Low-Complexity Wire-Tap Codes with Security and Error-Correction Guarantees

Yuval Cassuto
School of Computer and Communication Sciences
ALGO
EPFL, Lausanne
Switzerland
Email: yuval.cassuto@epfl.ch

Zvonimir Bandic
Hitachi Global Storage Technologies
San Jose Research Center
3403 Yerba Buena Road
San Jose, CA 95135, U.S.A.
Email: zvonimir.bandic@hitachigst.com

***Abstract*—New code constructions are proposed for the wire-tap channel with security and error-correction guarantees. For the case of error-free main channels, two families of codes are constructed with optimal encoding and decoding complexities for their wire-tap security. For the case of main channels with errors, two concatenation types are studied for the wire-tap and error-correcting codes. For each of these concatenated schemes, code families are constructed that give optimal cooperation between the wire-tap and error-correction properties. The motivation to study low-complexity wire-tap codes with security and error-correction guarantees comes from data storage applications. Due to imperfect physical erasure processes, important secret information needs to be protected from adversarial access to residual, post erasure, information, and at the same time be protected from errors when read by the legitimate device user.**

I. Introduction

Information security owes its deepest roots to information theory. Fundamental theorems by Shannon [5] achieved and bounded the ability to communicate with unconditional secrecy. Unconditional secrecy, thereafter dubbed information-theoretic secrecy, marks the boundaries of the security envelope in which other fields of information security are thriving. But nonetheless, when it comes to impact on practical security systems, the contribution of information theory remained secondary to that of the theory of computational security from theoretical computer science.

In this paper we pursue a branch of information-theoretic security that has a true and immediate application in security systems. The problem it addresses is *imperfect physical erasure* of secret data in storage devices. Regardless of the particular storage medium or technology, the physical process used to erase data is inherently imperfect. Ensuring a complete erasure of every information bit is either not possible, or has some high associated cost (e.g. time cost of repeated multiple erases [7]). The implication of an imperfect erasure process is that an adversary with access to the storage device after erasure would be able to extract secret information, and compromise the security of the system. The starting point of this paper is a simple observation that the imperfect erasure problem can be regarded as a wire-tap channel [9] problem. The wire-tap channel in this case is the physical erasure process that introduces errors and erasures to the adversary's observations of the stored information. The main channel is the usual storage channel, whereby the legitimate user reads the stored information prior to an erase operation, possibly with some errors due to imperfect write or read processes. The design objective in a wire-tap problem is to encode the information such that it can be perfectly reconstructed by the legitimate receiver on the main channel, and at the same time reveal no information to the adversary on the wire-tap channel. A framework that was developed to address wire-tap code design is the wire-tap channel II of Ozarow and Wyner [4]. The wire-tap channel II can be regarded as the coding-theoretic counterpart of the information-theoretic wire-tap channel. In that framework, the security of the information is ensured from a worst-case perspective, and concrete security guarantees are provided for the wire-tap codes. Given the conservative nature of actual security implementations, worst case security with guarantees is the preferred approach to handle the imperfect erasure problem. The availability of code families with the guaranteed security as their parameter, allows a system designer to choose the suitable protection given the particular system's properties and constraints.

The current paper contributes to the study of wire-tap II codes in two directions. First, it seeks to construct wire-tap codes with the lowest possible encoding and decoding complexities. Second, it provides optimal constructions with security and error-correction guarantees for the case of main channels with errors. Previous works that considered the case of main channels with errors either did not provide guaranteed security and error correction [6], or required wire-tap codes that are complementary dual [1]. The motivation to choose the encoding and decoding complexities – and not the code redundancy – as the primary criteria of optimization, is that in a typical storage device only an insignificant part of the storage capacity is used for secret information (e.g. for encryption keys), while reads and writes of this secret information need to be performed quickly and frequently. After an extended introduction of the wire-tap II framework in section II, the paper is divided into two parts. Part 1 is dedicated to the case of error-free main channels, where two families of codes are proposed in section III with optimal encoding and decoding complexities. A code is said to have optimal encoding and decoding complexities if the density of its parity-check matrix is the lowest possible given its wire-tap security parameter. In Part 2, we move to discuss code constructions for the case of main channels with errors, and study two concatenation types of wire-tap and error-correcting codes: outer code for errors and inner for wire-tap in section IV, and inner code for errors and outer for wire-tap in section V. In the former we propose a code construction with the lowest possible degradation of correction capability due to the wire-tap code. In the latter we propose a code construction that preserves the wire-tap security properties of the outer code in the presence of the inner error-correcting code. In both cases the constructions of

