Recall that the rate R^1 computed in [1] is bounded by

$$R^1 < \sup_{\substack{t>0 \\ 0<p<1}} [-t \cdot \log e - (1-d)\log((1-q)A + qB)]$$

(see (1)). Let $t^* > 0$ be such that for fixed d, p, $R^1(t^*)$ is maximized. Then for all $0 \leq q \leq 1, A^q \cdot B^{1-q} \leq qA + (1-q)B$ by convexity. Hence, we conclude that

$$R \geq R(t^*) \geq R^1(t^*).$$

In fact, the optimization of (14) for fixed d, p has a closed form since it results in a quadratic equation in t (similar to (1)). □

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for valuable comments and pointers to relevant prior work.

REFERENCES

[1] S. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1226–1237, Mar. 2006.

[2] E. Drinea and M. Mitzenmacher, "On lower bounds for the capacity of deletion channels," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 227.

[3] E. Drinea and M. Mitzenmacher, "Improved lower bounds for i.i.d. deletion channels," in *Proc. 42nd Annu. Allerton Conf. Communication, Control and Computing*, Monticello, IL, Oct. 2004.

[4] R. L. Dobrushin, "Shannon's theorems for channels with synchronization errors," *Probl. Inf. Transm.*, vol. 3, no. 4, pp. 11–26, 1967, Translated from *Probl. Pered. Inf.*, vol.3, no. 4, pp. 18–36, 1967.

[5] A. S. Dolgopolov, "Capacity bounds for a channel with synchronization errors," *Probl. Inf. Transm.*, vol. 26, no. 2, pp. 111–120, 1990, Translated from *Probl. Pered. Inform.*, vol. 26, no. 2, pp. 27–37, Apr./Jun. 1990.

[6] W. Feller, *An Introduction to Probability Theory and its Applications*, 2nd ed. New York: Wiley, 1971, vol. 2.

[7] A. Kavčić and R. Motwani, "Insertion/deletion channels: Reduced-state lower bounds on channel capacities," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 229.

[8] V. V. Petrov, *Limit Theorems of Probability Theory*. Oxford, U.K.: Clarendon, 1995.

[9] J. D. Ullman, "On the capabilities of codes to correct synchronization errors," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 95–105, Jan. 1967.

[10] N. D. Vvedenskaya and R. L. Dobrushin, "The computation on a computer of the channel capacity of a line with symbol drop-out," *Probl. Inf. Transm.*, vol. 4, no. 3, pp. 76–79, 1968, Translated from *Probl. Pered. Inform.*, vol. 4, pp. 92–95, 1968..

A Simple Lower Bound for the Capacity of the Deletion Channel

Michael Mitzenmacher, *Member, IEEE*, and Eleni Drinea

***Abstract*—We present a simple proof that the capacity of the binary independent and identically distributed (i.i.d.) deletion channel, where each bit is deleted independently with probability d, is at least $(1-d)/9$, by developing a correspondence between the deletion channel and an insertion/deletion channel that we call a *Poisson-repeat channel*.**

***Index Terms*—Binary deletion channel, channel capacity, insertion and/or deletion channels.**

I. INTRODUCTION

In this work, we consider a natural correspondence between the binary independent and identically distributed (i.i.d.) deletion channel (referred to henceforth simply as the *deletion channel*), where a fixed number of bits n are transmitted and each is deleted independently with probability d, and a simple insertion/deletion channel that we call a *Poisson-repeat channel*. Based on this correspondence, we are able to conclude that the capacity of the deletion channel in bits, which we denote here by C_d, is at least $0.1185 \cdot (1-d)$ for every $d, 0 < d < 1$. We prefer to write this in the simpler form

$$C_d \geq (1-d)/9$$

to emphasize that this bound is within a constant factor of the trivial upper bound on the capacity of $(1-d)$ (based on the capacity of the binary erasure channel) for all d. As far as we can tell, no previous work has given a capacity lower bound that is within a fixed constant factor of $(1-d)$. Our approach also naturally generalizes to larger alphabets, but for this work we focus on the binary case.

