SIAM J. COMPUT.
Vol. 17, No. 2, April 1988

003

# PRIVACY AMPLIFICATION BY PUBLIC DISCUSSION*

CHARLES H. BENNETT†, GILLES BRASSARD‡ AND JEAN-MARC ROBERT§

**Abstract.** In this paper, we investigate how the use of a channel with perfect authenticity but no privacy can be used to repair the defects of a channel with imperfect privacy but no authenticity. More precisely, let us assume that Alice and Bob wish to agree on a secret random bit string, and have at their disposal an imperfect private channel and a perfect public channel. The private channel is imperfect in various ways: transmission errors can occur, and partial information can leak to an eavesdropper, Eve, who also has the power to suppress, inject, and modify transmissions arbitrarily. On the other hand, the public channel transmits information accurately, and these transmissions cannot be modified or suppressed by Eve, but their entire contents becomes known to her. We consider the situation in which a random bit string $x$ has already been transmitted from Alice to Bob over the private channel, and we describe interactive public channel protocols that allow them, with high probability: (1) to assess the extent to which the private channel transmission has been corrupted by tampering and channel noise; and (2) if this corruption is not too severe, to repair Bob's partial ignorance of the transmitted string and Eve's partial knowledge of it by distilling from the transmitted and received versions of the string another string, in general shorter than $x$, upon which Alice and Bob have perfect information, while Eve has nearly no information (or in some cases exactly none), except for its length. These protocols remain secure against unlimited computing power.

**Key words.** cryptography, error-correcting codes, information theory, key exchange, privacy, randomness, universal hashing, $t$-resilient functions, wiretap channel

**AMS(MOS) subject classifications.** 94A60, 94A40

## 1. Introduction.

Alice and Bob wish to agree on a secret random bit string. In order to achieve this goal, they have at their disposal an imperfect private channel and an authenticated public channel. The private channel is imperfect in various ways: transmission errors can occur, and partial information can leak to Eve, the eavesdropper, who also can modify the transmissions arbitrarily, as explained below. The only limitation we impose on Eve is the knowledge by Alice and Bob of an upper bound on the amount of partial information that can leak to her when she eavesdrops on a private channel transmission.

We allow Eve to tamper arbitrarily with the private channel transmissions. For instance, she can suppress the transmission of selected bits, perhaps to replace them with bits of her choice, to inject new bits, to toggle transmitted bits or to jumble them around. We allow her to introduce as much malicious noise as she wishes. In this paper, we granted Eve unlimited tampering power even though probably no real channel performs quite this badly, so that our results will hold true in any circumstance.

The quantum channel of [BB1], [BB2] is a prime example of an imperfect private channel, and this paper effectively allows the removal of its previous defects. Indeed,

* Received by the editors October 28, 1985; accepted for publication (in revised form) March 9, 1987. Part of this work was presented at CRYPTO 85 under the title, "How to Reduce your Enemy's Information."

† IBM T. J. Watson Research Laboratory, Yorktown Heights, New York 10598.

‡ Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, C.P. 6128, Succ. "A" Montréal, Québec, Canada H3C 3J7. The work of this author was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant A4107 and National Science Foundation grant MCS-8204506. Part of this research was conducted while the author was at the University of California, Berkeley, California 94720.

§ School of Computer Science, McGill University, 805 Sherbrooke St. West, Montréal, Québec, Canada H3A 2K6. This work was partially supported by Natural Sciences and Engineering Research Council of Canada grant A4107. This research was conducted while this author was at the Université de Montréal, Montréal, Québec, Canada H3C 3J7.

the quantum channel allows an eavesdropper to attempt reading a few bits of the transmission with a reasonable probability of not perturbing it, and hence of escaping detection. It also allows several types of blind tampering, such as toggling a selected bit (even if unable to read it) by passing the corresponding faint light pulse through an appropriate sugar solution. More classically, Diffie and Hellman's public-key distribution scheme [DH] can be thought of as an imperfect private "channel." Indeed, it efficiently leaks partial information on the (not-so-random) string exchanged, even if the discrete logarithm is hard to compute, because it is easy for an eavesdropper to determine whether the resulting secret is a quadratic residue or not.

On the other hand, the public channel transmits information accurately (possibly because it is supplemented by a classic error-correcting code [vL], [MS]), and these transmissions cannot be modified or suppressed by Eve, but their entire contents becomes known to her. Newspapers are an example of a secure public channel on which eavesdropping is easy but tampering nearly impossible. Such inherently authentic public channels are commonly suggested for disclosing public keys in a public-key cryptography/digital signature system such as [RSA]. If message authenticity is not thus enforced by the physical properties of the channel, it can be provided by an unconditionally secure authentication scheme such as that of [WC]. In this latter case, a small number of shared secret random bits must be known initially between Alice and Bob, and some of these are used up in the process of authentication; thus the net effect of our protocol in this case can be viewed as key expansion rather than key distribution. Computationally secure authentication [Br], [GGM] can also be used if protection against unlimited computing power is not sought. We shall assume throughout that Alice and Bob did not share initially any secret information, except perhaps for what is needed to implement this public channel authentication feature.

In this paper, we assume that some random bit string has already been transmitted from Alice to Bob over the private channel. We investigate authenticated public channel protocols that, with high probability, detect tampering and transmission errors. Subsequent protocols transform both strings in such a way as to eliminate most, and in some cases all, of Eve's information on the resulting string, except for its length. These public channel protocols remain secure against unlimited computing power, so that the entire exchange is as secure as the initial private channel transmission. It should be noted that excessive tampering on the private channel can result in suppressing communications between Alice and Bob, but it cannot fool them into thinking that they share a secret random string when in fact their strings are different or otherwise compromised.

Let us make our setting mathematically precise. Alice and Bob initially share no secret information. Alice chooses a random string $x$ of length $N$ and transmits it over the private channel. Independently, Eve chooses an eavesdropping function $e: \{0,1\}^N \to \{0,1\}^K$, where $K < N$, and a tampering function $t: \{0,1\}^N \times \{0,1\}^\omega \to \{0,1\}^N$. Alice and Bob know $K$ but otherwise nothing about $e$ or $t$. When Alice transmits $x$, Eve learns the value $e(x)$ and forwards the potentially corrupted value $y = t(x, R)$ to Bob, where $R$ is a random string representing channel noise. Notice that Eve does not herself learn the value of $t(x, R)$ nor can she influence the random string $R$ (although she may choose a function $t$ that does not take $R$ into consideration). Her information about $x$ from the private channel transmission consists *only* of knowing $e$ and $e(x)$; her information about $y$ consists only of knowing $e$, $e(x)$, and knowing the function $t$ that was applied to $x$ in order to obtain $y$. In this very hostile context, we show that Alice and Bob can *publicly* agree on a protocol that will allow them to ascertain whether $x = y$ (with an exponentially small error probability) and, if this is so, to end up with

a shorter shared string $z$ on which Eve has nearly no information or, in some cases, no information at all. If $x \neq y$, they can detect this with high probability and, if the differences are not too great, continue with the protocol.

To summarize, we investigate how the use of a channel with perfect authenticity but no privacy can be used to repair the defects of a channel with imperfect privacy but no authenticity. In § 2, we explain why classic error-correcting codes are inappropriate in this context. In § 3, we investigate how transmission errrors and tampering can be detected with high probability, and sometimes corrected, at the cost of leaking some information to Eve. In § 4, we investigate how Alice and Bob can subsequently reduce arbitrarily Eve's information at the cost of reducing slightly the length of their shared random string, assuming they have an a priori upper bound on the amount of information that Eve collected on the private channel. In § 5, we investigate the possibility of depriving Eve entirely of any information on the final shared random string at the cost of reducing its length more substantially.

Before we get started, let us give the following definition and some notation: if $i<j$, a function $f:\{0,1\}^j \to \{0,1\}^i$ is *equitable* if $\#\{x|f(x)=a\}=2^{j-i}$ for every binary string $a$ of length $i$. If $x$ and $y$ are bit strings of equal length, $x \oplus y$ denotes their bit-by-bit exclusive-or. Finally, if $x$ is a length $N$ bit string and if $0 \leqq K \leqq N$, $x$ **mod** $2^K$ denotes the length $K$ bit string consisting of the rightmost $K$ bits of $x$, and $x$ **div** $2^K$ denotes the length $N-K$ bit string obtained from $x$ by deleting its rightmost $K$ bits. We shall herein assume that the reader is familiar with the notions of information theory [G], [Mc], [S], universal hashing [CW], [WC], error-correcting codes [vL], [MS], and the theory of finite fields [Be].

**2. The inadequacy of classical error-correcting codes.** Let us recall that the imperfect private channel considered here is susceptible not only to random transmission errors, but also to any amount of controlled and malicious tampering. This tampering capability does not directly give Eve any additional information on $x$. It could, however, give her indirect information, because it may force Alice and Bob to subsequently discuss the situation over the public channel, as explained in § 3. The classic theory of error-correcting codes [vL], [MS] is not quite adequate for our purposes because it is based on the assumptions that few errors are more likely to occur than many, and that errors are not maliciously set by an opponent.

