INVISIBLE FLOW WATERMARKS FOR CHANNELS WITH DEPENDENT SUBSTITUTION AND DELETION ERRORS

Xun Gong[†]*, Mavis Rodrigues*[†]*, Negar Kiyavash*[‡]

[†]Department of Electrical and Computer Engineering
[‡]Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
xungong1,mrodrig5,kiyavash@illinois.edu

ABSTRACT

Flow watermarking [1] is an efficient technique for linking packet flows that helps thwart various attacks in networks such as over the Internet. Current state-of-the-art watermarking schemes withstand packet losses at the expense of compromising invisibility. We present an invisible flow watermarking scheme that can endure large numbers of packet losses. To maintain invisibility, our scheme embeds quantization-index modulation watermarks into inter-packet delays (as opposed to intervals). As the watermark is injected within individual packets, packet losses may lead to watermark desynchronization and substitution errors. To deal with this issue we propose a maximum likelihood decoding (ML) scheme based on a hidden-Markov model (HMM) of the channel. Experimental results demonstrate that our scheme is robust to both network jitters and packet deletions while remaining invisible to an attacker.

Index Terms— network flow watermarking, deletion channels, hidden-Markov models

1. MOTIVATION

Detecting correlated network flows, aka flow linking, is a crucial technique in traffic analysis, specially in protecting against cyber-attacks. For instance, an attacker can defeat an anonymous system such as Tor[2] by matching the end flows. Moreover, linking flows can help expose, stepping stone attacker, i.e., intruders that use intermediate hosts to attack a network system. Recent work has shown that in spite of encrypted content, similarities in communication patterns such as packet sizes and timings can be used for flow linking [1, 2, 3, 4]. Two types of traffic analysis techniques *passive* and *active* are commonly used. Passive analysis like [1] makes use of the original characteristics in a packet flow, which is quite sensitive to network artifacts such as jitters and packet drops and requires a large number of observed packets for successful detection. Active analysis schemes on the other hand are able to perform reliable detection with shorter flows by injecting patterns such as watermarks into flows [2, 3, 4].

There are mainly two classes of flow watermarking approaches; *interval*-based and *inter-packet-delay* (IPD)-based. In interval-based schemes, the flow is first divided into fixed lengths of time intervals. Then timing patterns of all packets within an interval are reshaped to encode the watermark. For instance, in [3], when a '0' is embedded, all packets in a selected interval are squeezed into the subsequent interval. Since the watermark pattern is embedded within multiple packets, interval-based schemes are robust to packet losses. However, shifting packets in groups causes visible 'artifacts' that in turn can reveal the embedded watermark. Kiyavash et al. [5] showed that interval-based watermarking schemes are vulnerable to the *multi-flow attack* where upon observing a few flows, the attacker can detect the watermark as abnormally large number of empty time periods are created during the embedding process. Fortunately, the alternative solution, IPD-based flow watermarks resists this attack. In IPD-based schemes [2, 4], watermark bits are embedded into the inter-arrival times of packets in a flow-dependent manner. Thus, it is hard for the attacker to find noticeable artifacts even when with access to many watermarked flows. The drawback of this per-packet-embedding is that it requires synchronizations of packets when for watermark detection and therefore, packet losses could cause severe decoding errors.

In this paper, we propose a novel IPD-based flow watermarking scheme that can withstand packet losses. We embed the watermark within the IPDs using quantization index modulation (QIM) [6], that is invisible even under the multi-flow attack. To withstand packet losses that may lead to both deletion and substitution errors we develop a hidden-Markov model (HMM) for our channel with dependent deletion and substitution error states. At the detector, a maximum likelihood decoding algorithm paired with a forward-backward algorithm for deriving the posterior probabilities is used. Through simulations, we show that our scheme performs well in presence of both deletion and substitution errors. It is noteworthy that the substitution errors are either introduced

[1] aka one-bit watermarking due to the absence of hidden messages
[2] http://www.torproject.org

by network jitters or packet deletions within the network that desynchronize the watermark and merge consecutive IPDs.

2. SYSTEM MODEL

In this section, we describe the components of the proposed scheme. Figure 1 depicts our embedding and extraction procedures.

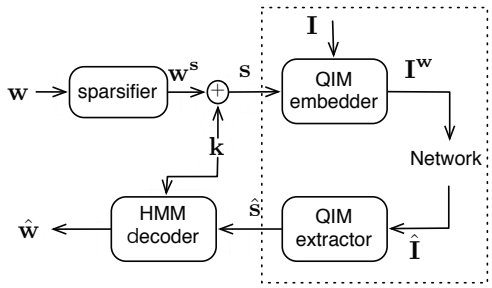


Fig. 1. Overview of our watermarking scheme.

The following notations are used throughout the paper.

- $\mathbf{w}, \mathbf{w^s}, \hat{\mathbf{w}}$: The original watermark $\mathbf{w}$ is a binary sequence of length N. $\mathbf{w^s}$ is a sparsified version of $\mathbf{w}$ extended to length $M = nN$ for an appropriate integer n. $\hat{\mathbf{w}}$ is the estimate of $\mathbf{w}$ extracted at the detector.
- $\mathbf{k}$: a length M pseudo-random binary sequence (key) available both at the embedder and the detector.
- $\mathbf{s}, \hat{\mathbf{s}}$: $\mathbf{s}$ is the length M sequence that is embed in flow IPDs and $\hat{\mathbf{s}}$ denotes the estimate of $\mathbf{s}$ at the detector.
- $\mathbf{I}, \mathbf{I^w}, \hat{\mathbf{I}}$: $\mathbf{I}$ is the IPD sequence in the original flow, $\mathbf{I^w}$ is its watermarked version. $\hat{\mathbf{I}}$ denotes the IPD sequence received after transversing the network.

