Watermarking Pattern Recognition in Channels with Substitution and Bursty Insertion and Deletion Errors

Boris Assanovich [1], Vicenc Almenar Terre [2] Felipe L. Penaranda-Foix [3]

1) GrSU, Oszeshko 22, Grodno 230023, Belarus, bas@grsu.by, www.grsu.by

2) UPV, Camino de Vera s/n. 46022, Valencia, Spain, valmenar@dcom.upv.es, www.upv.es

3) UPV, Camino de Vera s/n. 46022, Valencia, Spain, fpenaran@dcom.upv.es, www.upv.es

***Abstract**: A novel watermarking scheme for intrusion tracing in networks based on QIM modulation and linear error-correcting subcodes suitable for pattern recognition and combating with substitution, deletion and insertion errors in channels with invisible embedded watermarks is proposed in this paper. Statistical and computational experiments demonstrate that proposed scheme outperforms the known coded IPD-based flow watermarking schemes and has a significant advantage in detection rate and implementation complexity, but slightly inferior to the code rate, and hence to processing time for intrusion tracing.*

***Keywords*:** Flow watermarking, interleaving, inter-packet-delay, linear codes, VT-codes, quantization index modulation.

1. INTRODUCTION

Intrusion detection systems attempts to detect intrusion through analysing observed system or network traffic with the use of watermark tracing [1]. If the embedded watermark is both robust and unique, the watermarked back traffic can be effectively recognized and traced across intermediate nodes. This active approach of traffic analysis is often called "flow watermarking. To prevent an attacker to tolerate the packet delays and to eliminate embedded watermark, recent schemes of "flow watermarking" have concentrated on making them "invisible". The stepping-stone detection scenario with "flow watermarking" is depicted in Figure 1 where Attacker wants to attack a Victim without exposing his identity. Attacker remotely logins to a compromised intermediate host via SSH [2]. Tracing packet flows sent to Victim's machine would implicate Stepping Stone instead of Attacker. In real life, attackers may hide behind a chain of stepping stones, making it hard to determine the origin of the attack. Fortunately, flow watermarking is a possible solution for tracing back the attack source, revealing the attack path.

Flow watermarking is also classified as *interval*-based and *inter-packet-delay* (IPD)-based. The *interval*-based watermarking technique is robust to packet losses but is vulnerable to the *multi-flow attack* [3].

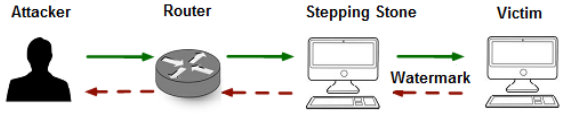

Fig. 1. Stepping-stone detection scenario

The *inter-packet-delay* (IPD)-based flow watermarking, in which the watermarks are embedded into the time intervals between arrivals of packets, resists this attack but causes a lot of errors in decoding during the loss of packet synchronization.

A novel IPD-based flow watermarking scheme that can withstand packet losses as well as packet insertions has been designed in [4]. In this scheme the watermark embedding has been done with the use of quantization index modulation (QIM) [5]. To withstand packet drops and packet insertions authors develop a Hidden-Markov Model (HMM) decoding scheme considering the communication channel with dependent deletion, substitution and insertion errors. However, the proposed watermark detector based on a maximum likelihood decoding (MLD) technique paired with a forward-backward algorithm is of high complexity.

In this paper we propose the alternative IPD-flow watermarking QIM scheme modification, which differs from [6] by the use of a matrix code interleaving and the application of linear codes with increased minimum Hamming distance. This approach allows not only to improve the system performance but also to achieve better noise immunity with respect to [4] at a relatively low complexity (by eliminating the MLD decoding).

2. ERROR CORRECTION

2.1. Linear Error-Correcting codes

In this section we briefly describe the linear codes and introduce the necessary notation. A very good survey of the theory of error-correcting codes is done in [7] 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, $b_i \in GF(2)$. Any linear code C is completely defined by generator matrix G and parity-check matrix H. To perform an error correction in codeword y, corrupted by t or less errors, a rather simple method of bounded distance decoding with syndrome S could be applied. It consists of following steps: the calculation of syndrome for a received word y

$$\mathbf{S}=\mathbf{y}\cdot\mathbf{H}^{\mathbf{T}}, \qquad (1)$$

search for a most plausible error pattern e, the estimation of transmitted codeword $\mathbf{x}'$. Decoder picks error pattern e of smallest weight satisfying $\mathbf{e}\cdot\mathbf{H}^{\mathbf{T}}=\mathbf{S}$. All procedures of decoding with syndrome S are linear and only the second step requires a nonlinear operation that can be performed by look-up tables.