For instance, let $x$ and $y$ be Alice and Bob's strings, respectively, and let $N$ be their length. Eve's tampering ability enables her to actually select $x \oplus y$, barring actual transmission errors. This is clearly intolerable if error detection is attempted through a linear error-correcting code [MS]. Indeed, let $x$ be the private channel transmitted code word corresponding to Alice's chosen random string. Let $z$ be any code word chosen by Eve. If she perturbs the private channel transmission so that Bob receives $y = x \oplus z$, it will not be possible for him to detect tampering. Notice that Eve can achieve this without gaining any knowledge of the contents of the original transmission $x$.

Nonlinear error-correcting codes are not susceptible to the above threat, but they fall to an even simpler one because Eve is also capable of replacing Alice's bits by bits of her choice. If Alice sends some code word over the private channel, it suffices for Eve to suppress the original communications entirely, and inject any other code word of her choice instead. This simple-minded attack can be hindered by the post facto application of an error-correcting code, as discussed in § 3.2.

Classic error detection is therefore impossible in our context if Eve knows in advance of the private channel transmission which code is to be used. On the other

hand, we assumed that Alice and Bob did not initially share any secret, except perhaps to implement the authenticated public channel. A solution is that Alice randomly chooses an error-correcting code, produces the code word $x$ corresponding to her chosen random string, and sends $x$ to Bob through the private channel. She then waits for Bob to use the authenticated public channel in order to acknowledge receipt of the private transmission. Only at this point does Alice use the public channel to send Bob a description of the error-correcting code. This allows Bob to recover the original string and to check for errors and tampering. Thiš randomization approach allows Alice to reveal sensitive information to Bob, hence to Eve, only after it is already too late for her to efficiently alter the private channel transmission in an undetectable way.

Although such use of randomness is our main tool in this paper, it remains true that classic linear error-correcting codes are not appropriate because, even randomly chosen, they still assume few errors to be more likely than many. For instance, let us assume that Alice uses a Hamming code of dimension $[N, K]$ and that the random part of the protocol is the order in which she sends the code-word bits. It is no longer possible for Eve to toggle selected bits and to be certain to escape detection because she does not know which bits to toggle. However, there are exactly $\binom{N}{2}/3$ code words of Hamming weight 3 [MS], whereas there are $\binom{N}{3}$ length $N$ bit strings of weight 3. Therefore, Eve can toggle 3 random bits and escape detection with probability

$$\left(\binom{N}{2}\Big/3\right)\Big/\binom{N}{3}=(N-2)^{-1}.$$

Using such a protocol, Alice and Bob could only achieve a very high probability of not being fooled, say $1-2^{-50}$, at the cost of exchanging unreasonably long strings.

It is instructive to compare our setting with the problem solved by the wiretap channel of Wyner [W] which achieves similar results in a more classically information-theoretic setting. In Wyner's setting, Alice encodes information by a channel code of her choice. The output of her encoder is fed into two classic (discrete, memoryless) communication channels: the *main channel*, leading to the intended receiver Bob, and the *wiretap channel*, of lesser capacity than the main channel, leading to the eavesdropper. All participants know the channel code and the statistical properties of the two channels. Under these conditions, Wyner showed that by appropriate choice of the channel code, Alice can exploit the difference in capacity between the two channels communicate reliably with Bob while maintaining almost perfect secrecy from the eavesdropper.

In our setting, the users have an additional resource: the authenticated public channel. This allows them to cope with a more powerful eavesdropper. Our eavesdropper is more powerful in two ways, either of which would be fatal in Wyner's setting: she can tamper with Alice's communications as well as listen to them, and she eavesdrops by evaluating an arbitrary $N$-bit to $K$-bit function *of her choice*, unknown to Alice and Bob.

In § 3.1, we describe error-detection schemes such that the probability of undetected tampering and transmission errors is independent of the number and position of altered bits. Moreover, this probability can be exponentially small in the length of the strings transmitted. Although never fully appropriate for the detection of tampering, classic error-correcting codes remain interesting in order to correct actual transmission errors over the private channel, as we investigate in § 3.2.

**3. Detection and correction of transmission errors and tampering.** Let $x$ be some random bit string selected by Alice. Assume she transmits it directly through the imperfect private channel, and let $y$ be the string thus received by Bob. Let $N$ be the

length of both strings. The public channel protocols described in § 3.1 allow Alice and Bob to detect whenever $x \neq y$ with an arbitrarily small error probability, independently of how $y$ differs for $x$. Should $y$ be found to differ from $x$, the protocol of § 3.2 can be used to reconcile them with high probability. The reconciliation protocol can also be used *preventively*, before using an error-detection protocol from § 3.1, if $y$ is expected to be different from $x$ merely due to normal transmission errors. The fact that these protocols leak information to Eve about $x$ is considered in § 4.

**3.1. Error detection.** A very simple but impractical way of testing whether $x = y$ is for Alice to choose a random function $f: \{0, 1\}^N \to \{0, 1\}^K$, where $K$ is a security parameter. After the private channel transmission is completed, she sends $f(x)$ to Bob over the public channel, together with a complete description of the function $f$. Should Bob find out that $f(y) = f(x)$, this would be considered as strong evidence that $y = x$, the error probability being $2^{-K}$, independently of the length of the strings and how they might differ. On the other hand, should $f(y)$ be different from $f(x)$, Bob could report to Alice with certainty that he did not receive the correct string. Notice that the amount of information on $x$ leaking to Eve from this protocol depends only on the security parameter $K$, and not on the length $N$ of the strings (except of course for the fact that $K < N$). This would not be the case if a classic error-detecting code has been used. Unfortunately, this scheme cannot be used in practice because there are $2^{K2^N}$ different such functions, and therefore as many as $K2^N$ bits are typically needed to merely transmit a description of the randomly chosen function.

In some cases, it may be preferable for Alice to choose randomly the function $f$ among the set of *equitable* functions only. This can be done in theory (although not in practice when $N$ is large) by randomly selecting a permutation $\pi: \{0, 1\}^N \to \{0, 1\}^N$ and defining $f(x) = \pi(x)$ **mod** $2^K$ for each string $x$ of length $N$. Notice that this allows a (very slight) reduction of the probability of undetected transmission errors or tampering from $2^{-K}$ to $(2^{N-K} - 1)/(2^N - 1)$.

Universal hashing [CW] provides an efficient way to achieve the same goal. After the private channel transmission is completed, Alice randomly chooses a function $f: \{0, 1\}^N \to \{0, 1\}^K$ *among some universal$_2$ class of functions.* She then sends both $f(x)$ and a description of $f$ to Bob. Thanks to universal hashing, the description of $f$ can be transmitted efficiently this time. After computing $f(y)$, Bob checks whether it agrees with $f(x)$. If it does, a basic property of universal hashing allows them to assume that $x = y$, their probability of error being bounded again by $2^{-K}$.

We refer the reader to [CW], [WC] for definition and discussion of universal hashing. Several universal$_2$ and strongly universal$_2$ classes are described there. Let us only stress here that they are entirely reasonable in practice. For instance, it suffices to send about $2N$ bits over the public channel to describe a function $f: \{0, 1\}^N \to \{0, 1\}^K$ randomly selected among $H_1$ [CW] or $P$ (§ 4), once the private channel transmission is completed. Compare this with the unreasonable $K2^N$ bits needed to describe a random function! Moreover, the computation of $f(x)$ is very efficient. Therefore, universal$_2$ hashing provides all the advantages of truly random functions, but none of the inconveniences.

As we mentioned in § 2, it is crucial that the actual verification function be transmitted to Bob only after Alice has received confirmation through the public channel that the private channel transmission is completed. This deprives Eve of any strategy that would reduce her chances of being detected. For instance, if she knew in advance that the function $f(x)$ simply returns the last $K$ bits of $x$, she could arbitrarily tamper with the other $N - K$ bits without fear of detection.

It is interesting to compare our use of universal hashing with the classic use of this technique for message authentication [WC]. The use of hashing for authentication depends on randomly choosing a hash function and keeping it secret at least until after the message to be authenticated has been received. In our protocol for error detection, the hash function cannot be kept secret, but it must be chosen randomly *after* the message has been transmitted.

An alternative to universal hashing comes to mind: polyrandom collections [GGM]. However, universal hashing is more appropriate in this context because it offers security against unlimited computing power. It does not rely on unproved assumptions, and it can be computed more efficiently.

**3.2. Reconciliation of the strings.** Whether $f: \{0, 1\}^N \to \{0, 1\}^K$ is chosen as a purely random function or within some universal$_2$ class of functions, what should Alice and Bob do if they find that $f(x)$ differs from $f(y)$? Whether anything can be done to recover from this situation depends on how much $x$ differs from $y$. If the Hamming distance between $x$ and $y$ is relatively small, due to limited channel noise and/or limited tampering, we show here how Alice and Bob can find and correct the errors in $y$. Using subsequent protocols from § 4, Alice and Bob can then distill from $x$ and $y$ a shorter string on which Eve has nearly (and in some cases exactly) no information. On the other hand, if the distance between $x$ and $y$ is large, Alice and Bob will be able to discover this fact, but they will have no recourse but to discard $x$ and $y$ and begin the protocol all over again by transmitting a new random string through the private channel. As mentioned in the introduction, Eve can effectively prevent Alice and Bob from agreeing on a secret bit string by interfering strongly with every private channel transmission, but she cannot (except with low probability) fool them into thinking they have succeeded when in fact they have not.

