A lower bound on the key equivocation of the simple substitution cipher applied on a binary memoryless source is derived. A comparison with existing lower bounds shows that this new bound performs better for short and moderately long cryptogram sequences.
CONTENTS

1. Introduction 1
2. The Lower Bound 3
3. Comparison with other bounds 6
   Appendix 10
   References 12
1. INTRODUCTION

The simple substitution cipher applied on a binary source can be described as in fig. 1. The message source alphabet $M$, the key source alphabet $K$ and the cryptogram alphabet $E$ are equal and $M=K=E=\{0,1\}$. The two enciphering transformations possible are described as addition modulo 2 of a key and the message symbols. This is a secrecy system according to [1].

![Blockdiagram showing how the simple substitution cipher is applied on a binary message source](image)

The wiretapper tries to find the message transmitted and/or the key used. In addition to the intercepted cryptogram he is assumed to know the statistics of the message and key sources and the details of the cipher. To measure his uncertainty about which message and which key that was transmitted and used respectively, the entropies of the
message and key given the cryptogram are used. These conditional entropies are called the key and the message equivocation respectively.

In this paper we derive a lower bound on the key equivocation when the message source is memoryless and the keys are equiprobable. This new bound is compared with existing bounds.
2. THE LOWER BOUND

The assumptions given in the introduction can be summarized as follows: The model used is a secrecy system. The message source is binary and memoryless with alphabet $\mathcal{M} = \{0, 1\}$. The cryptogram alphabet $\mathcal{E}$ equals $\mathcal{M}$ and the set $\mathcal{T}$ of enciphering transformations is the set of all bijective maps of $\mathcal{M}$ to $\mathcal{E}$. The keys are equiprobable.

Let the a priori probabilities of the message symbols be $P_M(0) = q_0$ and $P_M(1) = 1 - q_0 = q_1$. Let $H(K|E^L)$ denote the key equivocation when $L$ cryptogram symbols are observed. The key equivocation is measured in nats and all logarithms in this paper are taken to the base $e$. The entropy function $h(x)$ is defined by

$$h(x) = -x \log(x) - (1-x) \log(1-x).$$  \hspace{1cm} (1)

According to equation 22 in [2] the key equivocation under the assumptions above can be written as

$$H(K|E^L) = \sum_{x=0}^{L} \binom{L}{x} q_0^x q_1^{L-x} \log \left( \frac{q_0^x q_1^{L-x}}{q_0^x q_1^{L-x}} \right)$$  \hspace{1cm} (2)

To simplify the notation we define

$$a(L, x) = \frac{x^{L-x}}{q_0^x q_1^{L-x}}.$$  \hspace{1cm} (3)

and write (2) in the following way

$$H(K|E^L) = \frac{1}{2} \sum_{x=0}^{L} \binom{L}{x} \left[ a(L, x) \log \left( \frac{a(L, x) + a(L, L-x)}{a(L, x)} \right) + a(L, L-x) \log \left( \frac{a(L, x) + a(L, L-x)}{a(L, L-x)} \right) \right]$$  \hspace{1cm} (4)
The lower bound of the key equivocation is given in the following Theorem:

**Theorem 1**: If a binary memoryless source is enciphered by a simple substitution cipher with equiprobable keys and the a priori probabilities of the message symbols are \( q_0 \) and \( q_1 = 1 - q_0 \), then

\[
H(K|E_L^L) \geq (h(q_0)/\log(2))^L \log(2)
\]  

The idea behind the proof is to determine a parameter \( \alpha \) such that

\[
H(K|E_{L+1}^L) \geq \alpha H(K|E_L^L)
\]  

for all \( L = 0, 1, 2, \ldots \). In the proof we need the results stated in the two Lemmas below. The first Lemma is used in the proof of the second Lemma. The proofs are given in an Appendix.

**Lemma 1**: If \( 0 \leq x < 1 \) then

\[
h(x) \geq 4x(1-x)\log(2)
\]  

**Lemma 2**: If \( 0 \leq x, y < 1 \) and \( z = x(1-y) + (1-x)y \) then

\[
h(x) + h(y) - h(z) - h(x) \cdot h(y)/\log(2) \geq 0
\]  

**Proof of Theorem 1**: The relation

\[
\binom{L+1}{x} = \binom{L}{x-1} + \binom{L}{x}
\]  

between binomial coefficients gives that the expression of \( H(K|E_{L+1}^L) \) given by (4) can be written as
\begin{equation}
H(K|E_{\sim}^{L+1}) = \frac{1}{2} \sum_{0}^{X} \binom{L}{x} A(L,x) \tag{10}
\end{equation}

where

\begin{equation}
A(L,x) = q_0 a(L,x) \log \left( \frac{q_0 a(L,x) + q_1 a(L,L-x)}{q_0 a(L,x)} \right) + q_1 a(L,x) \log \left( \frac{q_1 a(L,L-x) + q_0 a(L,L-x)}{q_1 a(L,x)} \right) + q_0 a(L,L-x) \log \left( \frac{q_0 a(L,L-x) + q_1 a(L,L-x)}{q_0 a(L,L-x)} \right) + q_1 a(L,L-x) \log \left( \frac{q_1 a(L,L-x) + q_0 a(L,L-x)}{q_1 a(L,L-x)} \right) \tag{11}
\end{equation}

\( A(L,x) \) can be rewritten as

\begin{equation}
A(L,x) = [a(L,x) + a(L,L-x)] \left[ h(q_0) + h\left( \frac{a(L,L-x)}{a(L,x) + a(L,L-x)} \right) - h\left( \frac{q_0 a(L,x) + q_1 a(L,L-x)}{a(L,x) + a(L,L-x)} \right) \right] \tag{12}
\end{equation}

and identification of terms in the second factor in (12) with the variables used in Lemma 2 shows that

\begin{equation}
A(L,x) \geq \frac{h(q^0)}{\log(2)} [a(L,x) + a(L,L-x)] h\left( \frac{a(L,L-x)}{a(L,x) + a(L,L-x)} \right) =
\end{equation}

\begin{equation}
= \frac{h(q^0)}{\log(2)} \left[ a(L,x) \log \left( \frac{a(L,x) + a(L,L-x)}{a(L,x)} \right) + a(L,L-x) \log \left( \frac{a(L,x) + a(L,L-x)}{a(L,L-x)} \right) \right] \tag{13}
\end{equation}

Substitution of (13) into (10) and comparison with (4) gives

\begin{equation}
H(K|E_{\sim}^{L+1}) \geq \frac{h(q^0)}{\log(2)} \frac{H(K|E_{\sim}^{L})}{H(K)} \tag{14}
\end{equation}

From (2) it is evident that \( H(K|E_{\sim}^{L}) = h(q^0) \) and \( H(K) = \log(2) \) and a simple induction argument gives (5).
3. COMPARISON WITH OTHER BOUNDS

