Use of Linear Error-Correcting Subcodes in Flow Watermarking for Channels with Substitution and Deletion Errors

Boris Assanovich, William Puech, Iuliia Tkachenko

Use of Linear Error-Correcting Subcodes in Flow Watermarking for Channels with Substitution and Deletion Errors

Boris Assanovich[1], William Puech[2] and Iuliia Tkachenko[2,3]

[1] YK State University of Grodno, Belarus
bas@grsu.by

[2] LIRMM, UMR CNRS 5506
University of Montpellier 2, France
{William.Puech, Iuliia.Tkachenko}@lirmm.fr

[3] Authentication Industries, Montpellier, France

Abstract. An invisible flow watermarking QIM scheme based on linear error-correcting subcodes for channels with substitution and deletion errors is proposed in this paper. The evaluation of scheme demonstrates similar to known scheme performance but with lower complexity as soon as its implementation is mainly based on linear decoding operations.

Keywords. Flow watermarking, inter-packet-delay, deletion and substitution errors, linear error-correcting codes, VT-codes, quantization index modulation.

1. Introduction

Recently, 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.

Flow watermarking is also classified as *interval*-based and *inter-packet-delay* (IPD)-based. The first type of watermarking technique is robust to packet losses but is vulnerable to the *multi-flow attack* [1]. The IPD-based flow watermarking, in which the watermarks are embedded into the time intervals between arrivals of packets, resists this attack [2]. The drawback of this scheme is that it can cause a lot of errors in decoding during the loss of packet synchronization in the watermark detection process.

A novel IPD-based flow watermarking scheme that can withstand packet losses has been done in [2]. In this scheme the watermark embedding has been done with the use of quantization index modulation (QIM) [3]. In this approach the embedded marks are invisible. 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 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 (VT) codes [4] to reduce the complexity of flow watermarking method [2].

2. Error Correcting Codes

2.1 Linear Error-Correcting Codes

Most practical error-correcting codes used today in watermarking are binary. Linearity is an important structural property of codes, allowing a concise representation of codes, the accompanying encoding and decoding rules as well as the determination of the errors are correctable/detectable. A very good survey of the theory of error-correcting codes is done in [5] and the only necessary definitions are used throughout a paper. The symbols of binary linear codes are the elements of a field $GF(2)=\{0;1\}$ which is a code alphabet. Generally, a binary code C is defined as a set of finite sequences (vectors) $\mathbf{x}=(x_1...x_n)$, called codewords, encoded with the use of corresponding message vectors $\mathbf{b}=(b_1...b_k)$ from code symbols x_i, $b_i \in GF(2)$.

Linear $[n,k,d]$-code is defined by following parameters: Hamming distance between any binary codewords $d(\mathbf{x}_i;\mathbf{x}_j)$, weight of a codeword $wt(\mathbf{x}_i)$ and a code rate $R=k/n$. Any linear code C is completely defined by generator matrix $\mathbf{G}$ and parity-check matrix $\mathbf{H}$ whose columns and rows are respectively linearly independent. Every codeword of a linear block code C is a linear combination of the rows of a generator matrix $\mathbf{G}$. The error correction capacity t of linear error-correcting codes strictly depends on its minimum distance $d_{min}=\min\{d(\mathbf{x}_i;\mathbf{x}_j)\}$ and weight distribution of a code. To perform an error correction in codeword $\mathbf{y}$, corrupted by t or less errors, a rather simple method of bounded distance decoding with syndrome could be applied. It consists of following steps: the calculation of syndrome for a received word $\mathbf{y}$

$$\mathbf{S}=\mathbf{y}\cdot\mathbf{H}^{\mathrm{T}}, \tag{1}$$

search for a most plausible error pattern $\mathbf{e}$, the estimation of transmitted codeword $\mathbf{x}'$. Decoder picks error pattern $\mathbf{e}$ of smallest weight satisfying $\mathbf{e}\cdot\mathbf{H}^{\mathrm{T}}=\mathbf{S}$. All procedures of decoding with syndrome are linear and only the second step 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 $\mathbf{G}$ [5, p.357-367]. To change its properties, a binary code can be easily modified by different techniques [5]. 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 the other error correcting 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 [6] that linear codes and all cyclic codes can correct a single deletion or insertion error but not by both types of errors [6].