Reconciliation is particularly easy if it is suspected that only one or two errors have occurred. Then Bob can try computing $f(z)$ on all strings differing from $y$ by only one or two bits, in the hope of finding a match for $f(x)$ and thus a likely candidate for $x$. We call this approach *bit twiddling.*

In the presence of more than a very few errors, bit twiddling becomes too time-consuming, but it is still practical to find and remove the errors by a post facto application of error-correcting codes, as described below. If many errors are expected even when no tampering occurs, the error detection protocol of § 3.1 should be deferred until after most or all of the errors have thus been found and corrected.

Many traditional error-correcting codes, such as Hamming codes, can be written in a *systematic* format, in which each code word consists of the original source word followed by a string of check bits. Given a source word $x$, the encoder thus generates the concatenation $xC(x)$, where $C(x)$ is some check string depending on $x$, and sends this longer string into the channel. At the receiving end, the redundancy in the code is used (if one is lucky) to recover the original $x$ despite the potential corruption of both $x$ and $C(x)$ by channel noise. Systematic codes have no special advantage over unsystematic ones (in which source and check information are mixed) for most ordinary error-correcting applications, but they are useful in the present setting because they allow the calculation of the check information $C(x)$ to be performed post facto, after the uncoded source data $x$ has been sent through the private channel. They also allow sending $x$ and $C(x)$ on different channels.

In § 2, it was pointed out that error-correcting codes used in the traditional non-post facto manner cannot defend against malicious tampering, because Eve, knowing the code, can escape detection by deliberately mutating one channel code word into

another. For example, in the case of a systematic code, if $xC(x)$ was sent by Alice on the channel, Eve could substitute $zC(z)$, even without learning anything about $x$, and Bob would be none the wiser. This threat can be avoided by applying the code in a post facto manner, and by taking two further precautions:

(i) The check string $C(x)$ generated by the code is not transmitted through the private channel, where Eve could alter it, but through the public channel, where she can only listen to it.

(ii) Before application of the error-correcting code, the strings $x$ and $y$ to be reconciled are subjected to a preliminary *uniformization transformation*, consisting of random permutation and complementation. The purpose of this transformation is to make the difference between $x$ and $y$ behave as if $y$ were the result of sending $x$ through a memoryless binary symmetric channel. To achieve this preliminary goal, Alice randomly chooses a permutation $\sigma:\{1, 2, \cdots, N\}\to\{1, 2, \cdots, N\}$ and a length $N$ bit string $w$. She then transmits both $w$ and a description of $\sigma$ to Bob over the public channel. Finally, Alice and Bob transform their strings $x$ and $y$, respectively, by first shuffling the bits according to $\sigma$, then taking the exclusive-or of the permuted string with $w$. It is clear that this leaves the number of errors unchanged, but redistributes them to random places not under the control of Eve, who is thus prevented from exploiting knowledge of the error-correcting code $C$ to introduce a pattern of errors against which the code would perform less well than average. It is also clear that the uniformization transformation releases no new information to Eve about $x$ and $y$.

To summarize, in order to use a systematic error-correcting code $C$ for reconciliation, Alice and Bob first uniformize their strings $x$ and $y$, thus producing $x_0$ and $y_0$. Alice then applies $C$ to $x_0$ and transmits the result $C(x_0)$ to Bob over the public channel, thereby giving away at most $|C(x_0)|$ bits of information to Eve about $x_0$. Bob uses $C(x_0)$ to correct the errors in $y_0$, thereby recovering $x_0$ if the code $C$ is sufficient to correct the errors that have occurred between $x_0$ and $y_0$. If $C(x_0)$ is significantly shorter than $x_0$, there is still some information about $x_0$ that Eve does not know, and the methods of § 4 can be used to nearly obliterate the information known to her, by allowing Alice and Bob to derive from $x_0$ a shorter string, of length approximately $|x_0|-|C(x_0)|$, about which Eve has less than one bit of information.

It remains to be decided what kind of systematic error-correcting code to use. If $\varepsilon$, the density of errors, is not too great to begin with, a simple block code (such as a Hamming code if we expect at most two errors per block) would suffice to reduce the number of errors to the point where they can be found and corrected by bit twiddling. If $\varepsilon$ is greater (say 0.01 or more), a systematic convolutional code [G] would be better, since these tree codes, unlike block codes, can achieve exponential error reduction without exponential decoding effort, while still transmitting data at a rate at least half the theoretical channel capacity. As emphasized above, whatever error-correcting code is used should be used in a post facto manner, being applied to the uniformized version of Alice's data, and the resulting check information $C(x_0)$ being sent to Bob over the noiseless, tamper-proof public channel.

In the traditional non-post facto situation, where both $x$ and $C(x)$ are sent through a binary symmetric channel with error probability $\varepsilon$, the capacity $|x|/(|x|+|C(x)|)$ achievable by convolutional codes with polynomial expected decoding effort is denoted Rcomp, and has the value $1-\log^{1}(1+2\sqrt{\varepsilon(1-\varepsilon)})$, which ranges between 0.5 and 1.0 times the theoretical capacity $1-\mathbf{H}(\varepsilon)$, where $\mathbf{H}(\varepsilon)=\varepsilon\log 1/\varepsilon+(1-\varepsilon)\log 1/(1-\varepsilon)$ is the *entropy function* [G]. In the present situation, the original data is transmitted

[1] Unless otherwise stated, all logarithms in this paper are to the base 2.

with error probability $\varepsilon$, but the check information is transmitted noiselessly; and capacity should thus be defined differently, as the maximum of $(|x_0|-|C(x_0)|)/|x_0|$, taken over all possible codes $C$, since this is the fraction of the original information in $x_0$ that remains secret after reconciliation. Nevertheless, in the post facto situation, Shannon showed that the theoretical capacity is still given by $1-\mathbf{H}(\varepsilon)$. By arguments parallel to the derivation of Rcomp [G], it can be shown that the effective capacity achievable by convolutional coding is still given by

$$\text{Rcomp} = 1-\log\,(1+2\sqrt{\varepsilon(1-\varepsilon)}).$$

In practice, the parameter $\varepsilon$ would not need to be known beforehand, since it can be determined interactively, by having Alice first compute a generous amount of check information, but then release only as much of this as Bob finds he needs for efficient decoding. A table comparing Rcomp with Shannon's theoretical capacity $c$ for some values of $\varepsilon$ follows.

| $\varepsilon$ | Rcomp | $c$ |
|---|---|---|
| 0.001 | 0.9116 | 0.9886 |
| 0.01 | 0.7382 | 0.9192 |
| 0.03 | 0.5765 | 0.8056 |
| 0.05 | 0.4781 | 0.7136 |
| 0.10 | 0.3219 | 0.5310 |
| 0.25 | 0.1000 | 0.1887 |
| 0.40 | 0.0146 | 0.0290 |

**4. Reduction of the eavesdropper's information.** Assuming that Alice and Bob agree on their strings as a result of one of the protocols discussed above, Eve has two different sources of information on that string: partial eavesdropping on the private channel, as the original random bit string was being transmitted, and complete eavesdropping on the public channel, as the agreement protocol was being carried out.

We now investigate how to reduce Eve's information arbitrarily close to zero at the cost of slightly shrinking the random bit string shared between Alice and Bob. In § 4.1, we assume that no eavesdropping on the private channel has occurred, but that transmission errors were possible. We also assume that the number of errors, if any, is small enough to be handled by bit twiddling (this assumption is removed in § 4.3). In § 4.2, we assume, on the contrary, that the eavesdropper has acquired partial information on the private channel transmission, but that tampering and transmission errors have not occurred. We finally consider in § 4.3 the more realistic case where both eavesdropping and arbitrary tampering on the private channel are possible, so the eavesdropper may gain information both directly from eavesdropping on the private transmission and indirectly by listening to the public channel reconciliation and error-detection protocols of § 3. All these protocols are secure against an eavesdropper with unlimited computing power.

**4.1. Eliminating the public channel eavesdropper's information.** Let us assume for the moment that Eve is unable to eavesdrop on the private channel but that transmission errors may have occurred. Assuming the number of errors is small enough to be handled by bit twiddling, we now show that Alice and Bob can agree on a secret random string on which Eve has *no* information, except for its length. In a companion paper [BBR], we investigate how to handle efficiently an error rate that would make bit twiddling ineffective. The interactive public discussion protocol discussed there allows Alice and

Bob to agree with high probability on a (shorter) secret random string on which Eve still has no information. Alternatively, § 4.3 of the present paper uses the post facto systematic convolutional codes of § 3.2 in order to handle normal transmission errors, but at the cost of leaking to Eve an arbitrarily small fraction of one bit of information about the final random bit string.

Let $x$ be the random string sent from Alice to Bob over the imperfect private channel. Let $y$ be the string received by Bob. An error detection protocol of § 3.1 is first applied to make sure, with high probability, that the random strings of Alice and Bob are identical. If not, bit twiddling is attempted by Bob to transform $y$ into $x$. If bit twiddling fails, either repeat the whole process (Alice sends a fresh new random string to Bob, etc.), use the interactive protocol of [BBR], or refer to § 4.3. At this point, assume that Alice and Bob agree with high probability on some length $N$ string $x$, but that Eve has gained information on this string by listening to the public channel error-detection protocol. We wish to eradicate this information of hers.