In figures 2, 3 and 4 we have plotted the bound of (5) and the following bounds fetched from equations (10), (47) and (28) in [2] evaluated for the problem considered.

\[
H(K|E^L) \geq \log(2) - L(\log(2) - h(q_0)) \quad L \geq 0
\]

\[
H(K|E^L) \geq \sqrt{\frac{2}{\pi L}} \left( \frac{2L}{2L+1} \right)^2 (\sqrt{4q_0q_1})^L \log(2) \quad L = 2, 4, 6, \ldots
\]

\[
H(K|E^L) \leq (\sqrt{4q_0q_1})^L \log(2)
\]

First we observe that (15) gives a straight line as a lower bound, and that (5) and (15) are equal for \( L = 0 \) and \( L = 1 \). Both bounds give the correct value of \( H(K|E^L) \) for these points. Secondly, from the figures it is obvious that (5) gives a much better lower bound than (16) for moderate values on \( L \). Thirdly, \( \sqrt{4q_0q_1} \) is a fairly good approximation of \( h(q_0)/\log(2) \). As a matter of fact it is stated in Corollary 1 in [2] that

\[
h(q_0) \leq \sqrt{4q_0q_1} \log(2)
\]

and the difference between (17) and (5) is small especially when \( q_0 \) is close to 0.5.
Figure 2. Plot of bounds on the key equivocation when $q_0 = 0.35$. 
Figure 4. Plot of bounds on the key equivocation when $q_0 = 0.45$. 
APPENDIX

Proof of Lemma 1. Let \( f(x) \) denote

\[
f(x) = h(x) - 4x(1-x)\log(2).
\]

(19)

\( f(x) \) is symmetric about \( x=0.5 \), that is \( f(x) = f(1-x) \) when \( 0 \leq x \leq 1 \). Hence it is sufficient to show that \( f(x) \geq 0 \) when \( 0 \leq x \leq 0.5 \). The first and second derivatives of \( f(x) \) are

\[
\frac{d}{dx} f(x) = f'(x) = \ln(\frac{1-x}{x}) - 4(1-2x)\log(2)
\]

(20)

\[
\frac{d^2}{dx^2} f(x) = f''(x) = -\frac{1}{x(1-x)} + 8\log(2)
\]

(21)

Note that \( f(x) \) and its derivatives are continuous functions. We see that the equation \( f''(x) = 0 \) will have a single root \( x_0 \) in the interior of the interval \( [0,0.5] \). The change of sign of \( f''(x) \) at \( x = x_0 \) shows that \( x_0 \) corresponds to a minimum in \( f'(x) \). Hence \( f'(x) \) has only one extremal point in \( [0,0.5] \). \( f'(x_0) < 0 \) because \( f'(0.5) = 0 \) and

\[
\lim_{x \to 0^+} f'(x) = +\infty
\]

(22)

Thus, the equation \( f'(x) = 0 \) has only a single root in \( [0,0.5] \) and it corresponds to a maximum in \( f(x) \). But \( f(0) = f(0.5) = 0 \) which implies that \( f(x) \geq 0 \) when \( 0 \leq x \leq 0.5 \).

Proof of Lemma 2. We use the same technique as in the proof of Lemma 1. Let \( f(x) \) denote

\[
f(x) = h(x) + h(y) - h(z) - h(x)h(y)/\log(2)
\]

(23)

where \( z = x(1-y) + (1-x)y \). \( f(x) \) is symmetric around \( x=0.5 \) as can easily be verified. The right side of (23) is symmetric in \( x \) and \( y \). Hence it is sufficient to prove that \( f(x) \geq 0 \) when \( 0 \leq x \leq 0.5 \) and \( 0 \leq y \leq 0.5 \). \( y=0 \) gives \( z=x \) and
\( f(x) = 0 \), when \( y = 0.5 \) then \( z = 0.5 \) and \( f(x) = 0 \). It remains to prove \( f(x) \geq 0 \) when \( 0 < y < 0.5 \) and \( 0 < x < 0.5 \). The first and second derivatives of \( f(x) \) are

\[
\frac{d}{dx} f(x) = f'(x) = \ln \left( \frac{1-x}{x} \right) \left[ 1 - \frac{h(y)}{\log(2)} \right] - (1-2y) \log \left( \frac{1-x(1-y)-(1-x)y}{x(1-y)+(1-x)y} \right)
\]

(24)

\[
\frac{d^2}{dx^2} f(x) = f''(x) = -\frac{(1-2y)^2}{x(1-x)} \left[ \frac{\log(2)-h(y)}{\log(2)(1-2y)^2} - \frac{1}{1+2y^2-2y+y(1-y)\left[ \frac{x}{1-x}, \frac{1-x}{x} \right]} \right]
\]

(25)

We observe that the first term inside the brackets of (25) is less than 1 because the difference between the nominator and the denominator is

\[
\log(2) - h(y) - \log(2)(1-4y(1-y)) = 4y(1-y)\log(2) - h(y) \leq 0
\]

(26)

according to Lemma 1. The second term inside the brackets of (25) is equal to 1 when \( x = 0.5 \) and it is monotonically increasing in \( x \) in the interval \([0, 0.5]\).

Hence \( f''(x) = 0 \) has a single root in the interval \([0, 0.5]\) and it corresponds to a minimum in \( f'(x) \). Now \( f'(0, 5) = 0 \) and

\[
\lim_{x \to 0^+} f'(x) = +\infty.
\]

(27)

We also have \( f(0) = f(0.5) = 0 \). Hence the same arguments as in the proof of Lemma 1 gives \( f(x) \geq 0 \).
REFERENCES


A MULTIPLE USER SECRECY SYSTEM

Rolf Blom

INTERNAL PUBLICATION
LiTH-ISY-I-0289
ABSTRACT

In this paper we extend the model of a secrecy system, which have a single source communicating with a single destination, into a multiple user secrecy system. A multiple user secrecy system has several message sources that transmits messages over a common channel to a common destination. The concepts of message and key equivocations in a single user secrecy system are generalized to fit the new model. Two relevant new measures are also defined and used in the analysis of two examples.
## CONTENTS

1. Introduction  
   1

2. Notation  
   3

3. The Model  
   5

4. Performance Measures  
   8

5. Coder Design, A Simple Example  
   12

6. A Class of Multiple User Secrecy Systems  
   19

7. Concluding Remarks  
   26

   Acknowledgement  
   27

   References  
   28
1. INTRODUCTION

The block diagram in Figure 1 shows a model of how ciphers are used to protect the content of a message against un-toward exposure for the situation in which a single message source sends messages to a single destination. Before transmission each message symbol is transformed in the encipherer into a cryptogram symbol. The transformation performed is determined by the output from the key source. To recover the message at the receiving end the inverse of the enciphering transformation is applied to the cryptogram symbols.