The deletion channel has been the subject of recent study. The best lower bounds known for the capacity arise from an argument of Drinea and Mitzenmacher [2], [3], which we apply here to lower-bound the capacity of the Poisson-repeat channel. For deletion channels with larger alphabets, the work of Diggavi and Grossglauser [1] gives the best general capacity bounds. For more information and background, see [2], [3].

II. THE POISSON-REPEAT CHANNEL

We define a Poisson-repeat channel with parameter λ as follows: the input is a binary string of length n. As each bit passes through the channel, it is replaced by a discrete Poisson number of copies of that bit, where the number of copies has mean λ and is independent for each bit. Notice that some bits will be replaced by 0 copies. The receiver obtains the concatenation of the bits output by the channel.

We use basic facts about the Poisson distribution that can be found in standard texts (see, e.g., [4]). For example, the sum of a constant number of independent random variables with a Poisson distribution also has a Poisson distribution; similarly, if we have a number of items

Manuscript received February 14, 2006; revised June 13, 2006. The work of E. Drinea is supported by the National Science Foundation under Grant CCR-0118701. The work of M. Mitzenmacher is supported by the National Science Foundation under Grants CCR-9983832, CCR-0118701, CCR-0121154, and an Alfred P. Sloan Research fellowship.

The authors are with the Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: michaelm@eecs.harvard.edu; edrinea@deas.harvard.edu).

Communicated by R. J. McEliece, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2006.881844

X given by a Poisson distribution with mean λ, and each item is independently deleted with probability d, the number of remaining items Y is given by a Poisson distribution with mean $\lambda(1-d)$.

III. The Poisson-Repeat Channel and the Deletion Channel

We present two ways of deriving a correspondence between the deletion channel and the Poisson-repeat channel. The first gives a *deterministic* correspondence that provides useful intuition. The second gives a *randomized* correspondence that makes the analysis much easier and cleaner for deriving our result. These correspondences are based on the following idea: transmitting a codeword through a Poisson-repeat channel can be mimicked by a deletion channel by repeating the codeword bits an appropriate number of times.

Let $s = 1-d$ be the probability a bit successfully arrives. For our deterministic correspondence, temporarily assume that $1/s$ is an integer. A natural idea is to take a message that we want to send and replace each bit with $1/s$ copies of the bit, because then the expected number of copies that make it through the deletion channel is 1. If exactly one copy of each bit did make it through the channel, we would obtain the original message. Of course, the deletion channel is not so kind. Instead, each successive block of $1/s$ bits will yield $B(1/s, s)$ bits at the receiver, where $B(t,p)$ is a binomial random variable corresponding to t trials with success probability p. As s goes to 0, $B(1/s,s)$ converges to a Poisson random variable with mean l. That is, each bit in the original message yields a number of copies that is approximately distributed as a Poisson random variable.

This naturally gives a connection to the Poisson-repeat channel. Suppose that we have a codebook and a decoding scheme for the Poisson-repeat channel, with codewords of length $m = ns$. A corresponding codebook for the deletion channel can be obtained by taking each codeword for the Poisson-repeat channel and replacing each bit with $1/s$ copies. Now, after passing a derived codeword through the deletion channel, each bit in the original codebook yields a number of copies that is approximately Poisson distributed with mean 1, and hence we can simply apply the decoding scheme for the Poisson-repeat channel to determine the original codeword from the codebook for the Poisson-repeat channel. If we have a lower bound L_1 on the capacity of the Poisson-repeat channel with parameter 1, we should obtain a lower bound of approximately $L_1 s = L_1(1-d)$ for the deletion channel. When $1/s$ is not an integer we may use blocks of $\lceil 1/s \rceil$ bits; this just slightly changes the mean associated with the Poisson-repeat channel. While this is a reasonable and functional approach that can be made to yield an appropriate asymptotic result, it requires some care in dealing with the convergence issues to handle the asymptotics appropriately.