Let $f:\{0,1\}^N \to \{0,1\}^K$ be the function used in the error-detection protocol. Eve knows the $K$ bit value of $f(x)$, together with the function $f$ itself. Although this may not give her any physical bits of $x$, she has $K$ bits of information on $x$ in the sense of Shannon's information theory [S], assuming that $f$ is equitable. If $f$ is not equitable, Eve's expected information is less than $K$ bits, although she could know more occasionally. Her information can be characterized by the set $C = \{z \in \{0,1\}^N \mid f(z) = f(x)\}$ of possible candidates for $x$. From Eve's point of view, each element of $C$ is equally likely to be the string $x$ currently shared between Alice and Bob. Notice that Alice and Bob also have complete knowledge on the set $C$.

In order to eliminate Eve's information, Alice and Bob publicly agree on a function $g:\{0,1\}^N \to \{0,1\}^R$, for some integer $R \leqq N - K$, such that knowledge of the set $C$ yields no information on $g(x)$, the final string on which Alice and Bob agree. In other words, the purpose of this function $g$ is to shrink the string $x$ by at least $K$ bits, in order to compensate for the $K$ bits of information that Eve knows on $x$. It is clear that at least $K$ bits of $x$ must be sacrificed to privacy, but are $K$ bits enough in general? The proper choices of $R$ and of this information-reduction function $g$ depend on which error-detection function $f$ was chosen from § 3.1.

**4.1.1. Eliminating the information given by a truly random error-detection function.** Assume the error-detection function $f$ was chosen randomly from among all functions from $\{0,1\}^N$ to $\{0,1\}^K$. This error-detection protocol is of no practical interest, because it would require $K2^N$ bits to merely transmit function $f$. It is nonetheless instructive to figure out how the information given on $x$ by $f(x)$ can be eliminated in this context. Indeed, this provides a nice intuitive insight into the realm of information reduction. Moreover, it is interesting to compare what can be achieved in the truly random case with the practical world (§ 4.1.2). By analogy, recall Shannon's result that any channel can come arbitrarily close to achieving its theoretical transmission capacity through the use of random codes that cannot be implemented in practice [S], and that no practical codes known thus far can perform nearly as well. As we will see, such is not the case here. For this reason, we only state the main results pertaining to truly random functions and we refer the reader to [R] for the somewhat tedious proofs.

Recall that Eve's information about $x$ is characterized by the set $C$ of possible candidates, and that this set is also known for Alice and Bob. Let $\#C$ denote the number of elements in $C$. Let $y$ be the index of $x$ in $C$ when $C$ is ordered in lexicographic order. Then $y$ is available to both Alice and Bob, whereas it is just as likely to take any value between 1 and $\#C$ as far as Eve is concerned. If $\#C$ is large enough, such

a $y$ can thus be used as the resulting shared secret. The ideal situation occurs when the function $f$ is equitable. In this case, we always have $\#C = 2^{N-K}$ and thus $y$ is a uniformly distributed random bit string of length $N-K$ on which Eve has no information whatsoever save its length. This idea is captured in the following theorem:

THEOREM 1. *Let $N$ be the length of the originally transmitted bit string and let $K < N$ be the safety parameter used for error detection. Let $\pi : \{0,1\}^N \to \{0,1\}^N$ be a randomly chosen permutation. Let $f : \{0,1\}^N \to \{0,1\}^K$ and $g : \{0,1\}^N \to \{0,1\}^{N-K}$ be equitable functions defined by $f(x) = \pi(x)$ **mod** $2^K$ and $g(x) = \pi(x)$ **div** $2^K$. Then, knowledge of $\pi$ (hence of $f$ and $g$) and $f(x)$ yields no information on $g(x)$, except that it is of length $N-K$.*

From an information theoretic point of view, one might wonder if it is necessary that this information-reduction function $g$ be custom made for the particular error-detection function $f$ being used. At least two ideas come to mind: What happens if $g$ itself is chosen randomly, independently of $f$? and Could $g$ be known to Eve even before the private channel transmission takes place? Although these ideas do not allow wiping out Eve's information with certainty, they come close.

Let us first consider the case when the information-reduction function $g$ is randomly chosen among all functions $\{0,1\}^N \to \{0,1\}^R$, where $R \leqq N-K$ is the desired length of the final string. The intuitive hope is that a random $g$ is very likely to map the elements of $C$ nearly uniformly onto $\{0,1\}^R$, thus releasing very little information on the value of $g(x)$ from knowledge of $C$ and $g$ alone.

In order to state the theorems precisely, we need to introduce some information-theoretic formalism that will be used throughout § 4. Let $N$ and $K$ be as in Theorem 1. Let $S$ be any nonnegative integer smaller than $N-K$. Let $R = N-K-S$. Let **X**, **F** and **G** be three independent uniformly distributed random variables ranging over $\{0,1\}^N$, $\{f \mid f : \{0,1\}^N \to \{0,1\}^K\}$ and $\{g \mid g : \{0,1\}^N \to \{0,1\}^R\}$, respectively. Define new dependent random variables **Y** and **Z** ranging over $\{0,1\}^K$ and $\{0,1\}^R$, respectively, by setting $y = f(x)$ and $z = g(x)$.

The *expected amount of (Shannon) information* given on $g(x)$ by $y = f(x)$, $f$ and $g$ is defined by following formula:

$$\mathbf{I}(\mathbf{Z}; \mathbf{YFG}) = \sum_z \sum_y \sum_f \sum_g \text{prob}[z, y, f, g] \log \frac{\text{prob}[z \mid yfg]}{\text{prob}[z]}.$$

Notice that "expected" means "averaged over all possible choices for $f$, $g$ and $x$."

THEOREM 2. *Let $N$, $K$, $S$, $R$, **F**, **G**, **Y** and **Z** be as above, then*

$$\mathbf{I}(\mathbf{Z}; \mathbf{YFG}) \leqq \log(1 + 2^{-S}) < 2^{-S}/\ln 2.$$

*Furthermore, this bound is fairly tight because*

$$\mathbf{I}(\mathbf{Z}; \mathbf{YFG}) \geqq \left(\frac{1}{2} - \frac{1}{2^r}\right)\frac{2^{-S}}{\ln 2} - \frac{2}{2^N - 1}$$

*whenever* $3 \leqq S \leqq N-K-2$.

Intuitively, this says that if $f$ and $g$ are randomly chosen functions from $\{0,1\}^N$ to $\{0,1\}^K$ and to $\{0,1\}^R$, respectively, and if $x$ is randomly chosen among all bit strings of length $N$, then the expected amount of Shannon information given on $g(x)$ by $f(x)$, $f$ and $g$ is less than $2^{-S}/\ln 2$ bits.

As for the second idea, it turns out that a comparable level of information reduction can be achieved through the use of any ad hoc equitable function. In particular, it is enough to simply chop off any $K+S$ physical bits of $x$ in order to reduce the expected eavesdropper's information below $2^{-S}/\ln 2$ bits. This remains true even if the reduction function is chosen a priori and known to the eavesdropper before the private channel transmission.

THEOREM 3. *Let $N$, $R$ and $S$ be as above. Let $g:\{0,1\}^N \to \{0,1\}^R$ be any fixed equitable function, then the expected amount of information given on $g(x)$ by $f(x)$, $f$ and $g$ is less than $2^{-S}/\ln 2$ bits.*

**4.1.2. Eliminating the information given by universal hashing error detection.** Let us now assume that a practical error-detection protocol was used from those proposed in § 3.1: the function $f:\{0,1\}^N \to \{0,1\}^K$ was randomly chosen among some universal$_2$ class of hash functions. Rather than developing a general theory of information elimination in this context, let us design an ad hoc technique for a specific universal$_2$ class mentioned in [WC], which we call $P$. We assume here that the reader is familiar with Galois field theory [Be].

Consider $a, b \in \mathrm{GF}(2^N)$ such that $a \neq 0$. The degree one polynomial $q_{a,b}(x) = ax + b$, arithmetic being done in $\mathrm{GF}(2^N)$, defines a permutation of $\mathrm{GF}(2^N)$. If we let $\sigma:\mathrm{GF}(2^N) \to \{0,1\}^N$ stand for the natural one-to-one correspondence, this induces a permutation $\pi_{a,b}(x):\{0,1\}^N \to \{0,1\}^N$ defined by $\pi_{a,b}(x) = \sigma(q_{a,b}(\sigma^{-1}(x)))$. Therefore, for any fixed $K \leqq N$, the function $h_{a,b}(x):\{0,1\}^N \to \{0,1\}^N$ defined by $h_{a,b}(x) = \pi_{a,b}(x) \bmod 2^K$ is equitable. Define the class $P = \{h_{a,b} \mid a, b \in \mathrm{GF}(2^N), a \neq 0\}$. It is elementary to prove that $P$ forms a universal$_2$ class of hash functions, so that it can be used for the error-detection protocol of § 3.1. (In fact, the class becomes *strongly* universal$_2$ if we allow $a = 0$, but this is to be avoided here because $h_{0,b}$ is not equitable.)