978-1-4244-8264-1/10/$26.00 ©2010 IEEE

Part 1 are essential building blocks for the favorable properties achieved in Part 2.

II. The Wire-Tap Channel II

The wire-tap channel II, proposed by Ozarow and Wyner in 1984 [4], builds upon ideas from Wyner's original wire-tap channel [9], focusing on security properties from a worst-case perspective. A worst-case analysis means that coding schemes are secure for *any* bounded-size subset of the codeword locations that is accessible by the adversary. In this section we briefly review the definitions and construction methods from [4]. The notation used here is an adaptation of the original one.

A. Problem statement and model definition

Let k be the number of symbols in the secret data. The objective is to encode the k data symbols into $n > k$ code symbols, such that an adversary who observes a symbol subset of his choice of size $\mu < n$ can gain no information on the secret data.

B. Construction technique

For the error-free main channel case, the coding scheme proposed in [4] is now summarized. Given the parameters k, n, μ defined in the previous sub-section, a $[n, k, \mu]$ linear wire-tap code $\mathcal{C}$ is constructed using a $k \times n$ systematic parity-check matrix each of whose $k \times (n-\mu)$ sub-matrix has rank k. Given a code with the prescribed properties above, the coding system is used as follows.

Encoding: $n-k$ symbols are drawn at random and are used to encode a random, length n, codeword $\boldsymbol{c}$ from the code $\mathcal{C}$. The k information symbols are placed in a row vector $\boldsymbol{a} = [a_1, \ldots, a_k]$. Then the encoded vector $\boldsymbol{v}$ is simply

$$\boldsymbol{v} = \boldsymbol{c} + [a_1, \ldots, a_k, \underbrace{0, \ldots, 0}_{n-k}] \tag{1}$$

Another way to view the encoding function is as selecting a random member from the coset that corresponds to the secret vector $[a_1, \ldots, a_k]$.

Decoding: Given the length n vector $\boldsymbol{v}$, the legitimate user calculates $H\boldsymbol{v}^T$ to recover the secret data vector, where $(\cdot)^T$ represents the usual vector/matrix transposition operator.

$$H\boldsymbol{v}^T = H\boldsymbol{c}^T + H \cdot [a_1, \ldots, a_k, \underbrace{0, \ldots, 0}_{n-k}]^T = \mathbf{0}^T + \boldsymbol{a}^T = \boldsymbol{a}^T$$

Wire-Tapping: Given any μ known symbols of the encoded vector, the wire-tapper has to solve a linear system of the form:

$$\begin{bmatrix} | & | & & | \\ h_{i_1} & h_{i_2} & \cdots & h_{i_{n-\mu}} \\ | & | & & | \end{bmatrix} \begin{bmatrix} v_{i_1} \\ v_{i_2} \\ \\ \vdots \\ v_{i_{n-\mu}} \end{bmatrix} = \boldsymbol{a}^T + \boldsymbol{s}^T \tag{2}$$

The adversary's objective is to find $\boldsymbol{a}^T$, while $\boldsymbol{s}^T$ is known (calculated from the known μ symbols), and $[v_{i_1}, v_{i_2}, \ldots, v_{i_{n-\mu}}]$ is not known. Since by design any set of coordinates $\{i_1, i_2, \ldots, i_{n-\mu}\}$ corresponds to a sub-matrix with rank k, the linear system has a solution (in the form of a vector $[v_{i_1}, v_{i_2}, \ldots, v_{i_{n-\mu}}]$) for any vector $\boldsymbol{a}^T$. Hence, by knowing parts of $\boldsymbol{v}$, it has been shown [4] that the adversary gains no information on the secret data in $\boldsymbol{a}$. Note that if the sub-matrix has rank less than k, then some $\boldsymbol{a}$ will *n*ot have a corresponding $\boldsymbol{v}$ solution, and the adversary can exclude potential secret values – thus compromising the security of the information.

