FLOW WATERMARKING WITH VARSHAMOV–TENENGOL’TS CODES IN NETWORKS WITH PACKET LOSSES

An invisible flow watermarking QIM scheme based on linear error-correcting codes for channels with substitution and deletion errors has been proposed. The evaluation of scheme demonstrates similar to [1] performance, but with lower complexity, as soon as its implementation is mainly based on linear decoding operations.

Introduction

Recently [1], an active approach of traffic analysis called “flow watermarking” has been considered. This approach attempts to manipulate the statistical properties of packets flow to insert a watermark making it easier to detect the flow after passing through one or more relay hosts. To prevent an attacker to tolerate the packet delays and to eliminate embedded watermark, recent schemes have concentrated on making them “invisible”. This technique has been the subject of increased interest in the past decade, because it requires low computational and communication cost while providing high accuracy in linking traffic flows. In a inter-packet-delay (IPD) flow watermarking the watermarks are embedded into the time intervals between arrivals of packets. A novel IPD-based flow watermarking scheme that can withstand packet losses has been proposed in [1]. In this scheme the watermark embedding is done with the use of quantization index modulation (QIM) [2]. To withstand packet losses authors develop a Hidden-Markov Model (HMM) decoding scheme considering the communication channel with dependent deletion and substitution errors. However, the proposed watermark detector based on a maximum likelihood decoding algorithm paired with a forward-backward algorithm is of high complexity and requires a lot of computational resources. In this paper we propose the alternative IPD-flow watermarking QIM scheme, based on the use of linear error-correcting codes and Varshamov–Tenengol’ts [3] (VT-codes) to reduce the complexity of flow watermarking method [1].

Linear error-correcting codes

The error-correcting codes used today in watermarking are mainly binary. A very good survey of the theory of error-correcting codes is done in [4] and the only necessary definitions is used throughout this paper. Generally a binary code $C$ is defined as a set of finite sequences (vectors) $x = (x_1 \ldots x_n)$ called codewords encoded with the use of corresponding message vectors $b = (b_1 \ldots b_k)$ from code symbols $x_i, b_i \in GF(2)$.

Linear $[n,k,d]$-code is also defined by its parameters: Hamming distance between any binary codewords $d(x_i; x_j)$, weight of a codeword $wt(x_i)$ and a code rate $R = k/n$. Any linear code $C$ is completely defined by its generator matrix $G$ or parity-check matrix $H$ whose rows and columns are matrices respectively linearly independent. Every codeword of a linear block code $C$ is a linear combination of the rows of a generator matrix $G$. Since $G$ is $k \times n$ matrix and has rank $k$, the representation of $C$ is unique. The parity-check matrix $H$ could be found from $G$ by finding $n - k$ linearly independent solutions of the linear equation $G \cdot H^T = 0$. These equations are easy to solve when $G$ is systematic. The prescribed error correction capacity $t$ of linear error-correcting codes strictly depends on its minimum distance $d_{min} = \min \{d(x_i; x_j)\}$ and weight distribution of a code. To perform an error correction in codeword $y$, corrupted by $t$ or less errors, a highly efficient method of syndrome
decoding could be applied. It consists of the following steps: the calculation of syndrome for a received word $y$

$$S = y \cdot H^T,$$  

(1)

search for a most plausible error pattern $e$, the estimation of transmitted codeword $x'$. Decoder picks error pattern $e$ of smallest weight satisfying $e \cdot H^T = S$. All procedures of syndrome decoding are linear and only step 2 requires a nonlinear operation that can be performed by look-up tables. For example, the linear [6,3,3]-code $C = \{\{000000\}, \{110100\}, \{011010\}, \{101110\}, \{101001\}, \{011101\}, \{110011\}, \{000111\}\}$ is completely defined by its generator matrix $G$ [5, p.357-367] and with its eight syndromes can correct six 1-bit error patterns and one 2-bit error pattern $e$. To change its properties, a binary code can be easily modified by different techniques [4]. The number of its codewords can be increased or decreased. The process of deleting a codeword from the basis of $C$ to obtain a new code $C'$, where the minimum weight of remains the same, is referred to as taking a subcode of $C$.