THEOREM 4. *Let $a$ and $b$ be any elements of $\mathrm{GF}(2^N)$ such that $a \neq 0$. Let $x$ be a random string of length $N$. Then knowledge of $a$, $b$ and $h_{a,b}(x)$ gives no information on the string defined as $g_{a,b}(x) = \pi_{a,b}(x) \operatorname{div} 2^K$, except that it is of length $N - K$.*

*Proof.* This is an immediate consequence of the fact that $\pi_{a,b}$ is a permutation of $\{0,1\}^N$: knowledge of the last $K$ bits of $\pi_{a,b}(x)$ gives no information on its first $N - K$ bits. □

In conclusion, use of the universal$_2$ class $P$ allows Alice and Bob to verify whether their strings are identical, with an error probability of at most $2^{-K}$. If they turn out to be the same (or if they differ little enough that bit twiddling is applicable), they can be transformed into new strings that are only $K$ bits shorter, on which Eve has no information at all. This is clearly optimal.

It is worth mentioning that the conceptually simpler universal$_2$ class $H_1$ of [CW] can be used instead of $P$ for error detection. This still allows subsequently a good, efficient information-reduction scheme, but wiping out with certainty Eve's information does not seem to be feasible. This is due to the fact that the functions in $H_1$ are not equitable. Using this class, it is nonetheless always possible to reduce the eavesdropper's expected information below 2 bits, and even below any threshold $\delta > 0$ by choosing $N$ large enough. For more details, please consult [R].

The following section investigates the situation in which eavesdropping has occurred, but tampering and transmission errors are not a concern for Alice and Bob.

**4.2. Reducing the private channel eavesdropper's information.** Let us now assume that partial eavesdropping may have occurred on the private channel. Recall that eavesdropping consists of Eve selecting a function $e:\{0,1\}^N \to \{0,1\}^K$ of her choice, whose value $e(x)$ she learns when $x$ is transmitted over the private channel. Alice and Bob know $K$ but otherwise have no information on which function $e$ was chosen by Eve. If Eve chooses an equitable function, $e(x)$ gives her $K$ bits of information on $x$. Otherwise, her expected amount of information is smaller, but she could occasionally get more. In this section, we assume that transmission errors and tampering are not a worry for Alice and Bob, so that an error-detection protocol of § 3.1 is not carried out. This assumption is removed in § 4.3.

To summarize, let $x$ be the length $N$ bit string common to Alice and Bob, and let $e(x)$ be the $K$ bits of information known by Eve about $x$. Alice and Bob wish to publicly agree on some function $g:\{0,1\}^N \to \{0,1\}^R$, for some $R \leqq N-K$, such that knowledge of $e$, $e(x)$ and $g$ leaves Eve with an arbitrary small fraction of one bit of information about $g(x)$.

A similar question was addressed by Ozarow and Wyner [OW]. However, their solution is nonconstructive (based on random coding), it assumes that the eavesdropper can only read physical bits from the private channel, and it does not reduce her information below one bit. On the other hand, their setting is not restricted to the exchange of a random string, as they wish Alice to be able to safely transmit a message of her choice to Bob. Moreover, they do not need to use an authenticated public channel after the private transmission. Their results should also be compared with our information-elimination protocol from § 5.

Back to the analogy with Wyner's *original* wiretap channel [W], the requirement that the eavesdropping function be compressive is analogous to Wyner's requirement that the wiretap channel have less capacity than the main channel. One might hope to generalize our setting to cover eavesdropping through an arbitrary channel of capacity $K/N$, selected by Eve but unknown to Alice and Bob. However, this generalization would weaken our results rather than strengthen them. For example, Eve could satisfy the $K/N$ capacity bound by asking for all $N$ bits of Alice's message $K/N$ of the time, while asking for none of it the rest of the time.

Notice that the effect of eavesdropping over the private channel is very similar to that of eavesdropping over the public channel described in § 4.1 in that the information gained by Eve can be characterized by a set $E=\{z \in \{0,1\}^N \mid e(z)=e(x)\}$ of possible candidates for $x$. However, there is a fundamental difference: contrary to the previously discussed set $C$, it is not the case that Alice and Bob have complete knowledge of $E$. For this reason, it is not possible for them, in general, to eliminate Eve's information with certainty.

THEOREM 5. *If Eve is free to choose her function $e:\{0,1\}^N \to \{0,1\}^K$ without any constraints, there is always a chance that she will gain complete information on $g(x)$, no matter how Alice and Bob choose the function $g:\{0,1\}^N \to \{0,1\}^R$.*

*Proof.* The idea is to make $e$ very nonequitable, so as to have a small chance of learning $x$ exactly from $e(x)$. For example, Eve could choose:

$$e(x)=\begin{cases} x \bmod 2^K & \text{if } x \operatorname{div} 2^K = 0^{N-K}, \\ 0^K & \text{otherwise.} \end{cases}$$

With probability $(2^K-1)/2^N$, $e(x)$ yields complete information on $x$; hence on $g(x)$. □

The eavesdropping function described above is for gamblers only: with probability greater than $1-2^{K-N}$, $e(x)$ gives almost no information on $x$. In fact, the expected information on $x$ given by $e(x)$ is less than $(N+1/\ln 2)2^{K-N}$ bits. For instance, if Eve is allowed to observe the result of a 50-bit function applied to a 56-bit DES key [NBS], her expected information would be less than one bit if she had used the above very nonequitable eavesdropping function. Clearly, it would be more prudent for Eve to select some equitable function, so as to maximize her expected amount of information on $x$. Even then, an analogue to Theorem 5 can be given:

THEOREM 6. *No matter how Alice and Bob choose their function $g:\{0,1\}^N \to \{0,1\}^R$, for any $R>0$, there is always an equitable function $e:\{0,1\}^N \to \{0,1\}^K$, for any $K>0$, such that knowledge of $e$, $g$ and $e(x)$ yields some information on $g(x)$.*

*Proof.* Should Alice and Bob choose a nonequitable function $g$, Eve would have some information on $g(x)$ from the mere knowledge that $x$ is truly random, without

even looking at $e(x)$. On the other hand, assuming that $g$ is equitable, $e(x)$ could give as much as $\min(K, R)$ bits of information on $g(x)$. This is accomplished if one of the two functions is an equitable refinement of the other. Of course, Eve cannot select $e$ so that this will happen, because she does not yet know $g$ when she has to choose $e$. However, Alice and Bob cannot prevent this coincidence from happening, or even detect it, because they never get to know the function $e$. □

The above two theorems show that the best Alice and Bob can hope for is to reduce Eve's information arbitrarily close to zero. There can be no analogue to Theorems 1 and 4. Nonetheless, if we restrict even further Eve's choice of $e$, so that she can only read a selection of *physical* bits of $x$, it becomes possible again for Alice and Bob to eliminate her information entirely, as discussed in § 5.

As usual, we consider two approaches for the reduction of Eve's information: one based on truly random functions and one based on universal hashing techniques. Section 4.2.1 is only of theoretical interest, and the proofs can be found in [R]. Here again, it is nice to find out that the practical schemes of § 4.2.2 are just as efficient for information reduction as the unrealistic random scheme.

**4.2.1. A random information-reduction approach.** Recall that knowledge of $e:\{0,1\}^N \to \{0,1\}^K$ allows Eve to restrict the set of her possible candidates for $x$ to $E=\{z\in\{0,1\}^N \mid e(z)=e(x)\}$, where $\#E=2^{N-K}$ if $e$ is equitable.

LEMMA 7. *Let $E$ be a nonempty set of equally likely candidates for $x$, and let $R$ be an integer. Let $g:\{0,1\}^N \to \{0,1\}^R$ be a randomly chosen function. Then, knowledge of $E$ and $g$ yields less than an expected* $\log(1+2^R/\#E)$ *bits of information on $g(x)$. Here, the result holds for any specific $E$ and the average is only over all choices for $g$.*

THEOREM 8. *Let $e:\{0,1\}^N \to \{0,1\}^K$ be any function, let $S<N-K$ be a safety parameter, and let $R=N-K-S$. If $g:\{0,1\}^N \to \{0,1\}^R$ is chosen randomly, the expected amount of information on $g(x)$ given by knowledge of $e$, $g$ and $e(x)$ is less than* $\log(1+2^{-S})$ *bits, hence less than* $2^{-S}/\ln 2$ *bits.*

**4.2.2. A universal hashing information-reduction approach.** Contrary to the error detection protocols of § 3, it is no longer sufficient to consider universal$_2$ classes: here, we need *strongly* universal$_2$ classes [WC].