It is known that linear codes as the other error correcting codes are applied for channels with substitution

errors. However, there are channels that suffer from synchronization errors, which are associated with not receiving transmitted symbols or receiving addition extra symbols leading to deletion or/and the insertion errors. Some of these codes, known as VT codes [8] allow to correct errors of this type. Although it was recently demonstrated [9] the possibility of two types error correction by special codes design, in this work we simplify the solution of the problem by finding the subset of a linear code codewords, that has the ability to fix both types of errors. In Section 2.2, we recall the VT-codes and show how to get a subcode of a linear error-correcting code to combat with substitution, insertion and deletion errors.

2.2. Error-Correcting VT-Codes

Given a parameter a, with $0 \le a \le n$, the Varshamov-Tenegol'ts (VT) code $VT_a(n)$ is the set of binary codewords $x=(x_1 \ldots x_n)$ of length n so that the equality satisfies [8]:

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

These codes are single error correcting codes and optimal for a=0 as it was conjectured in [8] and will be discussed in this paper.

For example, after calculation $\sum_{i=1}^{n} i x_i \equiv 0 (\mathrm{mod} 7)$ 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 (insertion) in a block of length n. Levenshtein proposed a simple decoding algorithm [10] based on the deficiency in checksum and weight calculation for a VT code.

As an example, 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).

However, in general the $VT_0(n)$ codes are nonlinear and the dimension k to get a linear (n,k) code is bounded by $k \le \lfloor n/2 \rfloor$ [9]. We use this result and propose an algorithm [6] to find a linear substitution and deletion (insertion) correction code from VT-code. The algorithm produce matrixes G and H of C making the linear combinations of chosen VT-codewords. For example, the matrixes for a modified (6,3,3)-code C' are:

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

The use of G' and H' from (3) according to proposed algorithm results in obtaining the codewords of $VT_0(6)$ C^*={(110100), (011110), (101101), (110011)}. A code set C^* is a linear subcode with d_{min}=3 and can be used for error correction of one substitution and one deletion error. The algorithm proposed [6] 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] [7, p.378], consisting of 4 codewords and subcoding $VT_0(8)$. In our case it is convenient to select the subcode contains only two codewords C^*= (01111110; 11110001). It is easily seen that it corrects one deletion and two substitution errors.

In the proposed scheme, a EC-VTC is capable of correcting various errors of Binary Substitution, Insertion, Deletion (BSID) channel [11], however to perform the independent decoding of codewords from linear VT subcode placed in a continuous bit stream the boundaries of codewords must be known. We implement the independent decoding of them by accurately organizing a set of codewords from EC-VTC with possible reduction of a code rate R and inserting the periodic markers between the codewords [12] as discussed in Section 3.

3. SYSTEM MODEL

Despite the fact that communication channel used for flow watermarking does not coincide with BSID and has partially dependent errors, as will be shown below, in this paper we will use an effective mechanisms for errors separation, in particular interleaving [13], and apply more powerful error-correcting codes to combat with error bursts. The proposed watermark embedding scheme, depicted in Figure 2 with slightly changed structure and has a decoding principle based only on bounded distance decoding with syndrome calculation. Hence, the use of interleaving and deinterleaving in this scheme allow to correct not only bit substitutions but also both insertion and deletion burst errors compared with the previous scheme.

The proposed scheme comprises several layers. The pseudo-random key k_w known to both sides is used for security purposes. The network environment in which this flow watermarking structure used is characterized by the presence

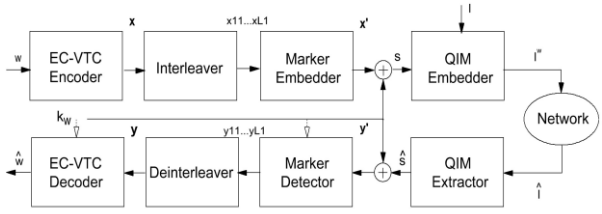

Fig.2. System model

of jitter, loss and the appearance of new IPD packets. According to the known results the IPD jitter can be approximated as independent identically distributed (iid) variable with zero-mean and Laplacian pdf [4], [14].