It is known that linear codes as other error correcting codes are applied for channels with substitution errors when transmitted symbols are received as the other symbols. However, there are channels that suffer from synchronization errors, which are associated with not receiving transmitted symbols leading to deletion errors. Therefore, there is a compelling reason to consider codes, that not only correct substitution errors, but can also recover from deletion errors. Recently it has been proved [5] that linear codes and all cyclic codes, except for repetition codes, of rates greater than 1/2 cannot correct a single deletion or a single insertion.

As opposed to [5] the use of binary linear codes for the correction of both types of errors by the same code with the application of two different decoders will be proposed. We start the next section from the definition of VT-codes and show how to get a subcode of a linear error-correcting code to combat with substitution and deletion errors.

**Varshamov–Tenengol’ts codes**

Given a parameter $a$ with $0 \leq a \leq n$ the Varshamov–Tenengol’ts (VT) code $VT_a(n)$ is the set of binary words $x = (x_1 \ldots x_n)$ of length $n$ that satisfies equality [4]:

$$\sum_{i=1}^{n} ix_i \equiv a(\mod(n+1)).$$  

(2)

These codes are single error correcting codes and for $a=0$ are optimal as it was conjectured in [3], and will be discussed in this paper.

For example, after calculation $\sum_{i=1}^{6} ix_i \equiv 0(\mod 5)$ the code $VT_0(6)$ with block length $n = 6$ is $VT_0(6)=\{(000000), (001100), (010010), (011110), (100001), (101101), (110011), (110100), (111111)\}$. Any code $VT_0(n)$ can be used to communicate reliably over a channel that introduces at most one deletion in a block of length $n$. Levenshtein proposed a simple decoding algorithm [6] for a VT code, which is reproduced below. Assume the channel code $VT_0(n)$ is used.

1. Suppose a codeword $x \in VT_0(n)$ is transmitted over the channel, the bit in position $p$ is deleted and $y$ is received. Let there be $L_0$ zeros and $L_1$ ones to the left of the deleted bit, and $R_0$ zeros and $R_1$ ones to the right of the deleted bit (with $p = 1 + L_0 + L_1$).

2. Compute the weight $wt(y) = L_1 + R_1$ of $y$ and the checksum $\sum_{i} iy_i$. If the deleted bit is 0, the new checksum is $R_1(\leq wt(y))$ less than it was before. If the deleted bit is 1, the new checksum is $p + R_1 = 1 + L_0 + L_1 + R_1 = 1 + wt(y) + L_0(> wt(y))$ less than it was before.

3. Hence, if the deficiency in the checksum, say $D$, is less than or equal to $wt(y)$ we know that a 0 was deleted, and we restore it just to the left of the rightmost $R_1$ ones. Otherwise a 1 was deleted and we restore it just to the right of the leftmost $L_0$ zeros.
As an example [6], assume the code $VT_0(6)$ is used and $x = (110100) \in VT_0(6)$ is transmitted over the channel. If the first bit in $x$ is deleted and $y = (10100)$ is received, then the new checksum is 4, and the deficiency $D = 7 - 4 = 3 > wt(y) = 2$. The decoder inserts a 1 after $n - D = 3$ zeros from the right to get $(110100)$. Thus a very simple algorithm of low complexity can be used to decode $VT_0(n)$ with deletion correction. However, in general the $VT_0(n)$ codes are nonlinear and the dimension $k$ is to get a linear $[n,k]$ codes is bounded by $k \leq \lfloor n/2 \rfloor$ [5]. We use this result and propose an approach to find a linear substitution and deletion correction code from VT-code. The algorithm contains of the following steps: organize the codewords of $VT_0(n)$ code in a lexicographically order; choose the $k_0$ linear independent codewords of maximum weight preserving $d(x_i; x_j) \geq d_{\text{min}}$; produce $G$ and $H$ of $C$ making the linear combinations of chosen VT-codewords. The use of this algorithm results in a subcode $C'$ that has at least $k_0 + 1$ codewords of $VT_0(n)$ code as soon as the linear combination of any codeword with itself makes a codeword $(0 \ldots 0)$, which is also a codeword of $VT_0(n)$ code. Considering the use of the proposed above algorithm, the flowing generator and parity check matrixes for a modified [6,3,3]-code $C'$ have been made:

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad H' = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (3)$$

The use of $G'$ and $H'$ from (3) results in a code set $C'$ with an increased number of codewords that belong to $VT_0(6)$, compared with $C$. If we prune the codewords that are not the codewords of $VT_0(6)$, we can make a subcode with the necessary properties $C^* = \{(000000), (110100), (011110), (101101), (110011)\}$ consisting of 5 codewords. $C^*$ is a linear code with $d_{\text{min}} = 3$, at the same time it is a $VT_0(6)$ subcode and that is why it can be used for error correction of one substitution and one deletion error. The examination of $C^*$ has shown, that its code rate is reduced to approximately by 1/2 relatively to the code rate of $C$. The algorithm proposed below can be applied to an arbitrary code to find the error-correcting VT-code (EC-VTC) that is a subcode of a linear code.

**Theorem:** If generator matrix of a linear error-correcting code $[n,k]$ has at least one nonzero codeword of $VT_0(n)$ code, then it is the linear error correcting VT-subcode.

The proof has been presented above with the help of algorithm and matrices (3) proposed.

However, to perform the independent decoding of codewords from EC-VTC, placed in a continuous bit stream, the boundaries of codewords must be known. We implement the independent decoding of them by accurately making a set of codewords $C^*$ from EC-VTC and inserting the periodic markers between the codewords as discussed in the next section.

**Scheme for flow watermarking**

We use the described above linear codes in a flow watermarking [1] considering the channel with deletion and substitution errors. The proposed watermark embedding scheme, depicted in Figure 1, has the same embedder and extractor based on QIM [2], but the different coding and decoding principles are used.

For the watermark embedding the flow of IPDs is modified with the use of QIM watermarking (Fig.1). A quantization step size $\Delta$, which is the distance between two quantizers, is used for QIM modulation:

$$I_i^w = \begin{cases} c\Delta, & \text{if } s_i = 0 \\ (c + 0.5)\Delta, & \text{if } s_i = 1 \end{cases}. \quad (4)$$

As packets can only be delayed by QIM Embedder, we choose parameter $c$ to be the smallest integer so such that the change in $I_i^w$ would delay the $i$-th packet. Then $I^w$ is transmitted and after the transfer over the network it is received in the form of estimated sequence of IPDs $I$ and received by the QIM Extractor.
For the flow $\hat{I}$ processed by QIM Extractor, the following QIM demodulation function is used to recover the embedded bits $\hat{s}$:

$$\hat{s} = \begin{cases} \text{mod}(|2\hat{I}_i/\Delta|, 2), & \text{if } 2\hat{I}_i/\Delta - |2\hat{I}_i/\Delta| \leq 0.5 \\ \text{mod}(|2\hat{I}_i/\Delta|, 2), & \text{if } 2\hat{I}_i/\Delta - |2\hat{I}_i/\Delta| > 0.5 \end{cases}$$

(5)

The embedding and extracting steps with possible IPDs distortion are presented in Figure 2.

As it was discussed the scheme in Figure 1 may be regarded as a communication channel with two types of errors: substitutions and deletions. The substitution error refers to bit flips due to network jitters or packet deletions that result in merger of two IPDs. It has been shown that network jitter may be approximated as independently identically distributed Laplace random variables with zero mean and a standard deviation of $\sigma$. Since during QIM demodulation we map each IPD to its closest quantizer, any jitter over $\Delta/4$ would possibly result in a substitution error (see Figure 2). The channel model developed in [1] handles the dependent substitution and deletion errors. However, to simplify the decoding we assume that the dependence exists only inside the received codeword, which is a reasonable limitation, as soon as any number of packet deletions results only in the presence or in the absence of a substitution error.