LEMMA 9. *Let $E$ and $R$ be as in Lemma 7. Let $H$ be a publicly known strongly universal$_2$ class of hash functions from $\{0,1\}^N$ to $\{0,1\}^R$. Let $g$ be a function chosen randomly within $H$. Then, knowledge of $E$ and $g$ yields less than an expected* $\log(1+2^R/\#E)$ *bits of information on $g(x)$. Again, the average is only over all choices of $g$, not over $E$ or $H$.*

*Proof.* Let us first recall some notation from universal hashing:
if $x\in\{0,1\}^N$, $z\in\{0,1\}^R$ and $g\in H$, then

$$\Delta_z^{x,g}=\begin{cases}1 & \text{if } g(x)=z,\\ 0 & \text{otherwise.}\end{cases}$$

If $E\subseteq\{0,1\}^N$, then

$$\Delta_z^{E,g}=\sum_{x\in E}\Delta_z^{x,g}=\#\{x\in E \mid g(x)=z\}.$$

Similarly, if $F\subseteq H$, then

$$\Delta_z^{x,F}=\sum_{g\in F}\Delta_z^{x,g}=\#\{g\in F \mid g(x)=z\}.$$

By definition of strong universal$_2$,

$$\#\{g\in H \mid g(x)=z \text{ and } g(x')=z'\}=\#H/2^{2R}$$

for any $x, x'\in\{0,1\}^N$ and $z, z'\in\{0,1\}^R$, provided $x\neq x'$.

An immediate consequence of this definition is that

$$\begin{aligned}\Delta_z^{x,H} &= \#\{g\in H | g(x)=z\}\\ &= \sum_{z'\in\{0,1\}^R} \#\{g\in H | g(x)=z \text{ and } g(x')=z'\}\\ &= 2^R \# H/2^{2R} = \# H/2^R,\end{aligned}$$

where $x'$ is chosen as any string different from $x$ in $\{0,1\}^N$.
Therefore,

$$\sum_{g\in H} \Delta_z^{E,g} = \sum_{x\in E} \Delta_z^{x,H} = \# E \# H/2^R.$$

Similarly,

$$\begin{aligned}\sum_{g\in H} (\Delta_z^{E,g})^2 &= \sum_{g\in H} \left(\sum_{x\in E} \Delta_z^{x,g}\right)^2\\ &= \sum_{g\in H} \sum_{x\in E} \sum_{x'\in E} \Delta_z^{x,g}\Delta_z^{x',g}\\ &= \sum_{x\in E} \sum_{x'\in E} \#\{g\in H | g(x)=z \text{ and } g(x')=z\}\\ &= \# E[\# H/2^R + (\# E-1)\# H/2^{2R}]\end{aligned}$$

(from splitting the cases $x'=x$ and $x'\neq x$)

$$= \frac{\# E \# H}{2^R}\left[1+\frac{\# E-1}{2^R}\right].$$

Let us now come back to Eve's set $E\subseteq\{0,1\}^N$ of equally likely candidates for $x$. In this proof, we consider two independent random variables $\mathbf{X}$ and $\mathbf{G}$ ranging over $\{0,1\}^N$ and the set of functions from $\{0,1\}^N$ to $\{0,1\}^R$, respectively, with the probability distributions

$$\text{prob}[\mathbf{X}=x] = \begin{cases} 1/\# E & \text{if } x\in E,\\ 0 & \text{otherwise,}\end{cases}$$

and

$$\text{prob}[\mathbf{G}=g] = \begin{cases} 1/\# H & \text{if } g\in H,\\ 0 & \text{otherwise.}\end{cases}$$

Consider also the dependent random variable $\mathbf{Z}$ ranging over $\{0,1\}^R$ corresponding to the equation $z=g(x)$. We wish to find an upper bound on $\mathbf{I}(\mathbf{Z};\mathbf{G})$, the expected information on $g(x)$ given from the knowledge of $g$ and of the fact that $x\in E$. For this, we need to compute a few conditional, joint and marginal probabilities: given $z\in\{0,1\}^R$ and $g\in H$,

$$\text{prob}[\mathbf{Z}=z|\mathbf{G}=g] = \Delta_z^{E,g}/\# E,$$

$$\begin{aligned}\text{prob}[\mathbf{Z}=z,\mathbf{G}=g] &= \text{prob}[\mathbf{G}=g]\,\text{prob}[\mathbf{Z}=z|\mathbf{G}=g]\\ &= \Delta_z^{E,g}/\# E \# H,\end{aligned}$$

and

$$\text{prob}[\mathbf{Z}=z] = \sum_{g\in H} \text{prob}[\mathbf{Z}=z, \mathbf{G}=g]$$

$$= \frac{1}{\#E\#H} \sum_{g\in H} \Delta_z^{E,g} = 2^{-R}.$$

By information-theoretic definitions, we therefore obtain:

$$\mathbf{I}(\mathbf{Z};\mathbf{G}) = \sum_{z\in\{0,1\}^R} \sum_{g\in H} \text{prob}[\mathbf{Z}=z, \mathbf{G}=g] \log \frac{\text{prob}[\mathbf{Z}=z|\mathbf{G}=g]}{\text{prob}[\mathbf{Z}=z]}$$

$$= \sum_z \sum_g \frac{\Delta_z^{E,g}}{\#E\#H} \log \frac{\Delta_z^{E,g} 2^R}{\#E}$$

$$= 2^{-R} \sum_z \sum_g \Delta_z^{E,g} \frac{2^R}{\#E\#H} \log \frac{\Delta_z^{E,g} 2^R}{\#E}$$

$$\leqq 2^{-R} \sum_z \log \sum_g (\Delta_z^{E,g} 2^R / \#E)^2 / \#H$$

(by Jensen's lemma [Mc], because $\sum_g \Delta_z^{E,g} 2^R / \#E\#H = 1$)

$$= 2^{-R} \sum_z \log \left( \frac{2^{2R}}{(\#E)^2 \#H} \sum_g (\Delta_z^{E,g})^2 \right)$$

$$= 2^{-R} \sum_z \log \left( \frac{2^R}{\#E} + \frac{\#E-1}{\#E} \right)$$

$$< \log(1+2^R/\#E). \qquad \square$$

THEOREM 10. *Let e, S and R be as in Theorem* 8, *and let H and g be as in Lemma* 9. *The expected amount of information on* $g(x)$ *given by knowledge of e, g and* $e(x)$ *is less than* $\log(1+2^{-S})$ *bits, hence less than* $2^{-S}/\ln 2$ *bits. Notice that this is true for every e, despite the fact that Eve already knows the class H, but of course not the specific function g, when she gets to choose her function e.*

*Proof.* This is an immediate consequence of Lemma 9 if $e$ is equitable. Indeed, the eavesdropper's set $E$ is then always reduced to exactly $2^{N-K}$ equally likely candidates for $x$. The expected information on $g(x)$ given by knowledge of $E$ and $g$ is thus less than $\log(1+2^R/2^{N-K}) = \log(1+2^{-S}) < 2^{-S}/\ln 2$ bits.

If $e$ is not equitable, the theorem still follows from Lemma 9, but through the use of Jensen's lemma [Mc]. For any $x\in\{0,1\}^N$, let $E_x = \{y\in\{0,1\}^N | e(y) = e(x)\}$. Since each $x\in\{0,1\}^N$ is equally likely a priori, Lemma 9 tells us that the expected information on $g(x)$ given by knowledge of $g$, $e$ and $e(x)$ is less than

$$\sum_{x\in\{0,1\}^N} 2^{-N} \log(1+2^R/\#E_x) \leqq \log \sum_x (2^{-N} + 2^{R-N}/\#E_x)$$

(by Jensen's lemma, because $\sum_x 2^{-N} = 1$)

$$\leqq \log(1+2^{-S})$$

(because $\sum_x 1/\#E_x \leqq 2^K$, by an easy exercise left to the reader). $\square$

Let us finally point out that *almost* strongly universal$_2$ classes [WC] can also be used in this information-reduction context. A similar analysis shows that if $g$ is chosen

randomly from an almost strongly universal$_2$ class, knowledge of $e$, $g$ and $e(x)$ yields at most $1+\log(1+2^{-(S+1)})$ bits of information to Eve about $g(x)$. This could be interesting when $R$ is much smaller than $N$: in that case the description of a randomly chosen function within an almost strongly universal$_2$ class requires significantly fewer bits to be transmitted over the public channel [WC]. □

**4.3. Putting the concepts together.** If Eve obtained some information on the string $x$ as it was being transmitted over the private channel, and additional information by listening to the public channel messages exchanged during the error-correction and error-detection protocols of §§ 3.2 and 3.1, it may still be possible for Alice and Bob to efficiently agree on a random string on which Eve has nearly no information. The key idea is that if Eve obtained an expectation of $K$ bits from the private channel transmission, and an expectation of $L$ more bits from eavesdropping on the subsequent public discussion, then she can expect at most $K+L$ bits from both sources combined. This is formalized in the following easily proven lemma.

LEMMA 11. *Let $e:\{0,1\}^N \to \{0,1\}^K$ and $f:\{0,1\}^N \to \{0,1\}^L$ be any two functions. Let $x$ be a random string of length $N$. The expected information on $x$ given by knowledge of $e$, $e(x)$, $f$ and $f(x)$ is at most $K+L$ bits.*

Therefore, an obvious adaptation of the protocol implied by Theorem 10 allows for the efficient reduction of Eve's information to less than $2^{-S}/\ln 2$ bits for any $S<N-K-L$, at the cost of ending up with a random string of length $N-K-L-S$. Notice that ad hoc information-elimination schemes, such as those in Theorems 1 and 4, offer no advantages in this context (unless an information-elimination scheme from § 5 is used initially). Therefore, the choice of a universal$_2$ class in the error-detection part of the protocol does not have to be motivated by the existence of a subsequent information-elimination scheme.