One simple instantiation of the construction method is to use the parity code to obtain a wire-tap code with $k = 1$ and $\mu = n - 1$. Refer to [4] for details.

Part 1 of this paper includes code constructions for the error-free wire-tap II method described above. Then in Part 2 the code constructions address the case where the main channel has errors.

Part 1: Error-free Main Channel

III. Optimal-Complexity Wire-Tap Codes

As stated in section II, for perfect wire-tap security against an adversary with access to any μ code symbols, the code's parity-check matrix is required to have all of its $k \times (n-\mu)$ sub-matrices with rank k. This code property is equivalent to requiring that the *dual* of the code will have minimum Hamming distance of at least $\mu+1$ [8]. To convince oneself in the last fact, observe that if (and only if) a code has minimum distance μ or less, then there exists a $k \times (n-\mu)$ sub-matrix of the *generator* matrix with rank smaller than k (such that a non-zero codeword has zeros in all of the $n-\mu$ coordinates that correspond to this sub-matrix). Hence the problem of finding good wire-tap codes is reduced to the most studied problem in coding theory: finding linear codes with high dimensions and large minimum Hamming distances. However, when the encoding and decoding complexities of the wire-tap codes are considered, traditional codes with large minimum Hamming distance may be inferior to the best possible wire-tap codes. Before proposing actual low-complexity codes, we discuss the encoding and decoding complexities in generality.

A. Encoding and decoding complexities

If secrets are to be injected to, and retrieved from the channel at a high rate, attention must be paid to the complexity of the encoding and decoding operations of the wire-tap code. Unlike codes for noisy channels, the decoding operation of wire-tap codes is a straightforward linear-code syndrome calculation, without the need to find the coset-leader in the received coset. Hence the encoding and decoding complexities of the wire-tap code are directly related to the densities of its generator and parity-check matrices, respectively. Recall that in sub-section II-B it was required to have a systematic representation of the parity-check matrix H, which allows the simple encoder of equation (1). When H is given in systematic form, both the encoding and decoding complexities can be determined from the density of H. Both the encoding and decoding complexities are quoted per information symbol, hence the appropriate matrix density to evaluate is the average row density, defined in the following.

Definition 1: The (row) **density** of a matrix is the number of its non-zero elements divided by its number of rows.

For a given wire-tap security parameter μ, minimizing the density of the systematic parity-check matrix thus minimizes both the encoding and decoding complexities of the code in the wire-tap scheme.

Construction 1: Let $A_k = \{1, 2, \ldots, k, k+1\}$ be the set of integers between 1 and $k+1$ (inclusive), and $S_k = \{\{i, j\} : i \in A_k, j \in A_k, i \neq j\}$ be the set of unordered pairs from A_k, with $|S_k| = k(k+1)/2$. Let $P^{(k)}$ be the $(k+1) \times k(k+1)/2$ matrix whose columns correspond to the elements of S_k: each column has exactly two ones, at locations i, j specified by the

pair of the given column (the other $k-1$ elements are zeros). The parity check matrix $H^{(k)}$ of the wire-tap code is obtained from $P^{(k)}$ by erasing any single row.

Theorem 1: The code from Construction 1 is a $[k(k+1)/2, k, k-1]$ binary wire-tap code.