For example, in Figure 2 four packets 0, 1, 2, 3 are sent, three packets 0, 2, 3 are received and packet 1 is lost. The first two IPDs $I_1$ and $I_2$ are transformed into $\hat{I}_1$ and the size of last IPD $I_2$ is changed and evaluated as $\hat{I}_2$. Hence the result of channel noise is the bit received before Packet 2 that is merger of the two intervals $\hat{s}_1 = s_1 \oplus s_2$ and the bit flipped after receiving Packet 3 resulting in $\hat{s}_2 = \bar{s}_3$. In general $\hat{s}_i = \sum_{j=r+1}^i s_j$ and can take only 0 or 1 binary values, where $r$ is the index of the last successfully received packet before $i$-th one.

Without loss of generality we consider the packet deletion probability $P_d$ and the packet substitution probability $P_s$ to be identical for all packets and assume that Packet 0 is always synchronized. This assumption allows scheme to be in the synchronized state before the decoding procedure and further to evaluate the distance between $w$ and of $\hat{w}$ and to decide whether the watermark is present. Considering that EC-VTC decoder is synchronized prior to decoding of a received sequence $\hat{s}$ we describe its operation principles below.
EC-VTC encoder and decoder

The original watermark $d$ is a sequence of data bits with each element $d_i \in GF(2)$. It is xored with pseudo-random sequence $k_w$, making a sequence $w = w_1w_2 \ldots w_N$, which becomes an input of an EC-VTC Encoder. This block transforms sequence $w$ of length $N$ into the corresponding number $|C^*|$ of codewords from EC-VTC. Note, that $|C^*| \geq 2^l$. In the proposed scheme we apply the encoding function $f$, which makes a bijective map of input binary symbols with length $l$ into the codewords of $C^*$, i.e. the arbitrary sequence $w$ is divided into blocks or vectors $b = (b_1 \ldots b_l)$ to produce the VT-codewords $x$ of length $n$.

Then a VT-codeword $x$ is concatenated with predefined marker pattern $z = z_1z_2 \ldots z_m$ of length $m$ in Marker Embedder, making a codeword $x_m$. The input sequence $s$ is made from concatenation of $N$ codewords $x_m$, has length $M = (n + m)N$ and is embedded in flow IPDs. $I^w$ is transmitted and after transversing the network is received in the form of estimated sequence $\hat{I}$ and demodulated. The result sequence $\hat{s}$ made from codeword $\hat{x}_m$ with inserted markers is then converted by Marker Detector into the sequence of VT-codewords $y$ with possible deletion errors. The algorithm of marker search finds the best markers positions performing the analysis of data sample from the input sequence $\hat{s}$. Marker Detector finds sequences of bits between the markers, which correspond to VT-codewords $y$ with possible deletion and/or substitution errors. The EC-VTC Decoder performs the error-correcting decoding using one of two algorithms, depending on the number of errors in $y$ occurred. The decision about the decoder type to be applied is based on the estimation of $y$ length. If the only one deletion is found, the Levenshtein’s decoding algorithm [6] is used, and if the number of deletion errors is greater than one, the maximum likelihood decoding (MLD) is applied.

Actually, to decode the received codeword $y$ there must be a $g$ function that performs the reverse mapping $b = g(y)$. As soon as the deletion channel is exploited, the length of $y$ is not necessary equal to the length of $x$. Hence, the decoding function must also perform the decoding of all words with length $v < n$. Due to the fact that the received vector $y$ may be of the arbitrary length $v$, then it would be the case, when $v < n - 1$, i.e. the number of deletions is above the error correction capacity of EC-VTC code used. Hence, the properties of $g$ must be extended for the case, when there are more than one deletion errors. To make a decoder of a relatively low complexity, the look-up table for MLD decoding is applied. Let $y = y_1, \ldots, y_v$ to be a received sequence. Note, that it should be $v$ groups of binary sequences, containing $C^l v \leq v!/j!(v - j)!$ versions of $j$ bit deletions from $y$ for $1 < v < n - 1$, that are distributed between $v$ tables. Then a MLD decoder performs a search of codeword $x^* \in C^*$ in a corresponding table, for which $Pr(x^* = x/y)$ is maximized. Further EC-VTC Decoder outputs the estimation $\hat{b}$ of a block $b$ encoded, which contains the estimated version of watermark $\hat{w}$. Sequence $\hat{w}$ of length $N$ then xored with a key-sequence $k_w$ of a same length, resulting in a sequence $\hat{d}$.