The object of a wiretapper, who intercepts the cryptograms, is to find the cleartext message and/or the key used. Naturally, the applicable methods and the difficulty to reach the goal varies with the wiretapper's a priori knowledge about the different parts of the model.

The theoretical framework of this paper is fetched from Shannon's paper [1], "Communication Theory of Secrecy Systems". We assume that the reader is familiar with that paper and use definitions and results presented there without any further reference. In [1] Shannon analyzed the situation described above and defined a formal model called a secrecy system. Let us call it a single user secrecy system (SUSS) to indicate that only a single message source is involved.

In this paper we extend the SUSS into a model that has a number of independent message sources which are to be enciphered and transmitted to a common destination over a common channel. We call our model a multiple user secrecy system (MUSS). There exists a new threat against the protection of messages in a MUSS compared to the threats against a SUSS. This new threat occurs when the wiretapper
knows the cleartext from some of the message sources, intercepts the corresponding cryptogram and tries to decrypt some other messages. An example of a practical situation which could be described as a MUSS is an enciphered transmission line between two concentrators.

In this paper we have two goals. The first is to introduce our model of a MUSS. Secondly, we want to show by considering a special case that this model is relevant. In section 2 we present the notation used. Our model of a MUSS is described in section 3 and in section 4 we discuss system permanence measures. Section 5 contains an analysis of a simple example which serves as an introduction to the class of MUSS's analyzed in section 6. The results are summarized in section 7 which also contains ideas for further research.

![Blockdiagram](image)

Figure 1. Blockdiagram depicting the flow of information in a secrecy system.
2. **NOTATION**

Let \( N \) denote the set of positive integers. With \( N \subseteq \mathbb{N} \), \( I_N = \{1, 2, \ldots, N\} \). The number of elements in a set \( V \) is denoted \( |V| \). The empty set is denoted \( \emptyset \). A set containing a single element will often be referred to by its element.

Let \( V \) be an arbitrary finite set. A sequence of length \( L \) of symbols in \( V \) will be denoted \( v^L \). The ensemble of sequences of length \( L \) in \( V \) is written \( V^L \). Random sequences and variables will be denoted by uppercase letters in a corresponding way.

Let \( V \) be the cartesian product of the sets \( U_j, j \in I_D \), this is denoted by \( V = U_1 \times U_2 \times \cdots \times U_D \). If all \( U_j = U \) we also write \( V = U^D \). An arbitrary element \( v \in V \) will be represented as \( v = (v_1, v_2, \ldots, v_D) \) where \( v_j \in U_j \). To refer to a subset of the components in \( v \) given by a set \( J \subseteq I_D \) we introduce the function \( c(\cdot', \cdot') \) which is defined as \( c(J,v) = (v_{j_1}', v_{j_2}', \ldots, v_{j_D}') \) where \( j_i \in J, j_i < j_{i+1}, |J| = D' \). We also operate with \( c(\cdot, \cdot') \) on sequences \( v^L \), then it is understood that \( c(J, \cdot') \) is applied on each component in \( v^L \) and thus \( c(J, v^L) \) will give a new sequence of length \( L \).

A shorthand notation for \( c(J,v) \) is \( v_{j'} \) which also is used for sequences, \( v^L_{j'} = c(J,v^L) \). To denote the subspace that \( v_{j'} \) belongs to we sometimes write \( V_{j'} \).

Let \( V \) be an arbitrary finite set with \( |V| = N \), then the set of all bijective transformations from \( V \) to \( V \) is denoted \( G(V) \). It is a group of transformations with \( N! \) elements and it can be identified with the group of permutations of \( N \) objects. We use the same convention as above to denote cartesian products of sets of transformations. The direct product of two groups \( A, B \) is written \( A \times B \). If \( f \) is a transformation, it's inverse is denoted by \( f^{-1} \) and if \( A \) is a set of transformation, the set of all inverse transformations of those in \( A \) is denoted \( A^{-1} \).
The device of juxtaposing two letters $u,v$ is so efficient that we will use it in two different senses, according to the meaning previously announced for the letters. Thus, if $u$ is a transformation and $v$ an element of its domain, $uv$ denotes the value of $u$ for the assignment $v$. If $u$ and $v$ both are functions $uv$ will denote the composite function.

In the same way as for the function $c(\cdot,\cdot)$, the expression $fv^L$ should be interpreted as $f$ applied on each component in $v^L$.

The notation of standard information quantities are as defined by Gallager [2].
3. THE MODEL

The model of a MUSS that we use has a certain structure. Let us therefore, describe the problem that made us chose this structure before we formally defined the model.

The problem considered can be formulated as follows: A number of sources generating message sequences should be enciphered and transmitted to a common destination. For the enciphering a single enciphering unit should be used. How can the enciphering unit be used to give protection of the messages even in the case when the wiretapper has observed the cleartext from some of the sources.

A straightforward solution to the problem would be to time multiplex the different sequences and then use the given enciphering unit to encipher the multiplexed sequence. If different keys are used for symbols from different sources this would be a safe procedure in the sense that if the given enciphering unit gives good protection of a message all sequences will get this protection. But is this the best way to do it and how should the message sequences be mixed if we restrict the number of keys available?

The structure of the model we propose is shown in Figure 2. With this structure we can model the time multiplex solution and other more complicated mixing schemes. We can also have sources with different alphabets. Informally it can be described as follows. The output symbols from the different message sources are fed into a coder (multiplexer). The coder outputs a vector with components belonging to the input alphabet of the encipherer. The enciphering is then performed separately on each component in the vector and the key may vary between components. At the receiving end the inverse operations are performed to recover the messages.
Figure 2. Block diagram depicting the structure and information flow in a multiple user secrecy system.
Now a formal description of the different parts of the model. The number of message sources is denoted by $J$. The message sources are discrete and independent with alphabets $X_j = I_{N_j}$, $j \in J$. A source will be referred to by its alphabet. We denote the cartesian product of all message source alphabets by $X$, $X = X_1 \times X_2 \times \ldots \times X_J$.

The input alphabet to the encipherer is $I_N$ and the dimension of the output vector from the coder is denoted by $D$. With $V = I_N^D$, the restriction we put on the coder mapping $f : X \rightarrow V$ is that it should be bijective. This implies that the relation

$$\prod_{j \in J} N_j = N^D \quad (1)$$

between the sizes of the message source alphabets and the input alphabet to the encipherer has to be satisfied.

The set of enciphering transformations $T$ is a subset of $G(I_N)^D$. To distinguish between input and output of the encipherer we use $Z = V$ to denote the output alphabet. Hence, with $t \in T$ we have $t : V \rightarrow Z = V$.