Proof: Each row of $H^{(k)}$ has k ones. Assume that there exists a $k \times (n-\mu) = k \times (n-k+1)$ sub-matrix with rank less than k. Since $H^{(k)}$ is systematic, any set of $n-k+1$ columns has at least one column with a single one at some row i_1. Thus the dependent row combination of that sub-matrix does not include row i_1. The number of columns of $H^{(k)}$ that have a one in row i_1 and the other one in one of the remaining $k-1$ rows is $k-1$. So if we erase row i_1 from the chosen sub-matrix there has to be a column with a single one. We can then, by induction, continue to exclude and erase rows until there are no more rows left for the dependent combination. ■

The matrix $H^{(k)}$ resulting from Construction 1 has k ones in each row. Therefore the density of $H^{(k)}$ is k. Exactly k of the columns of $H^{(k)}$ have a single one, hence $H^{(k)}$ is given in systematic form. Given its wire-tap security of $\mu = k-1$, Construction 1 is next shown to have optimal density (i.e. the lowest possible density given μ), and consequently both optimal decoding and optimal encoding complexities.

Proposition 1: Any parity-check matrix H of a $[n, k, \mu]$ wire-tap code has density of at least $\mu + 1$.

Proof: If the density of H is strictly smaller than $\mu+1$, then there exists a row i of H with μ or fewer non-zeros. Therefore, if we pick the $k \times (n-\mu)$ sub-matrix of H that has all zeros in row i, it clearly has a rank smaller than k, in contradiction to the wire-tap requirement. ■

Note: the first order Reed-Muller codes [2, Ch.13] also enjoy optimal density for their wire-tap security, but their length is exponential in their dimension, making them appropriate only for the less realistic case of μ that is in the same order as the block length n.

Example 1: For $k = 4$ we have $A_k = \{1, 2, 3, 4, 5\}$ and

$$\begin{aligned} S_k \;=\; & \{\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{2,3\},\{2,4\},\{2,5\}, \\ & \{3,4\},\{3,5\},\{4,5\}\} \end{aligned}$$

Then the matrix $P^{(4)}$ is given by

$$P^{(4)} = \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0 \\ 1&0&0&0&1&1&1&0&0&0 \\ 0&1&0&0&1&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&0&1 \\ 0&0&0&1&0&0&1&0&1&1 \end{bmatrix}$$

which after erasing the top row results in

$$H^{(4)} = \begin{bmatrix} 1&0&0&0&1&1&1&0&0&0 \\ 0&1&0&0&1&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&0&1 \\ 0&0&0&1&0&0&1&0&1&1 \end{bmatrix} \tag{3}$$

The matrix $H^{(4)}$ in (3) defines a $[10, 4, 3]$ wire-tap code. The density of the code is 4.

For better wire-tap security with a slightly higher (optimal) density we include the following construction.

Construction 2: Let $B_k = \{1, 2, \ldots, k, k+1, k+2\}$ be the set of integers between 1 and $k+2$ (inclusive), and $T_k = \{\{i,j,l\} : i \in B_k, j \in B_k, l \in B_k, i \neq j, j \neq l, l \neq i,\}$ be the set of unordered triples from B_k, with $|T_k| = k(k+1)(k+2)/6$. Let $Q^{(k)}$ be the $(k+2) \times k(k+1)(k+2)/6$ matrix whose columns correspond to the elements of T_k: each column has exactly three ones, at locations i, j, l specified by the triple of the given column (the other $k-1$ elements are zeros). The parity check matrix $\mathcal{H}^{(k)}$ of the wire-tap code is obtained from $Q^{(k)}$ by erasing any two rows.

Theorem 2: For $k > 4$, the code from Construction 2 is a $[k(k+1)(k+2)/6, k, k(k+1)/2 - 1]$ binary wire-tap code.

Proof: The proof will show that for $k > 4$, any linear combination of rows of $\mathcal{H}^{(k)}$ gives a vector with Hamming weight of at least $k(k+1)/2$, the weight of individual rows of $\mathcal{H}^{(k)}$. The next lemma expresses the weight of row linear combinations as a function of the number of rows summed in the linear combination.