Armed with this intuition, we can greatly simplify the argument by considering a somewhat less natural but quite useful randomized correspondence. Again, we consider a codebook and decoding scheme for the Poisson-repeat channel with m-bit inputs. However, instead of replacing each bit in a codeword with *exactly* $1/s$ copies of the same bit to obtain a codeword for the deletion channel, we now independently replace each bit with a *random* number of copies, according to a Poisson random variable with mean $1/s$. As a result, the number of copies of a bit that arrive at the receiver has exactly a Poisson distribution with mean 1, and the number of copies is independent for each bit. Therefore, we can directly apply the decoding algorithm for the Poisson-repeat channel when decoding to determine the original codeword from the codebook for the Poisson-repeat channel. A visualization of this correspondence is given in Fig. 1.

The one potential problem with this correspondence is that now the number of bits to be transmitted over the deletion channel is not fixed, but random, potentially violating our definition for the deletion channel. The sender therefore determines what to send by performing the replacements as described above, but only sends the resulting sequence through the deletion channel if it has exactly $n = \lceil m/s \rceil$ bits; otherwise, the replacement procedure is repeated until n bits are obtained. A simple application of Stirling's formula (again, see [4]) shows that the probability that n bits are obtained on each trial is $\Theta(1/\sqrt{n})$. Specifically, letting X be the number of bits obtained, we have

$$\begin{aligned}\mathbf{Pr}(X=n) &= \frac{\mathrm{e}^{-m/s}(m/s)^n}{n!} \\ &= \frac{\mathrm{e}^{-m/s}(m/s)^n}{\sqrt{2\pi n} n^n \mathrm{e}^{-n}(1+o(1))} \\ &= \frac{\mathrm{e}^{n-m/s}(m/ns)^n}{\sqrt{2\pi n}}(1-o(1))\end{aligned}$$

which is $\Theta(1/\sqrt{n})$ when $n = \lceil m/s \rceil$. Hence, on average the replacement procedure would have to be repeated $\Theta(\sqrt{n})$ times.

Because the length of the string sent through the deletion channel is now fixed to be n, the numbers of copies of each bit obtained by the receiver are not independent; there is some very slight dependence. To see that this dependence does not substantially change the error probability of the decoding, we handle this dependence explicitly. Suppose that we could provide a variable number of input bits to the deletion channel, so the replacement followed by the passage through the deletion channel exactly mimicked the Poisson-repeat channel. Let X be the resulting input length. Also, let $\mathcal{F}$ be the event that the decoder fails to decode successfully for the Poisson-repeat channel. Then the probability of failure for the deletion channel with the restriction on the input length is just

$$\begin{aligned}\mathbf{Pr}(\mathcal{F} \mid X=n) &= \frac{\mathbf{Pr}(\mathcal{F} \cap (X=n))}{\mathbf{Pr}(X=n)} \\ &\leq \frac{\mathbf{Pr}(\mathcal{F})}{\mathbf{Pr}(X=n)}.\end{aligned}$$

As we have already stated, $\mathbf{Pr}(X=n) = \Theta(1/\sqrt{n})$, so as long as $\mathbf{Pr}(\mathcal{F})$ is super-polynomially small in n, the decoding is successful with high probability even when the input length for the deletion channel is fixed. With this caveat, we find that $C_d \geq L_1 s = L_1(1-d)$.

Although this argument gives a randomized process for determining what to send through the deletion channel, via standard arguments, it implies the existence of a fixed set of codewords of length n for the deletion channel that could be used in place of the random generating process.

Notice that by replacing each bit in the codeword with a random number of copies that is distributed according to a Poisson distribution

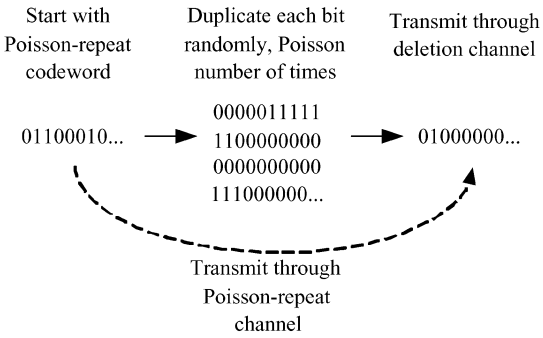

Fig. 1. We expand a Poisson-repeat codeword by replacing each bit with a Poisson-distributed number of copies with mean $1/(1-d)$ to obtain a codeword for the deletion channel. Transferring this derived codeword through a deletion channel with deletion probability d is equivalent to transferring the original codeword through a Poisson-repeat channel with parameter 1; this equivalence yields the capacity lower bound.

with mean λ/s for some constant λ, if we find a lower bound L_λ on the capacity of a Poisson-repeat channel with parameter λ, we can repeat the argument above to obtain a lower bound of $C_d \geq L_\lambda(1-d)/\lambda$.