3.1. QIM Embedder and Extractor

For the watermark embedding the flow of IPDs is modified with the use of QIM watermaking (Fig.2). A quantization step size Δ, which is the distance between two quantizers, is used for QIM modulation:

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

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 $\hat{s}$:

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

The embedding and extracting steps with possible IPDs distortion are presented in Figure 3.

As it was discussed before, the scheme in Figure 2 may be regarded as BSID channel. The substitution error refers to bit flips due to network jitters. However, several packet deletions that result in merger of two IPDs may also lead to errors. Since during QIM demodulation we map each IPD to its closest quantizer, any jitter over Δ/4 would possibly result in a substitution error (see Figure 3).

The channel model developed in [3] handles the dependent substitution and deletion errors. In our case, we moved to the channel with independent errors after interleaving, as well as using the fact that error dependence exists only inside the received codeword. For example, in Figure 3 four packets 0, 1, 2, 3 are sent, three packets 0, 2, 3 are received, packet 1 is lost and new packets 4 and 5 are added.

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

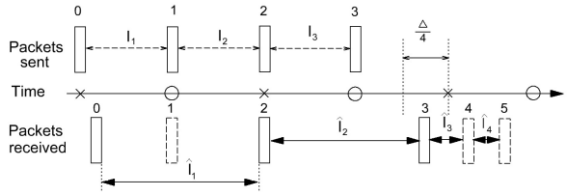

Fig.3 An example of IPDs distortion caused by jitters

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. As it is seen from Figure 3 two intervals s_3, s_4 are appeared resulting in insertion of new bits in the received watermarked data.

Without loss of generality, we 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], [12] or on one or more codewords received without errors and decoded with zero syndrome. This allows the scheme to be in the synchronized state before the decoding and mapping procedures and further evaluate the distance between w and $\hat{w}$ to decide whether the watermark is present. It is also possible to apply a different approach to design the synchronized watermark decoding, based on the separation of frames made from individual codewords using another unique special markers [12].

We assume the existence of the channel state when there are no errors and use the syndrome value calculation to determine correctly the beginning of the codeword with transmitted watermark data, as it is done in [15]. Considering that EC-VTC decoder is synchronized prior to decoding of a received sequence $\hat{s}$ we describe its operation principles below.

3.2. EC-VTC Decoder

In this section we consider the case of the matrix interleaving application, although the interleaver may optionally be disabled, depending on the channel state. More information about the decoder work description without interleaving can be found in [6].

We consider the original watermark $w=w_1w_2...w_N$ is a sequence of bits with each element from *GF*(2). This sequence is divided into blocks **b**=($b_1...b_l$) to produce the VT-codewords **x** of length *n* by VTC Encoder. Next we assume that a codeword **x** consists of bits with indexing ($x_{11}x_{12}...x_{1n}$). Hence, every bit of **x** is indexed by a pair of positive integers as x_{ij}. Clearly, $1 \le i \le L$ and $1 \le j \le n$, where *L* is an interleaving depth. A super codeword of length N_i to be transmitted can be presented by $L \times n$ interleaving matrix. To perform interleaving the codewords are put in matrix row by row and read from it column by column.

Further we denote the group of column matrix bits as **x'** and concatenate it with predefined marker pattern $z=z_1z_2...z_m$ of length *m* making $\mathbf{x'_m}$ xored with pseudo-random key sequence k_w, forming a sequence *s*, as depicted in Fig.2. The used key k_w is a sparse sequence containing binary ones only in several positions of a block with length $L+m$ and is applied for security and frame synchronization [6]. Actually sequence *s* is the concatenation of $N=L\times Q$ codewords, where *Q* is a number of interleaving operations, and has length $M = (L+m)\cdot Q \cdot n$ and is embedded in flow IPDs. After the modulation 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 by marker detection and further converted into VT-codewords **y** containing possible substitution or/and deletion and insertion errors. The EC-VTC decoder performs the error-correcting decoding using only bounded distance syndrome decoding, regardless of the error number in **y** that have occurred.