The specific transformation used to encipher a set of messages is chosen at random among the elements in $T$. In the figure this is indicated by the key source generating a key $k \in I_K$, $K_0 = |T|$, which specifies the transformation to be used. In the model we include the assumption that the keys are equiprobable.

At the receiving end the decipherer performs the inverse transformation of the encipherer. The set of deciphering transformations is denoted by $S$ and the decoder mapping is denoted by $r$, hence, $S = T^{-1}$ and $r = f^{-1}$. 
4. PERFORMANCE MEASURES

By system performance we will refer to the degree of protection that messages in a MUSS will have against cryptanalysis. We will look at the problem of generalizing the key and message equivocations used in a SUSS. We also introduce two new measures for a MUSS that are related to alphabet size and number of keys in a SUSS.

We always assume that a wiretapper knows the statistics of the individual sources \(X_j, j \in I_j\). He also knows the coder mapping \(f: X \rightarrow Y\), the set \(T\) of enciphering transformations and that the keys are equiprobable. Using this a priori knowledge, an intercepted cryptogram sequence and perhaps knowledge of one or more corresponding cleartext sequences of individual sources, the wiretapper tries to find a key that will correctly decipher the message sequences that he is interested in or he tries directly to determine the sequences.

In a SUSS the key and message equivocations are the entropy of the key and the message given the cryptogram. The type of equivocation that can be generalized to a MUSS right away is the message equivocation. For the case \(J = 3\) we give some examples of message equivocations: 

\[
\begin{align*}
H(X^L_1|Z^L) &= H(X^L_1X^L_2X^L_3|Z^L), \\
H(X^L_1X^L_2|Z^LX^L_3) &= H(X^L_1|Z^L), \\
H(X^L_1|Z^LX^L_2) &= H(X^L_1|Z^LX^L_2X^L_3). 
\end{align*}
\]

These examples reflect different cases in which the wiretapper does or does not know some cleartext message sequences. A simple calculation gives that there is \(3^J - 2^J\) different message equivocations of the types exemplified above.

To define equivocations corresponding to the equivocation of the key in a SUSS we need consider the wiretapper's situation in more detail. It should be obvious that if the wiretapper intercepts a cryptogram sequence and it is deciphered with two different transformations this might give
the same message sequence for a particular message source. Thus, keys may be equivalent if the purpose only is to decipher a subset of the message sources.

To formalize the notion of equivalent keys we define a set of equivalence relations between deciphering transformations. Let $s_1, s_2 \in S$ and $q \in I_j$ then $s_1$ and $s_2$ are $q$-equivalent if $c(q, rs_1, z) = c(q, rs_2, z)$ for all $z \in Z$. From elementary algebra we know that an equivalence relation defined on the elements of a set divides the set into disjoint equivalence classes. The set of $q$-equivalence classes in $S$ is denoted by $S_q$.

A word about notation. In this section we will denote the set of indices of the message sources $X_i$ which the wiretapper tries to decrypt by $q$. The indices of the set of message sources $X_i$ for which the wiretapper has observed the plaintext is denoted $P$. Of course, $P, q \in I_j$ and $P \cap q = \emptyset$.

Now we can see that a wiretapper who only wants to decipher the messages from sources $X_j$, $j \in q$, only has to consider one transformation out of each $q$-equivalence class. Hence, the number of keys he has to consider is $|S_q|$ and his a priori knowledge of the keys should be related to $S_q$.

To each transformation in $S$ a key is associated. Hence, the $q$-equivalence relation defined above can also be used to form corresponding equivalence classes among the keys. Let $K_q$ denote the stochastic variable defined by these key equivalence classes. Then the generalized equivocations of the key are defined as expressions of the type $H(K_q | Z^L)$ or $H(K_q | Z^L_1 \cdots Z^L_k \cdots )$, Observe that the probability of a given equivalence class is proportional to the number of elements in this class, hence the equivalence classes may not be equiprobable. Also observe that for intercepted sequences that do not contain all symbols in $Z$, transformations in different classes may be equivalent.
The discussion above shows that there exists a natural way to generalize both message and key equivocations in a SUSS to meaningful corresponding entities in a MUSS. However, the problem of evaluating these equivocations is not trivial. For a SUSS there only exists very weak bounds, [1], [3] on the key equivocation for an arbitrary system. Because of this lack of a measure that is easily evaluated we introduce two new measures.

The two new measures are not probabilistic but are related to key size and alphabet size. To explain how these measures were developed consider a case when the wiretapper has intercepted $z^L$ and knows the corresponding sequence $x_p^L$. Then he can by a deterministic algorithm find all deciphering transformations that from $z^L$ will yield $x_p^L$. Let us denote this set by $S(x_p^L, z^L) = \{s \in S | x_p^L = c(P, rsz^L)\}$. If $x_p^L \not= c(P, rsz^L)$ for all $s \in S$ (an error event) we assume that $S(x_p^L, z^L) = T$.

Now assume that the wiretapper wants to decipher sources $x_j$, $j \in Q$. Then he can form $Q$-equivalence classes of the transformations in $S(x_p^L, z^L)$. Denote the set of $Q$-equivalence classes in $S(x_p^L, z^L)$ by $S_Q(x_p^L, z^L)$. Then $|S_Q(x_p^L, z^L)|$ represents the number of keys that the wiretapper has to guess between. This number may be used as a measure to indicate "key size". We define the deterministic key uncertainty (DKU), denoted $v_k(Q|P)$ as

$$v_k(Q|P) = \min_{L \in \mathcal{N}} \min_{z^L \in z^L} \min_{x_p^L \in x_p^L} |S_Q(x_p^L, z^L)|$$

which gives the "key size" of the worst case for the legitimate users of the system. If $z^L = tfx^L$ for some $x \in X$ and $t \in T$ and $z^L$ contains all symbols in $Z$ then $S(x_p^L, z^L)$ will be equal to a $Q$-equivalence class in $S$. Hence, we may conclude that $v_k(Q|P)$ is the minimum number of $Q$-equivalence classes in a $P$-equivalence class.
The other measure we introduce reveals the least number of possible messages that a wiretapper may have to guess between. We call this the **deterministic message uncertainty** (DMU) and it is defined by

\[
V_m(Q|P) = \min_{L \in \mathbb{N}} \min_{z^L \in \mathcal{Z}^L} \min_{x_p^L \in X^L_p} \| \{ \text{c(Q, rsz^L)} | s \in S(x_p^L, z^L) \} \|
\]

That is, independent of which cryptogram sequence \(z^L\) and message sequence \(x_p^L\) the wiretapper knows, there is always at least \(V_m(Q|P)\) sequences \(x_p^L\) into which the cryptogram sequence can be decrypted.