To conclude, we formalize our results with the following statement, which follows from our discussion.

Theorem 1: Suppose that for some constant λ there exists a family of codes for the Poisson-repeat channel with parameter λ such that on m-bit inputs, the corresponding code has $c(m)$ codewords and error probability $f(m)$. Then for any constant d with $0 < d < 1$, there exists a family of codes for the deletion channel with deletion probability d such that on $\lceil \lambda m/(1-d) \rceil$-bit inputs, the corresponding code has $O(c(m))$ codewords and error probability $O(\sqrt{m} f(m))$.

IV. The Capacity of the Poisson-Repeat Channel

While it seems obvious that the Poisson-repeat channel with parameter 1 has a capacity bounded away from 0, until recently there has not been a general approach for obtaining provable lower bounds on the capacity of specific insertion/deletion channels. We derive a lower bound using the "jigsaw-puzzle" decoding approach for insertion/deletion channels described in [3]. This approach gives a lower bound for capacity in terms of a summation that can be evaluated numerically. Error bounds derived via this approach show that the probability of decoding error is super-polynomially small in n, as required by the argument of Section III. We emphasize that jigsaw-puzzle decoding is essentially just a variation of the standard Shannon-style argument; the challenge is in determining the appropriate typical codewords and the typical received sequences for each codeword for this type of channel.

We describe how we utilize the approach of [3] in the Appendix, and provide just a brief overview here. We consider random codebooks, where codewords are generated by laying out successive alternating blocks of zeroes and ones, until the desired codeword length n is reached. The length of each block is independently determined by some fixed distribution P, satisfying some minimal properties. We consider channels where each bit is independently replaced by a random number of copies of the same bit, with the number of copies given by a distribution G, again satisfying some minimal properties. We note that it is sufficient, for example, for G to have a tail that decreases at least geometrically fast, and the Poisson distribution has this property. Zero copies of a bit corresponds to a deletion. The Poisson-repeat channel fits this channel model.

Given P and G, there is an expression that gives a lower bound for the capacity of the channel; this expression is complex, but can be evaluated numerically. It is not clear how to optimize the choice for P; however, in previous work on deletion channels, codes based on random codebooks selected according to a first-order Markov chain, where each symbol is the same as the previous one with probability p, have performed well. Equivalently, codebooks are constructed by laying out alternating blocks of zeroes and ones, where the length of each block is geometrically distributed with mean $1/(1-p)$ for some p. (The case $p = 1/2$ corresponds to codewords chosen uniformly at random.) We have found the optimal value of p to two decimal places to maximize the resulting lower bound. Our current estimate gives $L_1 > 0.1171$ using $p = 0.87$.

We also used this approach to find the capacity for the Poisson-repeat channel with parameter λ for various λ, in order to find the value λ that maximizes the lower bound $L_\lambda(1-d)/\lambda$. The best lower bound on the ratio L_λ/λ that we have found is at $\lambda = 1.79$ using $p = 0.77$, with $L_{1.79}/1.79 > 0.1185$. Again, we prefer to more succinctly and conveniently say that $C_d \geq (1-d)/9$. Since insertions appear easier to handle than deletions (e.g., see the results in [3]), it is perhaps natural that a Poisson-repeat channel which on average yields slightly more than one copy of each transmitted bit can allow higher transmission rates.