Lemma 1: Let $\boldsymbol{x}$ be a linear combination of ℓ rows from $\mathcal{H}^{(k)}$, where $1 \leqslant \ell \leqslant k$. Then the Hamming weight of $\boldsymbol{x}$ is given by

$$w_H(\boldsymbol{x}) = f_k(\ell) \triangleq \ell\tbinom{k+2-\ell}{2} + \tbinom{\ell}{3} \tag{4}$$

The weight of a linear combination of rows of $\mathcal{H}^{(k)}$ depends only on ℓ, the number of rows in the combination, and is given by the function $f_k(\ell)$ above.

Proof: The bit x_t of the vector $\boldsymbol{x}$ is one if the ℓ rows of $\mathcal{H}^{(k)}$ taken to obtain $\boldsymbol{x}$ have either 1 one or 3 ones at column t. The number of columns that have a single one in a given set of ℓ rows equals to the left summand in (4). Similarly, the right summand is the number of columns that have all three ones in the set of ℓ rows. ■

Given the closed-form expression for row-combinations' weights in the lemma above, we wish to prove that combinations of $\ell > 1$ rows have weights greater than or equal to the weight of a single row. For convenience, denote $\bar{k} = k+2$. Expanding $f_k(\ell)$ from (4) yields

$$f_k(\ell) = \tfrac{1}{6}\left[\ell^3 - 3(\bar{k}-1)\ell^2 + (3\bar{k}^2 - 6\bar{k} + 2)\ell\right]. \tag{5}$$

$f_k(\ell)$ is a third-degree polynomial whose derivative with respect to ℓ is

$$f_k'(\ell) = 2\ell^2 - 2\bar{k}\ell + \tfrac{\bar{k}^2}{2} - \tfrac{\bar{k}}{2} + \tfrac{1}{3}$$

Solving $f_k'(\ell) = 0$, we find the local minimum of f_k at

$$\ell_{\min} = \tfrac{1}{2}\left[\bar{k} + \sqrt{\bar{k} - \tfrac{2}{3}}\right] \tag{6}$$

(f_k has a local maximum as well, but since our interest is in finding the *lowest* weight linear combination, we focus on minima of f_k.)

The absolute minimum of f_k in the interval $1 \leqslant \ell \leqslant k$ is therefore at either $\ell = \ell_{\min}$ or $\ell = 1$. The claim of the theorem, to be proved shortly, is that for any $k > 4$, the local minimum is not lower than the value at the left boundary, i.e. $f_k(\ell_{\min}) \geqslant f_k(1)$.

Substituting $\ell_{\min}$ from (6) into $f_k(\ell)$ in (5) we get

$$f_k(\ell_{\min}) = \tfrac{\bar{k}^3}{12} - \tfrac{\bar{k}^2}{4} - \tfrac{\bar{k}\sqrt{\bar{k}-\frac{2}{3}}}{6} + \tfrac{\bar{k}}{6} + \tfrac{\sqrt{\bar{k}-\frac{2}{3}}}{9}$$

In comparison, evaluating f_k at $\ell = 1$ results in

$$f_k(1) = \tfrac{1}{2}(\bar{k}-1)(\bar{k}-2)$$

It is simple to check that when $\bar{k} > 7$, then $f_k(1) < f_k(\ell_{\min})$. That proves the theorem for $k > 5$. For the special case of $k = 5$, while the local minimum is lower than the value at $\ell = 1$, it is not attained at an integer value of ℓ. When examining values of f_5 only at integer argument values, it turns out that the absolute minimum is attained at $\ell = 1$, hence the proof applies to any $k > 4$. Note that for $k = 4$, in contrast, the

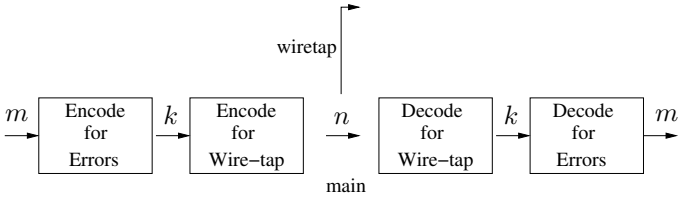

Fig. 1. Concatenation of outer error-correcting code with inner wire-tap code.

weight of a row of $\mathcal{H}^{(4)}$ is 10, while combinations of 4 rows result in vectors with weight of 8. ■

Constructions 1 and 2 can be generalized to taking all combinations of size $s > 3$ subsets. Proving that the resulting codes have optimal densities then reduces to using calculus (as in the proof of Theorem 2) to check that linear combinations of any ℓ rows give weights at least as large as the corresponding row weights.