**5. Elimination of the eavesdropper's information.** The protocols of § 4.2 should be sufficient for most applications, despite the fact that Eve retains an arbitrarily small fraction of one bit of information on the resulting shared random string. Although we were able to eliminate her information entirely in Theorems 1 and 4, the techniques used could only be applied because Alice and Bob had complete knowledge of Eve's information. As shown in Theorems 5 and 6, this cannot be extended whenever Eve is allowed to access a limited amount of information *of her choice* from the private channel transmission.

In this section, we investigate a protocol by which Alice and Bob can nonetheless wipe out Eve's information, assuming that she obtained a maximum of $K$ *physical* bits of her choice from the private channel transmission, as opposed to the more general $K$ bits of information in Shannon's sense discussed in § 4. Although the value of $K$ is known to Alice and Bob, they do not know, of course, which particular bits of their string are compromised. This protocol is expensive in the sense that the resulting string is generally substantially shorter than those resulting from the protocols of § 4; however, this is the unavoidable price to pay in order to make sure that Eve is left with no information at all.

An error-detection protocol could be applied, if desired, after Eve's information has been eliminated, still leaving her with no information if the universal$_2$ class $P$ of Theorem 4 is used. Bit twiddling on the initial strings is also possible afterwards in order to reconcile the final strings, if there were only one or two transmission errors. Unfortunately, the more sophisticated protocol of § 3.2 would transform Eve's knowledge from physical bits to information in Shannon's sense, so that the elimination

protocols described below could no longer be applied. We do not know how to efficiently eliminate Eve's information and reconcile Alice and Bob's strings if several transmission errors occurred.

**5.1. The notion of $(N, J, K)$-functions.** In order to eliminate Eve's information, we introduce the following definition: for any integers $N$, $J$ and $K$ such that $N \geqq J+K$, $J>0$ and $K>0$, a function $f:\{0,1\}^N \to \{0,1\}^J$ is $(N, J, K)$ if, no matter how one fixes any $K$ of its input bits, each of the $2^J$ output bits can be produced in exactly $2^{N-J-K}$ different ways by varying the remaining $N-K$ input bits. Intuitively, an $(N, J, K)$-function compresses an $N$ bit string into a $J$ bit string in such a way that knowledge of any $K$ of the input bits gives no information on the output. This is equivalent to the notion of $t$-resilient functions independently introduced by [CGHFRS].

Given such a function, Alice and Bob can apply it to their respective strings, thus producing a new (shorter) string on which Eve has no information. Notice that this still holds even if she already knows which function will be used by Alice and Bob in advance of her deciding which $K$ bits to read from the private channel. Therefore, the subsequent public transmission between Alice and Bob is not necessary in this case. By analogy with [OW], these $(N, J, K)$-functions are not restricted to the exchange of random strings. If Alice wished to communicate some *specific* $J$ bit string $y$ to Bob, she could send over the private channel some randomly chosen $N$ bit string $x$ such that $f(x)=y$. This would allow Bob to obtain $y$ unambiguously, assuming no transmission errors occurred, whereas Eve would gain no information on $y$ from eavesdropping over any $K$ bits of $x$. One (nonconstructive) protocol in [OW] achieves something similar with $N=J+K$, which is better than what we get here, but it yields slightly more than one bit of information to Eve about $y$.

*Example.* The function $f(u, v, w, x, y, z) = (u \oplus v \oplus w \oplus x,\ w \oplus x \oplus y \oplus z)$ is $(6, 2, 3)$: knowledge of any 3 of the input bits yields no information at all on the output.

The case $J = N-K$ is the best possible because there is obviously no hope of producing a completely secret string of length $N-K+1$ if Eve knows $K$ of the $N$ original bits. A function $f$ that is $(N, N-K, K)$ is said to be $(N, K)$. The following theorem shows how to build $(N, K)$-functions whenever they exist.

THEOREM 12. (1) *For any $N>1$, there are $(N, 1)$ and $(N, N-1)$-functions.*

(2) *For any $N>3$, there are no $(N, K)$-functions whenever $1<K<N-1$.*

*Proof.* (1) Produce the $i$th output bit as the exclusive-or of the $i$th and the $(i+1)$st input bits to get an $(N, 1)$-function; and produce the only output bit as the exclusive-or of all $N$ input bits to get an $(N, N-1)$-function.

(2) Assume for a contradiction that some $f:\{0,1\}^N \to \{0,1\}^{N-K}$ is $(N, K)$ for $N>3$ and $1<K<N-1$. By definition $f(x_1) \neq f(x_2)$ for any two distinct strings $x_1$ and $x_2$ that have at least $K$ bits in common. Let $X = \{x \in \{0,1\}^N \mid x \neq 0^N \text{ and } x \bmod 2^K = 0^K\}$, the set of nonzero $N$ bit strings ending with $K$ zeros. Notice that $f$ must be one-to-one on $X$ because any two strings in $X$ have their last $K$ bits in common. Now, consider the strings $u = 1^{N-K+1}0^{K-1}$ and $v = 1^{N-K}0^{K-1}1$. Both $u$ and $v$ have at least $K$ bits in common with each string of $X$. Therefore, $f(u) \notin f[X]$ and $f(v) \notin f[X]$. Since $f[X]$ spans all of $\{0,1\}^{N-K}$ but one string, we must have $f(u)=f(v)$. This is a contradiction because $u$ and $v$ have $(K-2)+(N-K)=N-2 \geqq K$ bits in common. □

Moreover, there exist only two distinct $(N, N-1)$-functions for each $N>1$: the one given in the proof of Theorem 12 and its complement:

THEOREM 13. *The only $(N, N-1)$ functions are $f(x_1, x_2, \cdots, x_N) = x_1 \oplus x_2 \oplus \cdots \oplus x_N$ and its complement.*

*Proof.* This easy proof by induction on $N$ is left to the reader. □

**5.2. How to build $(N, J, K)$-functions.** We wish to answer the following question: given $N$ and $K$, what is the maximum value for $J$ such that an $(N, J, K)$-function exists? Alternatively, given $N$ and $J$, find the maximum value for $K$. In other words, what is the longest secret random string on which Alice and Bob can agree if they start from a random string of length $N$, of which $K$ bits are compromised? Theorem 12 showed that $J$ must be strictly smaller than $N-K$ unless $K=1$ or $K=N-1$.

We were unable to answer the above question in its full generality. For this reason, we restrict our attention to the special class of $(N, J, K)$-functions for which every output bit is produced as the exclusive-or of some of the input bits. Such functions are referred to as xor-$(N, J, K)$-functions. We conjecture that these functions are as efficient as possible, in the sense that if no xor-$(N, J, K)$-functions exist for given values of $N$, $J$ and $K$, then no general $(N, J, K)$-functions exist either. This *Xor-Conjecture* is proved in [CGHFRS] for the case $J=2$.

The following characterization, known as the *Xor-Lemma*, allows us to establish an equivalence between xor-$(N, J, K)$-functions and binary linear codes [vL].

LEMMA 14 (independently discovered by [CGHFRS]). *Let $M$ be a $J \times N$ Boolean matrix. Let $f: \{0,1\}^N \to \{0,1\}^J$ be the function represented by $M$ in the natural way (i.e., $f(x) = xM^t$, all operations being performed modulo 2). The function $f$ is $(N, J, K)$ if and only if the exclusive-or of any set of rows of $M$ contains at least $K+1$ ones.*

*Proof.* One direction is obvious: if the exclusive-or of some subset of the rows contains $K$ ones or less, it is sufficient to know the value of these $K$ input bits to infer the exclusive-or of the corresponding output bits.

Conversely, assume that the exclusive-or of any set of rows of $M$ contains at least $K+1$ ones. We have to show that, no matter how many $K$ input bits are fixed, each of the $2^J$ output strings can be obtained in exactly $2^{N-J-K}$ different ways by varying the other $N-K$ input bits. Without loss of generality, let us assume that the first $K$ input bits are fixed, say to some $u \in \{0,1\}^K$. Let $M_1$ and $M_2$ stand for the first $K$ and the last $N-K$ columns of $M$, respectively.

The key observation is that $M_2$ has full rank, because if the exclusive-or of some rows of $M_2$ were zero, the exclusive-or of the same rows of $M$ would contain at most $K$ ones. A classic result of linear algebra applies to conclude that there exists a nonsingular $(N-K)\times(N-K)$ matrix $F$ such that $M_2 = RF$, where $R$ is the $J \times (N-K)$ matrix such that $R_{ii}=1$ for $1 \leqq i \leqq J$ and $R_{ij}=0$ otherwise [ND].