It is worth contrasting this result with the bounds reported in [2], [3], which give the best current lower bounds on the capacity of the deletion channel for specific rates. While for low values of d the lower bound we have derived here is far from optimal, it is very close to the reported bound for large d, being slightly smaller than the reported bound for $d = 0.9$ and slightly larger than the reported bound for $d = 0.95$ in [3]. This is perhaps not surprising, since fundamentally the results use the same underlying approach. We emphasize what is interesting about this result, in contrast to [2], [3], is that here our argument gives a very simple expression for the capacity lower bound, good for all values of d, based on a reduction argument. The approach of [3], in contrast, gives a method for calculating a specific lower bound given a specific value of d. This calculation can be quite time-consuming as d approaches 1, as it involves considering bit sequences on the order of length $1/(1-d)$.

V. Conclusion

We have demonstrated a connection between deletion channels and Poisson-repeat channels, and specifically with the Poisson-repeat channel with parameter 1. We have used this connection to provide a capacity result, showing that the capacity of the deletion channel with parameter d is within a constant factor of $1-d$, the capacity of the erasure channel with parameter d. This connection could also be possibly used constructively to give an efficient algorithm: an algorithm for encoding and decoding on a Poisson-repeat channel would immediately give a corresponding algorithm for the deletion channel. We believe that the approach of mapping one channel to another via a randomized encoding may prove useful more generally for proving bounds on insertion/deletion channels.

It is interesting to consider whether our argument might be asymptotically tight. That is, perhaps there exists some constant λ such that the capacity C_λ of the Poisson-repeat channel with parameter λ and the capacity C_d of the deletion channel with deletion probability d satisfy

$$\lim_{d \to 1} C_d/(1-d) = C_\lambda/\lambda.$$

More generally, it would be interesting to obtain some insight into the behavior of the function $C_d/(1-d)$, both for the binary alphabet and for more general alphabets.

Appendix

We summarize the framework [2], [3], following the notation in these works, to explain our calculations to lower-bound the capacity of Poisson-repeat channels. The lower bounds are based on an analysis of random codebooks, where codewords are generated by laying out successive alternating blocks of zeroes and ones, until the desired codeword length n is reached. The length of each block is independently determined by some fixed distribution P, with P_j being the probability that the block has length j. For our result, we use block lengths that are geometrically distributed, or equivalently, our codewords are generated by a first-order Markov chain.

We may think of the received string as also consisting of alternating blocks of zeroes and ones, with block lengths being given by a distribution $\mathcal{P}$ depending on P, and $\mathcal{P}_k$ being the probability that a block has length k. A block of length k in the received sequence arises from a group of one or more blocks from the transmitted codeword. Specifically, a block S in the received sequence is associated with the following blocks in the transmitted codeword: the block the first bit of S was derived from, and all consecutive blocks up to but not including the block the first bit of the block after S was derived from. The ordered sequence of lengths of this group of blocks in the codeword is called the *type* of the block in the received sequence. For geometrically

distributed block lengths, types can naturally be grouped into families [3], where each type in a family occurs with the same probability. For all $i \geq 0, z \geq 1, r \geq i, s \geq i$, $F(i,z,r,s)$ is defined to be the family of types that consist of the following: $2i+1$ blocks, the first of which has length z; the lengths of the i blocks whose bits differ from the first block sum up to s; and the lengths of the remaining i blocks whose bits are the same as the first block sum up to r.

Let K and T be random variables representing the length and type of a block in the received string. Also, let $H(X)$ be the entropy of X measured in bits. Finally, following the notation of [3], note that if we let $\rho_{a,b}$ be the probability that a bits transmitted over a Poisson-repeat channel with parameter λ yield b bits of output, we have

$$\rho_{a,b} = \frac{e^{-\lambda a}(\lambda a)^b}{b!},$$

since the sum of a independent Poisson random variables with mean λ has a Poisson distribution with mean λa. The following lower bound derives immediately from Theorem 4 in [3] (simply by specifying for the Poisson-repeat channel):

Theorem 2: Consider a Poisson-repeat channel with parameter λ and a geometric distribution P with parameter p governing the creation of a random codebook. The capacity of this channel is lower bounded by

$$\sup_{0<p<1} \frac{1}{\frac{1+D}{1-D}\cdot\sum_z zP_z}\left[H(\mathcal{P}) + \sum_k \sum_F \sum_{t\in F} \mathbf{Pr}[T=t, K=k] \cdot \log\left[\frac{d^{r+s+z}\lambda^k}{k!}\cdot((r+z)^k - r^k)\right]\right] \quad (1)$$

for $d = e^{-\lambda}$, $D = \sum_z P_z d^z$, and F standing for $F(i,z,r,s)$.