PART 2: MAIN CHANNEL WITH ERRORS

So far in the paper error-correcting codes were used to randomize information and render it completely equivocal to a wire-tapper with partial observations. In this part, codes will be used both for wire-tap security and for error-correction on the main channel.

IV. CODE CONCATENATION WITH LOW DENSITY WIRE-TAP CODES

One way to combine wire-tap codes and error-correcting codes is to first add redundancy to the information for error-correctability, and then randomize the redundant information using a wire-tap code. This concatenation scheme has been implicitly proposed in [6]. The main idea is to, similarly to the error-free case, use an error-correcting code with sufficient rate to hide the chosen coset ($[a_1, \ldots, a_k]$ in (1)), but introduce additional redundancy by picking only a *subset* of the 2^k cosets to represent information. That way the secrecy properties of the wire-tap code are preserved, since the adversary still cannot distinguish between cosets, and in addition main-channel errors may be corrected by the redundancy of the decoded coset. In mathematical terms, let $\boldsymbol{c}$ be a random codeword from the code $\mathcal{C}$ with block length n and dimension $n-k$. Then the encoded vector is set as in (1), but for a channel with errors the vector $[a_1, \ldots, a_k]$ is obtained by

$$[a_1, \ldots, a_k] = [b_1, \ldots, b_m]G_2$$

where $b_1, \ldots, b_m$ are the information bits and G_2 is a $m \times k$ generator matrix of a linear code with length k and dimension m. So out of the dimension k space of cosets spanned by $\mathcal{C}$, we use only a dimension m subspace to represent information. Note that in the error-free case $m = k$ and the matrix G_2 is the $k \times k$ identity matrix. This scheme can be viewed as a concatenation of an outer error-correcting code with an inner wire-tap code, as depicted in Figure 1.

This form of concatenation is very convenient for the wire-tap security properties, as these are not affected by the addition of an outer code. However, the design of the error-correcting code is a challenge, since it needs to correct main-channel errors while using redundancy introduced "behind" the wire-tap code. The implication of this fact is that code design for the noisy main channel case, with security and error-correction guarantees, is a largely open problem. In the following we present a general construction for the joint problem that offers such security and error-correction guarantees for infinite families of codes. An essential building block in the construction is the code family of Construction 1, proposed in section III for the error-free case.

Construction 3: Let $\mathcal{C}$ be a code with parity-check matrix $H^{(k)}$ from Construction 1. Let G_2 be a $m \times k$ generator matrix of a code that corrects $2t$ errors. The encoded vector is calculated by a concatenation of the codes defined by G_2 (outer) and $H^{(k)}$ (inner).

Theorem 3: The code resulting from Construction 3 is a wire-tap code with $\mu = k - 1$, and can correct t errors, or $2t$ erasures (or combination thereof) over the main channel.

Proof: Each column of $H^{(k)}$ has two ones. Hence each error/erasure induces at most two errors/erasures, respectively, on the outer codeword. ■

Wire-tap codes with low column weights are essential for good error correctability in the concatenated scheme. If codes are sought with both minimal row weights (for optimal encoding and decoding complexities) and low column weights (for good error correction), then it is not hard to see that the constructions of section III provide optimal redundancies given their row and column weights.