To simplify the decoder after the decoding operation of obtained data block **b*** that does not belong to watermarking dictionary, the random extraction of a permissible block **b'** from the dictionary is applied. If there are no substitutions and only one deletion or insertion occur, the Levenshtein's decoding algorithm [11] can be successfully used. However, if the burst errors occur, then matrix interleaving allows to distribute errors among the codewords of EC-VTC and to apply the independent decoding with correction both substitution, insertion and deletions errors.

4. PERFORMANCE EVALUATION

In this section we evaluate the robustness to packet drops and splits causing deletions (and possibly substitutions) and insertions in the presence of IPD jitter. The proposed watermarking scheme has been evaluated by simulation of packets, generated from independent Poisson process of rate 3.3 packets per second [14] and length of about 20000 synthetic packets with shifted mean of 25 ms and standard deviation of 10 ms. Network jitter was simulated as Laplace distribution shown for different deviations in Fig. 4. The pseudorandom bits of watermarks have been encoded by the use of 3 coding scenarios and then were randomly embedded into synthetic flows with the use QIM (4): 1) the codewords of EC-VTC (6,3,3) code with added uniform marker z=000 representing 2 bits per codeword with code rate R=2/9; 2) the codewords of interleaved EC-VTC (6,3,3) code with the interleaving matrix 9x6 whose columns are separated by uniform marker z=000 with the same code rate R=2/9; 3) the codewords of EC-VTC (8,2,5) code with the use of uniform marker z=0000 without interleaving and code rate R=1/12.

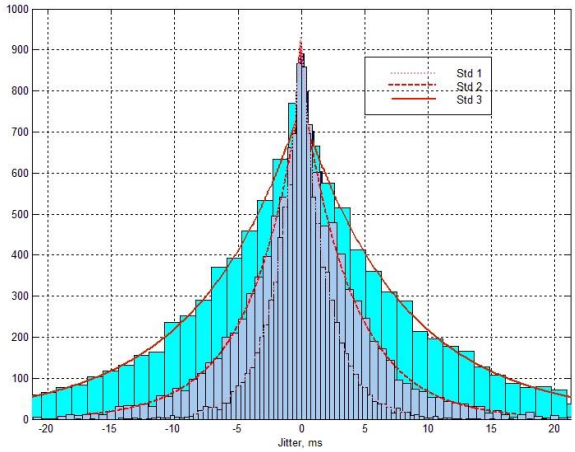

Fig.4. Simulated jitters and Laplace distributions

The watermark parameters were taken similar to the values from [6] and flow watermarking scheme [4]. The watermark extraction used QIM demodulation function (5) and the decoding was performed with the use of Levenshtein's algorithm, syndrome decoding and random mapping. The simulation of the watermark detection performance with the use of Kolmogorov-Smirnov (KS) test has shown that the watermark data was detected with accuracy about 98% for all jitter deviations in case of all simulated quantization levels. Furthermore, the simulation of 3 different coding schemes described above was performed to evaluate the fidelity of watermark data decoding with the use of Watermark Error Rate (WER) which was calculated according to the number of watermark bits in the erroneous codeword of applied linear code. The results obtained are depicted in Fig. 5.

To organize the simulation of synthetic channel with the error characteristics close to real network environment about 30% of substitution errors followed sequentially after deletion errors without bit insertions have been generated.

Thus, burst errors were not mimicked by any particular algorithm, and appeared randomly. Obviously, this is reflected in the chart, which shows the significantly better performance of linear code (8,2,5) as compared with other coding schemes and slight improvement in decoding performance through the use of interleaving for code (6,3,3) with depth L=6.