The relevance of the DKU and DMU measures is due to their definition as worst case indicators. Naturally one would like to have a MUSS with the values of the DKU and DMU measures as large as possible. They may also be used when one tries to build a uniformly good system, that is they indicate if all enciphering transformations approximately give the wiretapper the same cryptanalytic problem.
5. CODER DESIGN, A SIMPLE EXAMPLE

In this section we consider a specific simple MUSS in which we have chosen the set \( T \) of enciphering transformations to be a group. With this example as a starting point we discuss the problem of coder design. The measures considered are DKU's and DMU's.

The choice of \( T \) as a group of transformations makes it possible to use results derived for pure ciphers. We assume that the reader is familiar with the definition and properties of pure ciphers to which we will refer freely without giving any specific references. For an introduction to pure ciphers see [4].

Using the notation of section 3 we specify the parameters in the example as \( D=2, N_1=N_2=N=4 \) and \( T=\{ g \in G(I_4)^2 | c(1, g) = c(2, g) \} \). As there are \( 4! \) different transformations in \( G(I_4) \) the number of keys is \( K_0 = 24 \). That \( T \) as defined above is a group is obvious and by definition the keys in a MUSS are equiprobable. Hence, with \( \mathcal{V} \) taken as message alphabet, \( T \) constitutes a pure cipher.

As the group property of \( T \) implies that \( T^{-1} = T \) we will not make any distinction between the sets of enciphering and deciphering transformations.

Figure 3 on page 18 contains an example of a coder mapping designed along the ideas presented below. In the figure is also shown the actual form of different sets defined during the discussion. It may be helpful to look at the figure at various stages of the discussion to get a more tangible feeling for what is being done.

From the properties of pure ciphers it follows that enciphering of \( \mathcal{V} \) with \( T \) will divide \( \mathcal{V} \) into two residue classes.
They are $C_1 = \{ y \in V | c(1, y) = c(2, y) \}$ and $C_2 = \{ y \in V | c(1, y) \neq c(2, y) \}$. For $i \in I_2$ we have $ty \in C_1$ when $y \in C_1$, $t \in T$ and for arbitrary $y_1, y_2 \in C_1$ there exist a $t \in T$ such that $ty_1 = y_2$. This implies that a given cryptogram $z \in C_1$ can be deciphered into any one of the messages mapped on $C_1$. The number of elements in $C_1$ and $C_2$ are 4 and 12 respectively.

The DMU is defined as the least number of possible messages among which a wiretapper ever has to guess. We see that if the wiretapper intercepts a single cryptogram symbol $z \in C_1$ he will only have the 4 messages mapped on $C_1$ to chose between. Hence, $v_m([1,2])$ can be upper bounded by 4. Naturally we only obtain an upper bound because we have not considered what happens when the wiretapper intercepts a sequence of cryptogram symbols. The same reasoning as above for a cryptogram symbol $z \in C_2$ gives $v_m([1,2]) \leq 12$. Thus, we see that the residue class with least number of elements constrains the value of $v_m([1,2])$. We suspect that this is the case for all types of DMU's.

The other DMU's of interest are $v_m(1)$, $v_m(2)$, $v_m(1|2)$ and $v_m(2|1)$. The last two are applicable when the wiretapper knows one of the two messages. For further analysis of our example we define some new sets. Let $X' \subset X$ denote the subset of messages which is mapped on $C_1$ by the coder mapping. Then we define the sets $X'_i = \{ c(i, x) | x \in X' \}$ for $i \in I_2$. This means that $X'_i$ contains the symbols in $X_i$ represented in the elements in $X'$. Using the sets defined above, the DMU's mentioned above can be upper bounded by

$$v_m(i) \leq |X'_i| \tag{4}$$

$$v_m(i|3-i) \leq \min_{x_{3-i} \in X'_{3-i}} \{ c(i, x) | x \in X' \text{, } c(3-i, x) = x_{3-i} \} \tag{5}$$

We observe that if $|X'_{3-i}| \geq 3$ there must exist an element in $X'$ with unique $(3-i):$th component and thus $v_m(i|3-i) = 1$. 

With $|x'_{3-1}|=1$ we get $v_m(i|3-i)\leq 4$ but $v_m(3-i|i)=1$. The conclusion we draw is that if we shall be able to design a coder with all DMU's greater than 1 we have to chose the set $X'$ in such a way that $|X'_1|=|X'_2|=2$. This in turn implies that $X'$ must have the form

$$X' = \{(a,c),(a,d),(b,c),(b,d)\} \quad a \neq b, \quad c \neq d, \quad a,b,c,d \in I_4.$$  

(6)

For a set $X'$ chosen in accordance with (6) we see that the DMU's concerning a single source are upper bounded by 2. This is true for an arbitrary bijective mapping which has $C_1$ as image of $X'$.

In the following we assume that $X'$ has the form given in (6). We define $f_1$ to be a fixed but arbitrary bijective mapping $f_1:X'\to C_1$. Then we show how $f_1$ can be extended into a coder mapping $f:X\to Y$. To do this define

$$H_i = \{z \in Z | c(i,f_1^{-1}y) = c(i,f_1^{-1}y) \text{ for all } y \in C_1 \} \quad i \in I_2.$$  

(7)

That $H_i$ is a group is immediate from its definition. Then if we use $H_i$ as a set of enciphering transformations on $Y$ we will get a new partitioning of $Y$ into residue classes. Each one of the new residue classes is a subset of the residue classes generated with $T$. This is so because $H_i$ is a subgroup in $T$.

From the definitions of $X'$ and $H_i$ we see that the elements in $X'$ with equal $i$:th component will belong to the same residue class generated by $H_i$. This observation can be extended into a construction principle. To define a coder mapping a distinct element in $X_i$ is associated with each residue class of $Y$ generated by $H_i$. For this to define a correct coder mapping we have to ensure that the total number of elements belonging to residue classes associated with an element in $X_i$ is $|X|/|X_i|$. To show that this con-
dition can be satisfied in our example is a straightforward task. The result is that \( H_i \) generates two residue classes with 4 elements and four residue classes with 2 elements and using this technique we extend \( f_1 \) into a coder mapping.

What remains to show is that \( v_m(\{1,2\}) = 4 \) and that \( v_m(1) = v_m(2) = v_m(1|2) = v_m(2|1) = 2 \). Assume that an arbitrary message sequence \( x^L \) is enciphered with \( t \). Then the cryptogram sequence \( z^L \) is \( z^L = tfx^L \). If the wiretapper does not know any of the individual message sequences in \( x^L \), he only knows that \( t \) belongs to \( T \). Besides, this means that \( v_k(\{1,2\}) = 24 \) as expected. If any of the components in \( z^L \) belong to \( C_2 \), that component may stem from 12 different messages. Thus, a cryptogram sequence containing at least one symbol in \( C_2 \) can be deciphered into at least 12 different message sequences. A component in \( z^L \) belonging to \( C_1 \) is deciphered into one out of four messages. Hence, the value of \( v_m(\{1,2\}) = 4 \) and it is achieved by sequences \( z^L \) which have all components equal to a single element in \( C_1 \). A similar argument shows that \( v_m(1) = v_m(2) = 2 \).