Various simplifications can be made to this expression, particularly when choosing codewords governed by block lengths determined by a geometric distribution; the key, however, is that a lower bound for the expression can be calculated numerically. (Truncating the sum appropriately to make it finite will still yield lower bounds, as in [3].)

REFERENCES

[1] S. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1226–1237, Mar. 2006.

[2] E. Drinea and M. Mitzenmacher, "On lower bounds for the capacity of deletion channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4648–4657, Oct. 2006.

[3] ——, "Improved lower bounds for i.i.d. deletion and insertion channels," *IEEE Trans. Inf. Theory*, submitted for publication.

[4] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

How to Generate Cyclically Permutable Codes From Cyclic Codes

Minoru Kuribayashi, *Member, IEEE*, and Hatsukazu Tanaka, *Fellow, IEEE*

***Abstract*—On the basis of the characteristics of cyclic codes, the codeword space can be partitioned into small subspaces where cyclically shifted codewords of a particular codeword occupy the same subspace. A cyclically permutable code generates codewords belonging to each subspace. However, no approach for the efficient construction of cyclically permutable code from binary cyclic codes has been proposed thus far. In this study, we propose an approach for the efficient and systematic construction of a cyclically permutable code from a cyclic code by utilizing an algebraic property. The proposed coding method improves the robustness of watermarking, particularly for video frames, against a clipping attack.**

***Index Terms*—Cyclically permutable code, cyclic shift invariant, watermark, clipping attack.**

I. INTRODUCTION

A cyclic code [1] is a block code in which a cyclic shift of every codeword yields another codeword belonging to the same code. In this correspondence, a linear cyclic code, which is both linear and closed under cyclic shifting, is employed. Gilbert [2] defined cyclically permutable code (CPC) as a binary block code of block length n such that each codeword has a cyclic order n and the codewords are cyclically distinct (i.e., the same codeword cannot be obtained by cyclically shifting another codeword once more than once). In this scheme, large sets of cyclically permutable codewords are created by interleaving the cyclic shifts of several shorter words and selecting cyclically inequivalent subsets from the resulting set. Maracle and Wolverton [3] proposed an efficient algorithm for generating these cyclically inequivalent subsets. However, in order to use this procedure, sets of cyclically inequivalent placement vectors which indicate the positions of shorter words being interleaved should be selected; consequently, the codeword space is not exploited efficiently. In [4], the correspondence between $m \times n$ arrays and N-tuples, where $N = mn$, is utilized to construct binary constant-weight cyclic codes. This method, when combined with a simple method that selects a large subset of codewords with a full cyclic order, is used to construct a constant-weight CPC. In [5], on the basis of the combinatorial design of a difference family, several constructions for constant-weight CPC are presented. Further, by modifying the recursive constructions for the difference families, other constructions for constant-weight CPC are shown. Although such codes are applied in code-divisible multiple-access (CDMA) communication systems, their procedure for CPC construction is fairly complicated; further, their error correction capability is not discussed. To the best of our knowledge, no approach for efficient and systematic CPC generation directly from binary cyclic codes has been proposed thus far.

Based on the characteristics of cyclic codes, the existence of cyclically equivalent sets can be intuitively determined. Inaba and Nakahara [6] proposed an encoding procedure for obtaining cyclically inequivalent subsets from a cyclic code. However, the scheme only shows an example of the procedure, and the generated code is not a CPC because the cyclic order may be a divisor of the code length.

Manuscript received March 29, 2005; revised June 23, 2006.

M. Kuribayashi is with the Faculty of Engineering, Kobe University, Kobe, 657-8501, Japan (e-mail: kminoru@kobe-u.ac.jp).

H. Tanaka is with the Kobe Institute of Computing, Kobe 650-0001, Japan (e-mail: tanaka@kic.ac.jp).

Communicated by R. J. McEliece, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2006.881834