As opposed to [6] the application of binary linear codes for the correction of both types of errors by the same subcodes with the use of two different decoders will be proposed. In Section 2.2, we define the VT-codes and show how to get a subcode of a linear error-correcting code to combat with substitution and deletion errors.

2.2 VT-Codes for Deletion and Substitution Errors Correction

Given a parameter a, with $0 \le a \le n$, the Varshamov-Tenegol'ts (VT) code $VT_a(n)$ is the set of binary words $\mathbf{x}=(x_1 \ldots. x_n)$ of length n so that the equality satisfies [4]:

$$\sum_{i=1}^{n} i x_i \equiv a(\mathrm{mod}(n+1)). \tag{2}$$

These codes are single error correcting codes and optimal for $a=0$ as it was conjectured in [4, 7] and will be discussed in this paper.

For example, after calculation $\sum_{i=1}^{n} ix_i \equiv 0 (\mathrm{mod}\, 7)$ 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 [8] based on the deficiency in checksum and weight calculation for a VT code. Assume the channel code $VT_0(n)$ is used.

As an example, assume the code $VT_0(6)$ is used and $\mathbf{x}$=(110100)$\in VT_0(6)$ is transmitted over the channel. If the first bit in $\mathbf{x}$ is deleted and $\mathbf{y}$=(10100) is received, then the new checksum is 4, and the deficiency D=7−4=3>$wt(\mathbf{y})$=2. The decoder inserts a 1 after n-D=3 0's 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 $[n,k]$ is bounded by $k \le \lfloor n/2 \rfloor$ [6]. 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 linear independent codewords of maximum weight preserving $d(\mathbf{x}_i;\mathbf{x}_j) \ge d_{min}$; produce $\mathbf{G}$ and $\mathbf{H}$ 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+1 codewords of $VT_0(n)$ code as soon as the linear combination of any codeword with itself makes a codeword (0…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:

$$\mathbf{G}' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \ \mathbf{H}' = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{3}$$

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 $VT_0(6)$ and a codeword with all zeros, we can make a subcode with the necessary properties C^*={(110100), (011110), (101101), (110011)} consisting of 4 codewords. C^* is a linear subcode with d_{min}=3, at the same time it is a $VT_0(6)$ code and 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 ½ 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 C^* of a linear code. For example, there is a EC-VTC, coinciding with linear error correcting code [8,2,5] [5, p.378], consisting of 4 codewords and subcoding $VT_0(8)$. It is easily seen that it corrects one deletion and two substitution errors. It is known that the size of any $VT_0(n)$ is about $2^n/n$ [6], then an additional limitation on its linear properties leads to a reduction in its rate of at least by ½.

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^r from EC-VTC with possible reduction of a code rate R and inserting the periodic markers between the codewords as discussed in Section 3.

3. System Model

3.1 Scheme for Flow Watermarking

We use the described above linear codes in a flow watermarking [1], [2] 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.

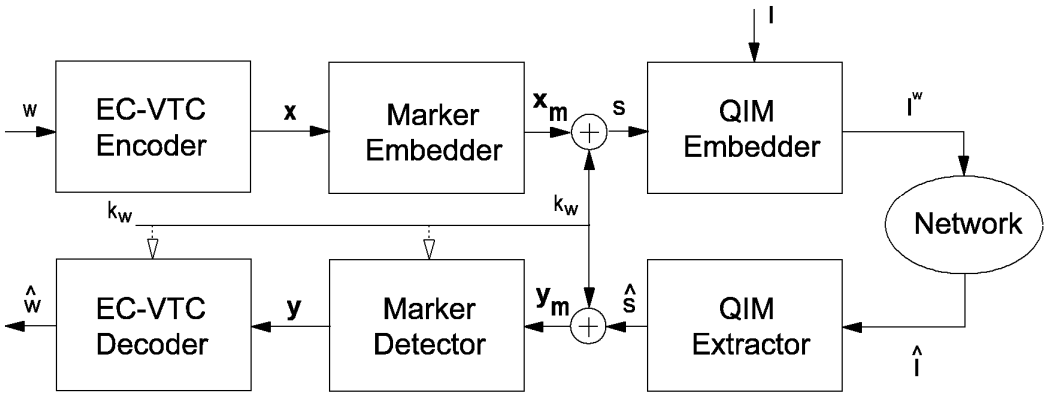

Fig. 1. System model

3.2 QIM Embedder and Extractor

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

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

As packets can only be delayed by QIM Embedder, we choose parameter c to be the smallest integer so 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 $\hat{I}$ 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 $\hat{s}$:

$$\hat{s}_i = \begin{cases} \text{mod}(\lfloor 2\hat{I}_i/\Delta \rfloor, 2) & \text{if} \quad 2\hat{I}_i/\Delta - \lfloor 2\hat{I}_i/\Delta \rfloor \le 0.5 \\ \text{mod}(\lceil 2\hat{I}_i/\Delta \rceil, 2) & \text{if} \quad 2\hat{I}_i/\Delta - \lfloor 2\hat{I}_i/\Delta \rfloor > 0.5 \end{cases}. \tag{5}$$

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

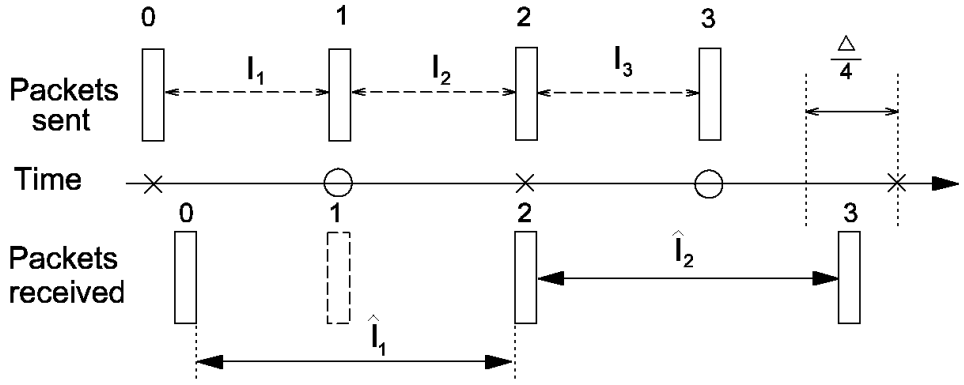

Fig. 2. An example of IPDs distortion caused by jitters

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. 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 [2] 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 the merged of the two intervals $\hat{s}_1 = s_1 \oplus s_2$ and the bit flipped after receiving Packet 3 resulting in $\hat{s}_2 = \overline{s}_3$. 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 can be easily performed by the use of so-called frame synchronization based on special markers [6, 9] or on one or more codewords received without errors. This allows the scheme to be in the synchronized state before the decoding procedure and further evaluate the distance between w and of $\hat{w}$ 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.

3.3 EC-VTC Decoder

The original watermark $w=w_1w_2\ldots w_N$ is a sequence of bits with each element from *GF*(2). This sequence is divided into blocks $\mathbf{b}=(b_1\ldots b_l)$ to produce the VT-codewords **x** of length *n* by VTC Encoder. Then a codeword **x** is concatenated with predefined marker pattern $z=z_1z_2\ldots z_m$ of length *m* making $\mathbf{x_m}$ and xored with pseudo-random key sequence k_w, forming a sequence *s*, as depicted in Fig.1. The used key k_w is a sparse sequence containing a binary 1 only in one position of block with length $n+m$ and is applied for security and frame synchronization. Actually sequence *s* is made from concatenation of *N* codewords $\mathbf{x_m}$, 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 $\hat{s}$ is xored with key sequence k_w, separated into codewords y_{m} by marker detection [9] and further converted into VT-codewords **y** containing possible substitution or/and deletion 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 [8] is used, and if the number of deletion errors is greater than one, the maximum likelihood decoding (MLD) maximizing $\Pr(\mathbf{x}^* = \mathbf{x}/\mathbf{y})$ is applied. The bounded distance decoding with the use of syndrome calculation is performed in case of absence of deletion errors or after their correction as well in process of frame synchronization.