To prove that \( v_m(1|2) = v_m(2|1) = 2 \) assume that the wiretapper has intercepted a cryptogram sequence \( z^L \) and that he knows the corresponding sequence \( x_i^L \) generated by source \( X_i \). The wiretapper can now determine a transformation \( t \) such that \( c(i, rtz^L) = x_i^L \). By construction \( c(i, rhtz^L) = x_i^L \) when \( h \in H_i \). Let \( y \) denote an arbitrary component in \( y^L = tz^L \). Then the minimum number of elements in \( \{ hy | h \in H_i \} \) is equal to the size of the residue class in \( Y \) generated by \( H_i \) with least number of elements. As the \( i \)-th components of the elements mapped into this residue class are equal the \( (3-i) \)-th component must be distinct. The least number of elements in a residue class is 2 which gives a lower bound on \( v_m(3-i|i) \) by 2. As discussed earlier the construction gave \( v_m(3-i|i) \leq 2 \) which gives \( v_m(3-i|i) = 2 \).

The evaluation of \( v_k(1) \), \( v_k(2) \), \( v_k(1|2) \) and \( v_k(2|1) \) is rather trivial. If the wiretapper does not know any of
the individual message sequences corresponding to a crypto
togram sequence, he cannot exclude any of the keys. Hence 
v_k(1)=v_k(2)=K_0. For the case when he knows x^l_1 correspond-
ing to z^L, an argument along the lines used to lower bound 
v_m(3-i|i) shows that v_k(3-i|i)=|H_1|=4.

To conclude this section we give a very simple implement-
tion of the coder mapping specified in figure 3. We 
observe that for example the element 1 in X_1 is mapped 
into elements of Y in which the first and second compo-
nents belong to the set A(1,0)={1,3}. The element 2 in X_1 
is always mapped into an element in Y with first component 
in A(1,1)={2,4} and second component in A(1,0). As a matter 
of fact an element in X_1 is always mapped into elements 
of Y for which each component belong to either A(1,0) or 
A(1,1). The same thing is true for the elements in X_2 
with A(1,0) and A(1,1) replaced by A(2,0)={1,2} and 
A(2,1)={3,4} respectively. That we obtain these relations 
is not due to destiny but to the fact that for i\in I_2, j_1, 
j_2\in\{0,1\} an element h\in H_1 maps A(i,j_1)\times A(1,j_2) onto itself.

We can specify the coder mapping in the following way: 
x=(x_1,x_2) is mapped on the element y=(y_1,y_2) which have 

\[ y_j = A(1,b_2(j,x_1)) \cap A(2,b_2(j,x_2)) \quad j\in I_2 \]  

where \( b_2(\ell,m) \) is defined as a coefficient in the binary 
representation of m-1 as indicated in the following equa-
tion:

\[ m-1 = \sum_{\ell=1}^{2} b_2(\ell,m) 2^{\ell-1} \quad m\in I_4. \]  

Thus, the coder mapping may be implemented in a very 
simple manner. The only operations involved are inter-
sections of sets and representation of numbers in base 2. 
With suitable representation of the sets, intersections
may be easily accomplished, and usually numbers are represented in base 2 in digital equipment.

A decoder can be designed that have the same structure as the coder. As a matter of fact the coder mapping is its own inverse which is shown for a general case in section 6. There it is also shown how the mapping may be implemented in an even simpler way.
6. A CLASS OF MULTIPLE USER SECRECY SYSTEMS

The class of MUSS's treated in this section have a fixed coder mapping. The construction of the coder mapping is based on the technique exemplified in section 5. The set of enciphering transformations is assumed to belong to a certain class defined below. For the members of this class we evaluate the DMU's and DKU's and compare them with a MUSS performing encryption on a time multiplexed sequence of messages.

The members of the class of MUSS's considered have $D=J$ and $N_{j}=N=a^{D}$, $a>2$ for all $j \in I_{j}$. The class of sets of enciphering transformations is characterized by the condition that a member $T$ in the class can be defined as

$$T = \{ g \in \mathcal{G}(I_{N})^{D} | c(j,g) = c(k,g) \quad j,k \in U_{1}, \quad i \in I_{D}, \}$$  \hspace{1cm} (10)

where $1 \leq D' \leq D$ and the sets $U_{1}$ are disjoint and collectively contain all elements in $I_{j}$. This means that the encryption of an element $y \in Y$ is such that the components in $y$ specified by $U_{1}$ are encrypted with one transformation in $\mathcal{G}(I_{N})$, for the components specified by $U_{2}$ another transformation is used and so on. We call this encryption with $D'$ independent keys. Naturally, a set of transformation defined according to (10) becomes a group.

To give a formal definition of the coder mapping we introduce the following notation. Let $b_{a}(j,i)$ denote the $j$:th digit in the representation of $i-1$, $i \in I_{N}$ in base $a$, that is

$$i-1 = \sum_{j=1}^{D} b_{a}(j,i)a^{j-1}. \hspace{1cm} (11)$$

Let $L$ denote the set $L=\{0,1,\ldots,a-1\}$. For $J \subseteq I_{D}$ and $|J|=J'$ we define
Example MUSS

Parameters: \( D=2 \quad N_1=N_2=N=4 \)

Set of enciphering transformations: \( T = \{ g \in G(I_4)^2 \mid c(1,g) = c(2,g) \} \)

Message alphabets: \( X_1 = X_2 = I_4 \quad X = I_4 \quad I_4 \quad Y = X \)

Defined sets:
\( X'_1 = X'_2 = \{1,4\} \)
\( X' = \{(1,1),(4,1),(1,4),(4,4)\} \)
\( H'_1 = \{e,(1 \ 3),(2 \ 4),(1 \ 3)(2 \ 4)\} \)
\( H'_2 = \{e,(1 \ 2),(3 \ 4),(1 \ 2)(3 \ 4)\} \)

\( H_1 = T \cap H'_1 \)
\( H_2 = T \cap H'_2 \)

Coder mapping:

<table>
<thead>
<tr>
<th>( f )</th>
<th>( X \rightarrow Y )</th>
<th>Residue classes generated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>((x_1 x_2) = x \in X )</td>
<td>((y_1 y_2) = y \in Y )</td>
<td>( T )</td>
</tr>
<tr>
<td>1 1</td>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>4 1</td>
<td>2 2</td>
<td>1</td>
</tr>
<tr>
<td>1 4</td>
<td>3 3</td>
<td>1</td>
</tr>
<tr>
<td>4 4</td>
<td>4 4</td>
<td>1</td>
</tr>
<tr>
<td>3 1</td>
<td>1 2</td>
<td>2</td>
</tr>
<tr>
<td>2 1</td>
<td>2 1</td>
<td>2</td>
</tr>
<tr>
<td>3 4</td>
<td>3 4</td>
<td>2</td>
</tr>
<tr>
<td>2 4</td>
<td>4 3</td>
<td>2</td>
</tr>
<tr>
<td>1 2</td>
<td>3 1</td>
<td>2</td>
</tr>
<tr>
<td>2 2</td>
<td>4 1</td>
<td>2</td>
</tr>
<tr>
<td>3 2</td>
<td>3 2</td>
<td>2</td>
</tr>
<tr>
<td>4 2</td>
<td>4 2</td>
<td>2</td>
</tr>
<tr>
<td>1 3</td>
<td>1 3</td>
<td>2</td>
</tr>
<tr>
<td>2 3</td>
<td>2 3</td>
<td>2</td>
</tr>
<tr>
<td>3 3</td>
<td>1 4</td>
<td>2</td>
</tr>
<tr>
<td>4 3</td>
<td>2 4</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 3. Example of a coder mapping and some sets defined in the discussion in section 5. Observe that the transformations in \( H'_2 \) are represented as permutations given as the product of disjoint cycles. \( e \) represents the identity permutation.
A(J,w) = \{i \in I_N | b_a(j,i) = c(j,w), j \in J \} \text{ for } w \in L^J' \tag{12}

The coder mapping is defined by the rule that \(x=(x_1, x_2, \ldots, x_D) \in X\) is mapped on the element \(y=(y_1, y_2, \ldots, y_D) \in Y\) which have

\[ y_i = \bigcap_{j \in I_D} A(j, b_a(i, x_j)). \tag{13} \]

To see that the intersection in (13) always will result in a set containing a single element recall the definition of \(A(j,w)\). It says that \(A(j,w)\) contains all numbers \(i \in I_N\) that gives a representation of \(i-1\) in base \(a\) which have the \(j:\text{th}\) digit equal to \(w\). The intersection is taken over \(j \in I_D\), hence all digits in a number in base \(a\) are specified and we have shown our point. The explanation also reveals that

\[ y_i = 1 + \sum_{j \in I_D} b_a(i, x_j) a^{j-1}. \tag{14} \]

A very illustrative way of imagining this mapping is to take a \(D \times D\) matrix with the element in row \(i\) and column \(j\) equal to \(b_a(i, x_j)\). Then the \(j:\text{th}\) column represents \(x_j-1\) and the \(i:\text{th}\) row represents \(y_i-1\) in base \(a\). This picture shows both that the mapping is bijective and that it is its own inverse.

Thus, we end up with a very simple coder. What characteristic properties will this coder have when it is used together with a set of enciphering transformations belonging to the class defined above. To answer this question in terms of DNU and DKU values we have to analyze what happens when a wiretapper intercepts a cryptogram sequence and knows some of the corresponding message sequences.

Let \(J_2\) contain the indices of the sources the wiretapper knows and let \(J_1\) contain the indices of the sources that the wiretapper wants to decrypt. The number of elements in \(J_1\) and \(J_2\) are denoted \(J_1\) and \(J_2\) respectively. Then define the groups
\[ F(J_1) = \bigotimes_{w \in L_{J_1}} G(A(J_1, w)) \]  \hspace{1cm} (15)

and

\[ H(J_1) = T \cap F(J_1)^D. \]  \hspace{1cm} (16)

The number of elements in \( A(J_1, w) \) is a \( D-J_1 \) and this gives

\[ |H(J_1)| = [(a_{-1})!]^{D'\alpha_{J_1}}. \]  \hspace{1cm} (17)

Now assume that the wiretapper has intercepted a crypto-gram sequence \( z^L \) and that he knows the corresponding message sequences \( x^L_2 \). He can then determine a transformation \( t \) such that \( c(J_2, rz^L_1) = x^L_{J_2} \). From the definition of the con-\[ d \] er mapping it follows that all elements \( x \in X \) with \( c(J_2, x) = x_{J_2} \) are mapped on \( A(J_2, w_1) \times A(J_2, w_2) \times \ldots \times A(J_2, w_D) \) for a certain set of \( w_1 \in L_{J_2} \). Hence \( H(J_2) \) contains all the transformations in \( T \) for which \( c(J_2, htx^L_1) = x^L_{J_2} \). A similar argument shows that \( c(J_1, rhtz^L_1) = c(J_1, rz^L_1) = x^L_{J_1} \) if and only if \( h \in H(J_1) \). And this implies that if \( h_1, h_2 \in H(J_2) \) belong to the same coset generated by \( H(J_1) \cup J_2 \) we have \( c(J_1 \cup J_2, rh_1tz^L_1) = c(J_1 \cup J_2, rh_2tz^L_1) \). Hence the number of equivalence classes in the set of possible deciphering keys equals the number of cosets in \( H(J_2) \) generated by \( H(J_1 \cup J_2) \). Thus, we get the following expression for the DKU's:

\[ v_k (J_1 | J_2) = \frac{|H(J_2)|}{|H(J_1 \cup J_2)|} = \frac{(a_{-1})!^{D'\alpha_{J_2}}}{D-J_2} \]  \hspace{1cm} (18)

With the convention that \( |\emptyset| = 0 \) and \( H(\emptyset) = T \), (18) is the correct expression for all the DKU's.
To find the values of the DMU's we assume that the wiretapper has intercepted an arbitrary sequence $z_j^L$, knows the corresponding sequence $x_j^L$ and has found the set $E$ of all transformations for which $c(j^2,rez^L)=x_j^L$ when $e\in E$. With reference to the argument above we see that $h\in H(j^2)$ implies $h\in E$ for all $e\in E$. Let $y^L=ez^L$ for an arbitrary $e\in E$. Then the wiretapper has at least the set of messages which is mapped on $C=\{hy^L|h\in H(j^2)\}$ to chose between. Among these messages there may be some that have the same $x_j^L$ sequence. Our problem is to find an expression giving the least number of different sequences $x_j^L$ that can be mapped on a set $C$. We do this by first lower bounding this number when $L=1$. Then we give an example for which the lower bound is achieved. This will show that the lower bound gives the desired minimum because the lower bound is by the construction of $C$ applicable on each component in a sequence $y^L$.