V. BACKWARD CONCATENATION: OUTER WIRE-TAP AND INNER ERROR-CORRECTING CODES

Even with optimal-density wire-tap codes, the concatenated codes of section IV still pay a high (factor 2) penalty in their error-correction capability due to their wire-tap security requirement. Consequently, as the number of main-channel errors increases, the efficiency of that coding scheme degrades. In this section we propose an alternative coding scheme that is more suitable for main channels with a large number of errors, on the expense of more modest wire-tap security guarantees. A very natural way to allow redundancy-efficient error-correction capability is to reverse the concatenation of codes: implementing an outer wire-tap code and an inner error-correcting code. We now explain this *backward concatenation* scheme. Let G_1 be a $(n-k) \times n$ generator matrix of a wire-tap code. The output of the wire-tap encoder is set, as in the error-free case to

$$\boldsymbol{v} = \boldsymbol{u}G_1 + [a_1, \ldots, a_k, \underbrace{0, \ldots, 0}_{n-k}]$$

where $\boldsymbol{u}$ is a random row vector of length $n-k$, and $a_1, \ldots, a_k$ are information symbols. The wire-tap encoded vector is then input to an encoder of an error-correcting code with generator matrix G_2 of dimensions $n \times N$. The final encoded vector output from the error-correcting code's encoder is the length N vector

$$\boldsymbol{x} = \boldsymbol{v}G_2 = \boldsymbol{u}G_1G_2 + [a_1, \ldots, a_k, \underbrace{0, \ldots, 0}_{n-k}]G_2 \qquad (7)$$

This backward concatenation can be viewed pictorially in Figure 2.

Clearly, the legitimate receiver enjoys the full error-correction capability of the code defined by G_2. As long as the number of main-channel errors is within this correction capability, the legitimate receiver can first correct the errors and then decode the resulting length n wire-tap vector to recover

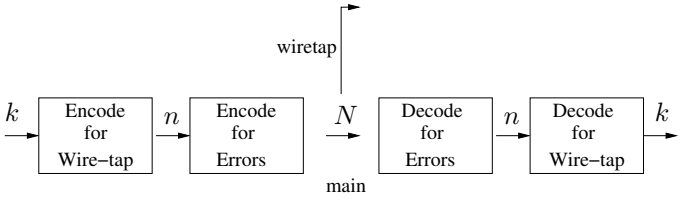

Fig. 2. Backward concatenation: outer wire-tap code with inner error-correcting code.

the information. However, the wire-tap guarantees provided by the code defined by G_1 are in general not preserved at the output of the error-correcting code encoder. The best way to see this fact is by an example. In the following example, we see that even a weak error-correcting code operating on the wire-tap encoded vector can completely destroy its security properties, despite the wire-tap code itself having very strong wire-tap security.

Example 2: Let $k = 1$ and let G_1 be the $(n-1) \times n$ generator matrix of the parity code. In section II we saw that code achieving wire-tap security of $\mu = n - 1$. Now suppose we want to use this wire-tap code over a main-channel with errors, and choose to protect it with the single-error detecting parity code (hence $N = n + 1$). Then if the adversary has access to the single parity bit of this error-detecting code, then he knows whether the weight of the wire-tap codeword is even or odd. But the parity of the wire-tap codeword is *exactly* the secret bit we try to conceal from the adversary. Therefore, the introduction of an error-correcting code dropped the wire-tap security from $\mu = n - 1$ to $\mu = 0$.

The way that the error-correcting code affects the wire-tap security is now examined algebraically. The adversary has access to parts of the length N vector $\boldsymbol{x}$ calculated in (7). Thus the wire-tap properties are not determined by the code defined by G_1, but rather by the code defined by G_1G_2. In the case of Example 2, $G = G_1G_2$ has a column of zeros, which implies that the dual of the combined code has minimum distance 1, thus $\mu = 0$ (recall the relation between the dual code's minimum distance and μ from the first paragraph of section III).

The lesson from Example 2 and the explanation that follows is that the matrix G_2 that generates the inner error-correcting code needs to be chosen with care. On one hand the code generated by G_2 should have good error-correction properties (i.e. large minimum Hamming distance). On the other hand, the combined code generated by G_1G_2 needs to have good wire-tap security properties. The ultimate design goal is to construct a code such that the code G_1G_2 has the same μ as the outer wire-tap code G_1. This is obviously the best one can hope for, since the adversary can choose exclusively from the n wire-tap code symbols of the combined length N code. We now present a construction that meets this goal of preserving the exact wire-tap security of the outer wire-tap code, for the case $\mu = 1$.