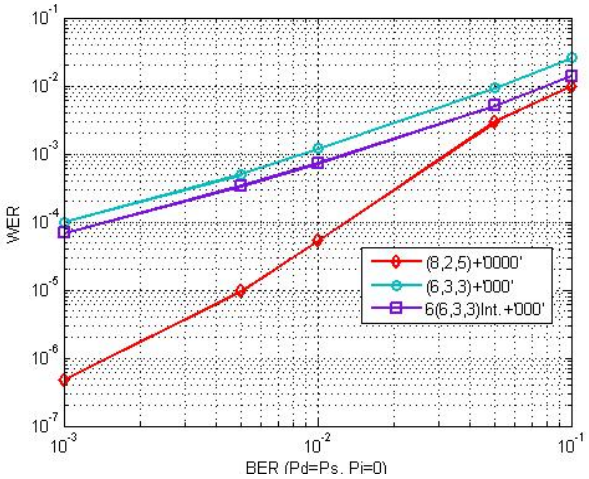

Fig. 5. Simulation results of WER of three different coding schemes

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} with fixed insertion probability P_i = 0.01 has been done for linear code (8,5,2) thus yielding to a minimum value of WER. 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 expressed in percentages (with fixed insertion probability P_i = 0.01) adopted the following corresponding numeric values: 1%- 0.999, 2%- 0.999 5%- 0.993, 10%- 0.990. The improvement compared to [4] and [6], is obviously achieved through the use of a more powerful error-correcting code (8,2,5) that have rather low code rate 1/12.

In simulation experiments we obtained the high true positive rates up to 0.99, even when about 10% of packets were deleted. However the value of TP drops to 15% when packet deletion ratio is at 20%, which is rare in a network environment. To examine the visibility of proposed scheme, the Kolmogorov-Smirnov (K-S) has been performed on watermarked and unwatermarked flows and demonstrated the statistical invisibility of watermark according to the values of K-S distances that are below 0.03.

5. CONCLUSION

An invisible flow watermarking scheme based on linear error correcting codes for channels with substitution, insertion and deletion errors, representing network jitter, packet drops and splits, has been developed. The described scheme is based on relatively low-rate linear error-correcting code, formed on the basis of proposed algorithm with a creation of a linear code that is a subcode of VT-code. Statistical and computational experiments have demonstrate that proposed scheme outperforms the scheme from [4] and [6] and has a significant advantage over the prior art [6] in detection rate and implementation complexity, but slightly inferior to the code rate, and hence to processing time for intrusion tracing.

6. REFERENCES

[1] X. Wang, D. Reeves, S. F. Wu and J. Yuill. Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework", *Proc. 16th Int'l Conf. Information*, pp. 369–384, 2001.

[2] RFC4252. The secure shell (SSH) authentication protocol.

[3] N. Kiyavash, A. Houmansadr and N. Borisov. Multi-flow attacks against network flow watermarking schemes. *USENIX Security Symposium*, pp. 307-320, 2008.

[4] X. Gong, M. Rodrigues, N. Kiyavash. Invisible Flow Watermarks for Channels with Dependent Substitution, Deletion, and Bursty Insertion Errors. *IEEE Transactions on Information Forensics and Security* 8(11), pp. 1850-1859, 2013.

[5] 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.

[6] B. Assanovich, W. Puech, I. Tkachenko. Use of Linear Error-Correcting Subcodes in Flow Watermarking for Channels with Substitution and Deletion Errors. In *Proceedings of the Intern. Conf. on CMS*, Magdeburg, Germany, Sept. 25-26, pp. 105-112, 2013.

[7] B. Sklar. 2001. Digital Communications: Fundamentals and Applications. 2nd ed. Prentice-Hall. 2003.

[8] R. P. Varshamov and G. M. Tenengol'ts. Correction code for single asymmetric errors. *Avtomat. Telemekh.* 26. 2, pp. 286-290, 1965.

[9] 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.

[10] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady*. 10, 8, pp.707-710, 1966.

[11] Buttigieg and J. A. Briffa. Codebook and marker sequence design for synchronization-correcting codes. in *Proc. IEEE Int. Symp. Inform. Theory*. St. Petersburg, Russia, Jul. 31–Aug. 5, pp. 1504–1508, 2011.

[12] J. Chen, M. Mitzenmacher, C. Ng and N. Varnica. Concatenated codes for deletion channels. In *Proc. of the 2003 IEEE Intern. Symp. on Inform. Theory.* Yokohama, Japan, Jun. 29-Jul. 4, p. 218, 2003.

[13] L. Cheng, T. G. Swart, H. C. Ferreira and K.A.S. Abdel-Ghaffar. Codes for Correcting Three or More Adjacent Deletions or Insertions. *Proc. IEEE International Symposium on Information Theory*, Honolulu, Hawaii, USA, June 29 - July 4, 2014.

[14] 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.

[15] S. Houcke and G. Sicot. Blind Frame Synchronization for Block Code. In *Proceedings of EUSIPCO, European Sig. Proc. Conf*, Florence, Italy, Sept., 2006.