For example, if $d=1100...$ xoring with $k_w = 1010...$, the first two bits of sequence $w = w_1w_2 \ldots w_N$ are $w_1w_2 = 01$. With the use of EC-VTC $C^s = \{(110100), (110011), (011110), (101101)\}$, the codeword $x = (110011)$ is obtained which, with appended marker pattern 000, results in $x_m = (110011000)$. Then if $\hat{s} = 11011000...$ is received with second bit deleted and third flipped, then a marker detection is easily performed and $y = (11011)$ is obtained and, after deletion and substitution error correction, results in $w_1w_2 = 01, d_1d_2 = 11$. However, if the first and second bits are deleted in $x$ and the received codeword $y = (0011)$, the decoder would perform MLD decoding.

Performance evaluation

The proposed watermarking scheme has been evaluated by simulation of packets, generated from independent Poisson process of rate 3 packets per second and length of about 4000 packets with the shifted mean of 25 ms and the standard deviation of 10 ms. The pseudorandom bits of watermarks have been encoded by subcode $C^s$ with added uniform marker $z=000$ and randomly embedded into 3600 flows with the use QIM modulation (4). The watermark parameters were taken similar to the
values from [1] to get approximately the same number of watermark bits as \( N=50, n+m=9, M=450 \). Note, that the block length of EC-VTC [6,3,3] with appended \( z \) marker bits results in \( n+m=9 \) and close to the sparsified version [1] of watermark. The watermark extraction was made with the use of QIM demodulation function (5) and evaluation was performed with the use of Levenshtein’s, MLD and syndrome decoders.

The evaluation of the proposed scheme against packet deletions by considering the varying packet deletion probabilities \( P_d=\{0.01, 0.02, 0.03, 0.1\} \). The watermarks were randomly embedded into 3600 flows. Also the other 3600 unmarked flows were used to obtain the false positive rates. The detection threshold was chosen such that the false positive rate was kept below 1\% for all deletion probabilities presents the detection results. True Positive (TP) detection rates for deletion ratios \( P_d \) that have the corresponding values: 1\% – 0.9999; 2\% – 0.9998; 3\% – 0.9998; 10\% – 0.9942; 20\% – 0.6621.

We see that the detector has rather high true positive rates, maintaining true positive rate (TP) up to 99\%, even when less than 10\% of packets were deleted. However the value of TP drops to 66\% when packet deletion ratio is at 20\%, which is rare in a network environment. Thus, in comparison with the other IPD-based watermarking schemes [1], which suffer from desynchronization, the proposed scheme is robust against packet losses and network jitters.

**Conclusion**

An invisible flow watermarking scheme based on linear error correcting codes for channels with substitution and deletion errors, representing network jitter and packet drops, has been developed. The described scheme is based on relatively low-rate linear code, formed on the basis of proposed algorithm to create a linear error-correcting code that is a subcode of VT-code. Statistical experiments demonstrate that proposed scheme is of similar to [2] performance, but has a much lower complexity, as soon as it uses a simpler implementation, mainly based on linear decoding operations.

**List of references**


_Assanovich Boris Alekseievich_, Associate Professor of Information Systems and Technologies Department, Yanka Kupala State University of Grodno, PhD, bas@grsu.by.

_Gavrilova Irina Leonidovna_, Senior Lecturer of Industrial Electronics Department, Yanka Kupala State University of Grodno, iren_k@tut.by.

_Karacharskaya Margarita Leonidovna_, 1st category engineer of Industrial Electronics Department, Yanka Kupala State University of Grodno, m.karacharskaya@grsu.by.

_Baufalik Marta Vladimirovna_, 4th year student of Physics and Technical Faculty, Yanka Kupala State University of Grodno, mbiit0914@mail.ru.