Now, consider any $y \in \{0,1\}^J$. Let $z$ be any string in $\{0,1\}^{N-K-J}$. Let $v = (y \oplus uM_1^t, z)(F^t)^{-1}$. Let $x=(u, v)$. We have $f(x) = xM^t = uM_1^t \oplus vM_2^t = uM_1^t \oplus [(y \oplus uM_1^t, z)(F^t)^{-1}F^tR^t] = y$. Furthermore, it is clear that different values for $z$ give in this way different values for $v$, hence for $x$. Therefore for any $y \in \{0,1\}^J$, there are at least $2^{N-K-J}$ different $x \in \{0,1\}^N$ such that the first $K$ bits of $x$ are $u$ and $f(x)=y$. By a pigeonhole argument, $2^{N-K-J}$ is the exact number of such $x$'s. This is the required condition for $f$ to be $(N, J, K)$. □

THEOREM 15 (independently discovered by [CGHFRS]). *For given values of $N$, $J$ and $K$, there exists an xor-$(N, J, K)$-function if and only if there exists an $[N, J]$ binary linear code with minimum distance at least $K+1$ between any two code words.*

*Proof.* This is an immediate consequence of Lemma 14, if one makes the correspondence between the matrix $M$ used to represent the $(N, J, K)$-function and the generator matrix $G$ of the binary linear code. □

Consequently, our problem is equivalent to a classic problem of algebraic coding theory. Unfortunately, no efficient algorithms, much less closed-formed formulae, are known to determine the largest possible minimum code-word distance among all $[N, J]$ binary linear codes. There are, however, several well-known lower and upper bounds

on this value [vL], [HS], [MS], and these bounds apply just as well to our problem.

For instance, Hamming codes [MS] tell us that xor-$(2^L-1, 2^L-L-1, 2)$-functions exist for every $L \geqq 2$. Conversely, Hamming's upper bound [MS] shows that no xor-$(2^L-1, 2^L-L, 2)$-functions can exist because

$$2^{(2^L-1)-(2^L-L-1)} = \sum_{i=0}^{1} \binom{2^L-1}{i}.$$

Notice that elimination of Eve's information in this case ($K=2$) costs $L-2-S$ additional bits than if we had been satisfied to reduce her information below $2^{-S}/\ln 2$ bits, as in § 4.2.

Similarly, Griesmer's upper bound and the simplex code [MS] allow for the building of xor-$(2^L-1, L, 2^{L-1}-1)$-functions for any $L \geqq 2$, whereas neither xor-$(2^L-1, L, 2^{L-1})$-functions nor xor-$(2^L-1, L+1, 2^{L-1}-1)$-functions can exist. Moreover, Varsharmov-Gilbert's lower bound together with McEliece's upper bound [vL], [MS] allow for the construction of xor-$(N, J, K)$-functions such that $J$ is at least half the optimal (xor) value, as long as $K/N < 0.3$ and $N$ is large enough. Also, the *zigzag* of [BCR] yields an xor-$(3^L, 2^L, 2^L-1)$-function, for every positive integer $L$.

Finally, the general question of $(N, J, K)$-functions is solved completely when $K > \frac{2}{3}N-1$, thanks to the proof of the Xor-Conjecture for $J=2$ [CGHFRS]. In this case, it is easy to see that no xor-$(N, 2, K)$-functions can exist, and therefore no general $(N, 2, K)$-functions can exist either. Since the exclusive-or of all the input bits is $(N, 1, K)$, we conclude that 1 is the maximum possible value for $J$ when $\frac{2}{3}N-1 < K < N$. On the other hand, there is always an $(N, 2, \lfloor \frac{2}{3}N-1 \rfloor)$-function. We encourage the reader to consult [CGHFRS] for additional results on $(N, J, K)$- (*alias* $t$-resilient) functions.

**6. Conclusions.** If no eavesdropping occurred over the private channel, it is possible for Alice and Bob to publicly verify that no transmission errors or tampering occurred either, with a $2^{-K}$ error probability, and end up with an entirely secret final string that is only $K$ bits shorter than the original private transmission. This is optimal.

If partial eavesdropping occurred over the private channel, leaking up to $K$ bits of information to Eve, in Shannon's sense, it is still possible for Alice and Bob to publicly verify that no transmission errors or tampering occurred, with a $2^{-L}$ error probability, and end up with a final string that is $K+L+S$ bits shorter than the original private transmission, on which Eve has less than $2^{-S}/\ln 2$ bits of information on the average. Moreover, discrepancies between the transmitted and received versions of the private channel transmission, whether they are due to channel noise or tampering, can be handled at the cost of a further reduction in the length of the final shared secret string. If the discrepancies are too numerous, no final shared secret string can be constructed, but Alice and Bob will detect this condition with very high probability, and will not be misled into constructing a string that is neither shared nor secret.

Finally, if partial eavesdropping over the private channel is restricted to $K$ physical bits secretly chosen by Eve, it becomes possible again for Alice and Bob to verify with high probability that no errors or tampering occurred, and to end up with a new string on which Eve has no information whatsoever. However, the new string will be substantially shorter than if Alice and Bob had tolerated knowledge by Eve of an arbitrarily small fraction of one bit of information. This remains possible even if a small number of transmission errors occurred, but we do not know how to eliminate Eve's information and reconcile the strings efficiently in the presence of severe transmission errors.

## REFERENCES


[BB1] C. H. BENNETT AND G. BRASSARD, *Quantum cryptography and its application to provably secure key expansion, public-key distribution and coin-tossing*, Proc. IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, December 1984, pp. 175–179.

[BB2] ———, *An update on quantum cryptography*, in Advances in Cryptology: Proc. CRYPTO 84, G. R. Blakley and D. Chaum, eds., Lecture Notes in Computer Science 196, Springer-Verlag, Berlin, 1985, pp. 475–480.

[BBR] C. H. BENNETT, G. BRASSARD AND J.-M. ROBERT, *A perfect secrecy interactive reconciliation protocol*, in preparation; some of the results can be found in *How to reduce your enemy's information*, in Advances in Cryptology: Proc. CRYPTO 85, H. C. Williams, ed., Lecture Notes in Computer Science 218, Springer-Verlag, Berlin, 1986, pp. 468–476.

[Be] E. R. BERLEKAMP, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[Br] G. BRASSARD, *On computationally secure authentication tags requiring short secret shared keys*, in Advances in Cryptology: Proc. CRYPTO 82, D. Chaum, R. L. Rivest and A. T. Sherman, eds., Plenum, New York, 1983, pp. 267–275.

[BCR] G. BRASSARD, C. CRÉPEAU AND J.-M. ROBERT, *Information theoretic reductions among disclosure problems*, Proc. 27th IEEE Symposium on Foundations of Computer Science, Toronto, Ontario, 1986, pp. 168–173.

[CW] J. L. CARTER AND M. N. WEGMAN, *Universal classes of hash functions*, J. Comput. System Sci., 18 (1979), pp. 143–154.

[CGHFRS] B. CHOR, O. GOLDREICH, J. HASTAD, J. FREIDMANN, S. RUDICH AND R. SMOLENSKY, *The bit extraction problem or t-resilient functions*, Proc. 26th IEEE Symposium on Foundations of Computer Science, Portland, Oregon, 1985, pp. 396–407.

[DH] W. DIFFIE AND M. E. HELLMAN, *New directions in cryptography*, IEEE Trans. Inform. Theory, IT-22 (1976), pp. 644–654.

[G] R. G. GALLAGER, *Information Theory and Reliable Communications*, John Wiley, New York, 1968.

[GGM] O. GOLDREICH, S. GOLDWASSER AND S. MICALI, *How to construct random functions*, Proc. 25th IEEE Symposium on Foundations of Computer Science, Singer Island, FL, 1984, pp. 464–479.

[HS] H. J. HELGERT AND R. D. STINOFF, *Minimum distance bounds for binary linear codes*, IEEE Trans. Inform. Theory, IT-19 (1973), pp. 344–356.

[vL] J. H. VAN LINT, *Introduction to Coding Theory*, Graduate Text in Mathematics 86, Springer-Verlag, New York, 1982.

[MS] F. J. MACWILLIAMS AND N. J. A. SLOANE, *The Theory of Error-Correcting Codes*, North-Holland, New York, 1977.

[Mc] R. J. MCELIECE, *The theory of information and coding*, in Encyclopedia of Mathematics and its Applications, Vol. 3, Addison-Wesley, Reading, MA, 1977.

[NBS] NATIONAL BUREAU OF STANDARDS, *Data Encryption Standard*, FIPS PUB 46, Washington, DC, 1977.

[ND] B. NOBLE AND J. W. DANIEL, *Applied Linear Algebra*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1977.

[OW] L. H. OZAROW AND A. D. WYNER, *Wire-tap channel* II, Bell System Tech. J., 63 (1984), pp. 2135–2157.

[RSA] R. L. RIVEST, A. SHAMIR AND L. ADLEMAN, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM, 21 (1978), pp. 120–126.

[R] J.-M. ROBERT, *Détection et correction d'erreur en cryptographie*, Masters thesis, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal, Québec, Canada, 1985.

[S] C. SHANNON, *The mathematical theory of communication*, Bell System Tech. J., 27 (1948), pp. 379–423, 623–656.

[WC] M. N. WEGMAN AND J. L. CARTER, *New hash functions and their use in authentication and set equality*, J. Comput. System Sci., 22 (1981), pp. 265–279.

[W] A. D. WYNER, *The wire-tap channel*, Bell System Tech. J., 54 (1975), pp. 1355–1387.