Construction 4: Let $G_1 = [1, \ldots, 1]$ be the $1 \times n$ generator matrix of the length n binary repetition code. The dual of the repetition code is the parity code with minimum distance 2, thus G_1 provides $\mu = 1$ wire-tap security. Let G_2 be a $n \times N$ generator matrix of a t error-correcting code, such that the columns of G_2 have odd weights.

Theorem 4: The code obtained by the concatenation of G_1 (wire-tap) and G_2 (error-correcting) codes in Construction 4 can correct t errors on the main channel and has wire-tap security of $\mu = 1$.

Proof: Since every column of G_2 has an odd weight, the product G_1G_2 equals the $1 \times N$ matrix $[1, \ldots, 1]$. As argued in Construction 4 for G_1, this matrix provides wire-tap security of $\mu = 1$. ∎

We next propose a construction for a family of error-correcting codes whose generator matrices have odd-weight columns. This construction is based on Construction 2.

Construction 5: Define a code by the generator matrix

$$G = [I_n | Q^{(n-2)}]$$

where I_n is the $n \times n$ identity matrix, $Q^{(i)}$ is the $(i+2) \times i(i+1)(i+2)/6$ matrix defined in Construction 2, and the $|$ operator represents horizontal concatenation of matrices. The length of the code is $N = n + n(n-1)(n-2)/6$ and its dimension is n. All of its columns have odd weights.

Proposition 2: For $n > 6$ the code from Construction 5 has minimum distance $(n-1)(n-2)/2 + 1$.

Proof: Similarly to the proof of Theorem 2 for Construction 2, it is possible to prove that for $n > 6$ the minimum weight of the code equals the weight of a single row. The number of ones in a row of $Q^{(n-2)}$ is $(n-1)(n-2)/2$, which together with the single one in a row of I_n gives the claimed minimum distance. ∎

If we use the generator matrix from Construction 5 as G_2 in Construction 4, we obtain a family of codes with k information bits and the following properties:

- Length $N = n + n(n-1)(n-2)/6$, where $n = k + 1$
- Wire-tap security $\mu = 1$
- Number of correctable errors on the main channel $t = \lfloor k(k-1)/4 \rfloor$

Note that an alternative construction for the backward concatenation scheme has appeared in [1], however that construction requires wire-tap codes that are complementary dual. The scheme proposed here works with optimal-complexity wire-tap codes that are in general not complementary dual.

VI. Acknowledgments

The authors wish to thank Martin Hassner for valuable discussions. This research was funded in part by the ERC Advanced Researcher Grant of A. Shokrollahi entitled ECC-SciEng.

References

[1] V. Korjik and D. Kushnir, "Key sharing based on the wire-tap channel type II concept with noisy main channel," in *Advances in Cryptology – ASIACRYPT 1996, Lecture Notes in Computer Science 1163*, Springer Berlin / Heidelberg, 1996, pp. 210–217.

[2] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.

[3] J. Massey, "Linear codes with complementary duals," *Discrete Mathematics*, vol. 106/107, pp. 337–342, 1992.

[4] L. Ozarow and A. D. Wyner, "Wire-tap channel II," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 10, pp. 2135–2157, Dec. 1984.

[5] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 28, pp. 656–715, Oct. 1949.

[6] A. Thangaraj, S. Dihidar, R. Calderbank, S. McLaughlin, and J. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Transactions on Information Theory*, vol. 53, no. 8, pp. 2933–2945, 2007.

[7] US Department of Defense, "DoD 5220.22-M, protection of classified information," *National Industrial Security Program Operating Manual*, 2006.

[8] V. Wei, "Generalized Hamming weights for linear codes," *IEEE Transactions on Information Theory*, vol. 37, no. 5, pp. 1412–1418, 1991.

[9] A. D. Wyner, "The wire-tap channel," *Bell Systems Technical Journal*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.