Now the lower bound. With $L=1$, $C$ is a residue class in $Y$ generated by $H(j^2)$. From the definition of the coding mapping we see that all messages $x\in X$ with $c(j^2,x)=x_j^L$ will be mapped into a subspace $Y'=\cup Y$ which is defined by $Y'=A(j^2,w_1^L)\times A(j^2,w_2^L)\times \ldots A(j^2,w_D^L)$ for a certain set of $w_i\in L^1$, $i\in D^1$. In the same way the set of messages $x\in X$ with $c(j^2,x)=x_j^L$ and $c(j^1,x)=x_j^L$ are mapped into a subspace of the form $Y''=A(j^2\cup j_1^L,w_1^L)\times A(j^2\cup j_1^L,w_2^L)\times \ldots A(j^2\cup j_1^L,w_D^L)$ where $w_i\in L^1+j_1^L$. $Y''$ is of course a subset of $Y'$ and $C$ is also a subset of $Y'$. We shall show that $C$ contains elements in at least a $D^1\cdot J^1$ different subsets of same type as $Y''$ in $Y'$. This will prove that the number of different sequences $x_j^L$ in $C$ is at least a $D^1\cdot J^1$.

A set $A(j^2,w_i)$ has a $D^1\cdot J^2$ elements and a set $A(j^2\cup j_1^L,w_i^L)$ has a $D^1\cdot J_1^L$ elements. Hence $A(j^2,w_i)$ is the union of a $J^1_1$ disjoint sets $A(j_1^L\cup j_2^L,w_i^L)$. Thus $Y'$ is the union of a $D^1\cdot J_1^L$.
different sets of the same type as \( Y' \). Two elements \( y_1, y_2 \in C \) will represent different messages \( x_{j1} \) when \( c(i, y_1) \) and \( c(i, y_2) \) belong to different sets \( A(J_2 \cup J_1, w_i') \) for at least one \( i \in I_D \).

From the definition of \( H(J_2) \) it is seen that for every element \( y \in C \), an arbitrary choice of \( a \in A(J_2, w_i') \) and \( i \in I_D \) there exists a \( h \in H(J_2) \) such that \( c(i, hy) = a \). Then recall the definition of \( T \). It states that the transformation applied on components in \( y \) given by \( U_i \) can be chosen independently for the different sets \( U_i \). Let \( V \) be a set containing a single element from each set \( U_i \), \( i \in I_D \); then \( |V| = D' \). With reference to what is stated above we conclude that for each \( y \in C \) it is always possible to choose a \( h \in H(J_2) \) in such a way that \( c(V, hy) \) is equal to any element in \( A(J_2', w_{i1}) \times \ldots \times A(J_2', w_{iD'}) \), \( i_1, \ldots, i_{D'} \in V \). Thus, \( C \) will always contain elements in at least a \( D'J_1 \) different sets of type \( Y' \) and we have proven our lower bound.

To demonstrate an element in \( Y \) which belongs to a residue class that achieves this lower bound we chose a \( y \) with all components equal. As all components in \( y \) given by \( U_i \) will be transformed in the same way, we see that there will only exist a \( D'J_1 \) different messages \( x_{j1} \) associated with this residue class. And this concludes the derivation of

\[
\nu_m(J_1 | J_2) = \frac{D'J_1}{1}.
\]  

(19)

Now we want to show that there is a MUSS in the class defined in this section that in some respects may be held as better than a system that encrypts the different message sources separately and then time multiplexes them. The system we compare with is a MUSS with the identity mapping used as coder mapping. The comparison is made when both systems have \( T \) defined by (10) and \( D' = D \).
The DMU's and the DKU's of the time multiplexing system are given by

\[ v'_m(J_1 | J_2) = a^{DJ_1} \]  

(20)

\[ v'_k(J_1 | J_2) = (a^{D!})^{J_1} \]  

(21)

where, as in the following, \( J_1 = |J_1| \) and \( J_2 = |J_2| \).

We compare the systems by comparing their DMU's and DKU's. The first observation is that \( v'_m(J_1 | J_2) = v'_m(J_1 | J_2) \) for all possible pairs of \( J_1 \) and \( J_2 \). Hence the systems are equal in this respect. To compare the DKU's we calculate their logarithmic ratio. We have to distinguish between two cases. The first is when \( J_1 + J_2 = D \) and the second is when \( J_1 + J_2 < D \). For \( J_1 + J_2 = D \) we have

\[ r_k(J_1 | J_2) = \log \frac{v_k(J_1 | J_2)}{v'_k(J_1 | J_2)} = Da^2 \log((a^{D-J_2}!)) - J_1 \log(a^{D!}) \]  

(22)

To see what (22) means when the parameters \( a \) and \( D \) are varied we use Stirlings formula to approximate the factorials. Then we get

\[ r_k(J_1 | J_2) \approx -J_2 a^D + \frac{1}{2}D[\log(2\pi) + (D-J_2)\log(a)]a^{J_2} - \frac{1}{2}(D-J_2)\log(2\pi) - \frac{1}{2}D(D-J_2)\log(a). \]  

(23)

The dominating term in (23) is \( -J_2 a^D \) which implies that \( r_k(J_1 | J_2) \) is less than zero when \( a \) and \( D \) are sufficiently large. Hence, when the wiretapper knows all other message sequences than those he tries to decrypt, he would have a smaller number of keys to chose between in our system than in the time multiplex system.
With $J_1 + J_2 < D$ the log-ratio is approximately given by

$$r_k(J_1 | J_2) \approx J_1 a^D - \frac{1}{2} D \left[ \log(2\pi) + (D - J_1 - J_2) \log(a) \right] a^{J_1 + J_2} +$$

$$+ \frac{1}{2} D \left[ \log(2\pi) + (D - J_2) \log(a) \right] a^{J_2} -$$

$$- \frac{1}{2} J_1 \left[ \log(2\pi) + D \log(a) \right]$$

(24)

The dominating term is $J_1 a^D$ and by suitable choice of $a$ and $D$ we can make $r_k(J_1 | J_2) > 0$ for all possible $J_1$ and $J_2$. This means that our MUSS can be designed in such a way that when the wiretapper knows all but one of all other message sequences than those he tries to decrypt, he will have more keys to chose among in our system than in the time multiplex system.
7. CONCLUDING REMARKS

The examples of MUSS's presented in sections 5 and 6 shows that the defined model of a MUSS can be used in the analysis of relevant problems. We have used the DMU's and DKU's to evaluate system performance. The evaluation of different equivocations are left as an open problem. A preliminary analysis of this problem for the class of MUSS's defined in section 6 shows that even with the assumption that the message sources are memoryless it will be very hard to find any bounds.

The technique used in the design of the coder mapping described in section 5 relied on forming subgroups in T with certain properties. A rewarding but probably very difficult area for further research would be to analyze the structure of groups that can be used together with this method of design.
ACKNOWLEDGEMENT

The original idea of a MUSS is due to professor Ingemar Ingemarsson and the author would like to thank him for all the valuable discussions and comments on this subject.
REFERENCES