For example, consider that a key generator outputs the sequence of digits g=03… and a key sequence is made according to the expression $k_w=g$ mod $(n+1)$. The received and extracted by demodulator sequence $\hat{s}$=110100000.11110000 is then xored with key sequence k_w=000000000.001000000 and decoded with the use of mapping the subcode C^r={(110100), (110011), (011110), (101101)} to data blocks **b**={00,01,10,11}. Hence, after marker detection, 2 codewords are obtained $\mathbf{y_1}$=110100, $\mathbf{y_2}$=11010. The calculation of (1) gives S=0 that can be used to indicate about the correct boundaries of received error-free codeword $\mathbf{y_1}$, infer the presence of synchronization and allow to start decoding by applying the Levenshtein's algorithm for the deleted zero bit in the last position of $\mathbf{y_2}$. However, if the second bit is also deleted and $\mathbf{y_2}$=1101, the evaluation of its length results in the use of MLD decoding. The use of VT-code as a subcode of any linear code preserves its minimum distance d_{min} and does not change code performance in channel with substitution errors which is well analyzed in [5]. However, the reduction in probability of codeword error when using MLD strictly depends on subcode used and on weight distribution $wt(\mathbf{x}_i)$ of its codewords.

4. Performance Evaluation

In this section we evaluate the robustness to packet losses. 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 shifted mean of 25 ms and standard deviation of 10 ms. Network jitters was simulated as Laplace distribution with zero mean and the same deviation. The pseudorandom bits of watermarks have been encoded by subcode C^r 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 [2] to get the approximately the same number of watermark bits as N=50, n=9, M=450. Note, that the block length of EC-VTC [6,3,3] with appended z marker bits results in n=9 and close to the sparsified version [2] of watermark. The watermark extraction was made with the use of QIM demodulation function (5) and the decoding 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, 0.2} has been done. The watermarks were randomly embeddered into 3600 flows. Also the other 3660 unmarked flows were used to obtain the false positive rates. The detection threshold was chosen so that the false positive rate was kept below 1% for all deletion probabilities. True Positive (TP) detection rates for deletion ratios P_d gave corresponding values: 1% - 0.9999, 2% - 0.9998, 3% - 0.9998, 10% - 0.9951; 20% - 0.6655.

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], [10] which suffer from desynchronization, the proposed scheme is robust against packet losses and network jitters presented as deletion and substitution errors. The use of pseudo-random key sequence on transmission side improves the security of overall scheme and provides the frame synchronization in watermarking system. To examine the visibility of proposed scheme, the Kolmogorov-Smirnov (K-S) has been performed on 4500 watermarked flows against 4500 unwatermarked flows and demonstrated the statistical invisibility of watermark according to the values of K-S distances that are below 0.03. Obviously, to defeat the multi-flow attack, as suggested in [1] the use of random function position of embedding positions can selected within the described above synchronization method.

5. 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 and computational 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 with much less space and time complexity and only perform MLD with the use of look-up tables when the packet loss increases significantly.

References

1. N. Kiyavash, A. Houmansadr and N. Borisov. Multi-flow attacks against network flow watermarking schemes. *USENIX Security Symposium*, pp. 307-320 (2008)
2. X. Gong, M. Rodrigues and N. Kiyavash. Invisible flow watermarks for channels with dependent substitution and deletion errors. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Proc.,* Kyoto, Japan, March 25-30. pp. 1773-1776 (2012)
3. B. Chen and G. W.Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. Inf. Th.* 47, pp. 1423-1443 (2001)
4. G. M. Tenengol'ts. Class of codes correcting bit loss and errors in the preceding bit. *Avtomat. Telemekh.* 37, 5, pp. 797-802 (1976)
5. B. Sklar. 2001. *Digital Communications: Fundamentals and Applications.* 2nd ed. Prentice-Hall. 2003.
6. K. A. S. Abdel-Ghaffar, H. C. Ferreira and L. Cheng. Correcting Deletions Using Linear and Cyclic Codes. *IEEE Trans. Inf. Th.* 56,10, pp. 5223-5234 (2010)
7. R. P. Varshamov and G. M. Tenengol'ts. Correction code for single asymmetric errors. *Avtomat. Telemekh.* 26. 2, pp. 286-290 (1965)
8. V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady*. 10, 8, pp.707-710.
9. J. Chen, M. Mitzenmacher, C. Ng and N. Varnica. Concatenated codes for deletion channels. In *Proceedings of the 2003 IEEE International Symposium on Information Theory.* Yokohama, Japan, June 29-July, p. 218 (2003)
10. A. Houmansadr, N. Kiyavash and N. Borisov. RAINBOW: A Robust and Invisible Non-Blind watermark for network flows. In *Proceedings of the 16th Annual Network & Distributed System Security Symposium.* San Diego, USA, Feb. 8-11 (2009)