2.1. Sparsification

In the first step of embedding, the binary watermark $\mathbf{w}$ is sparsified, by mapping each bit of $\mathbf{w}$ to a longer binary sequence of length n according to a deterministic sparsification table. The resulting sequence $\mathbf{w^s}$, is xored with a key $\mathbf{k}$ resulting in $\mathbf{s}$ which is embedded into the flow $\mathbf{I}$ using QIM.

The sequence $\mathbf{k}$ serves as a 'helper' for watermark synchronization [7]. The intuition is that changes in the 'pattern' of $\mathbf{k}$ provide information about deletions that occured. For instance, consider the case when $\mathbf{w^s}$ is all-zeros, if $\mathbf{k}$ is '0111001001' and '01100101' were received, it is easy to conclude that a '1' in the second run and a '0' in the fifth run were deleted. In practice, any '1's in $\mathbf{w^s}$ will create a bit flip in $\mathbf{k}$. Furthermore, the network could introduce more substitution errors. Therefore to retain the patterns of $\mathbf{k}$ necessary for synchronization, we need to ensure that $\mathbf{w^s}$ is sparse[3]. We denote the density with f and it is a parameter of the scheme also available at the extractor.

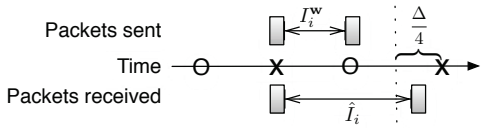


Fig. 2. An example for substitution errors caused by IPD jitters. '$\mathbf{x}$'s are '0' quantizers and '$\mathbf{o}$'s are '1' quantizers. The bit embedded on I_i is '1', but the decoded bit from $\hat{I}_i$ is '0'.

2.2. QIM Embedder and Extractor

In the next step, we modify the IPDs in the original flow using QIM watermarking. We pick a quantization step size Δ, which is the distance between two consecutive '0' quantizers. If s_i is '0', the IPD I_i, is changed to $I_i^{\mathbf{w}} = c\Delta$. Otherwise, $I_i^{\mathbf{w}}$ is set to $(c + 0.5)\Delta$. As packets can only be delayed, we choose c to be the smallest integer such that the change in $I_i^{\mathbf{w}}$ would delay the i th packet.

Once the flow $\hat{\mathbf{I}}$ is received at the detector, the following QIM decoding function is used to recover the embedded bits $\hat{\mathbf{s}}$.

$$\hat{s}_i = \begin{cases} \mod(\lfloor \frac{2\hat{I}_i}{\Delta} \rfloor, 2) & \text{if } \frac{2\hat{I}_i}{\Delta} - \lfloor \frac{2\hat{I}_i}{\Delta} \rfloor \leq 0.5 \\ \mod(\lceil \frac{2\hat{I}_i}{\Delta} \rceil, 2) & \text{if } \frac{2\hat{I}_i}{\Delta} - \lfloor \frac{2\hat{I}_i}{\Delta} \rfloor > 0.5 \end{cases} \quad (1)$$

2.3. HMM Decoder

At the HMM decoder, we first develop a hidden-Markov model of the channel. Based on this model the posterior probabilities $P(\hat{\mathbf{s}}|w_j)$ are calculated. Watermark bits w_j are subsequently decoded as $\hat{w}_j$ using ML decoding.

Note that in Figure 1 the QIM embedder, the network, and the QIM extractor may be regarded as a communication channel (within the dashed box) with two types of errors: *substitutions* and *deletions*. The substitution error refers to a bit flip due to either network jitters or deletions that result in merger of two IPDs. It has been shown that the network jitter may be approximated as independently identically distributed Laplace random variables with zero mean and a standard deviation of σ [4] . Since during QIM decoding we map each IPD to its closest quantizer, any jitter over $\Delta/4$ would possibly result in a substitution error (see Figure 2). In general, the probability of a substitution error caused by jitters can be

[3]Note that the choice of sparsification factor trades off the rate of the watermark and the detection performance. In most flow linking applications, rate is not of concern and a large sparsification factor may be picked.

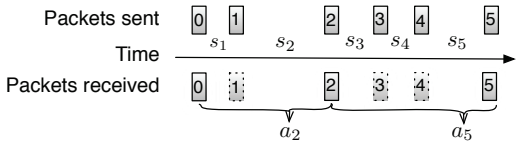


Fig. 3. Merging of IPDs when packets are dropped.

estimated as[4]

$$P_s \approx 2 - (1 + sgn(\frac{\Delta}{4}) \cdot (1 - e^{\frac{-|\Delta|}{2\sqrt{2}\sigma}})). \quad (2)$$

The deletion error refers to a bit lost due to packet drops. Davey and Mackay [7] proposed a probabilistic decoding scheme to handle independent deletion and substitution errors in a communication channel. Our channel differs from the model in [7] as a single packet drop results in merger of two consecutive IPDs. For instance, in Figure 3, the deletion of Packet 1 merges the bit s_1 and s_2 into $s_1 \oplus s_2$. This causes a deletion of s_1 and possibly a substitution error of s_2 in the received stream. Therefore, we develop a new channel model to handle dependent substitution and deletion errors. Without loss of generality, we consider the packet deletion probability P_d to be identical for all packets, and assume that Packet 0 is always synchronized[5].

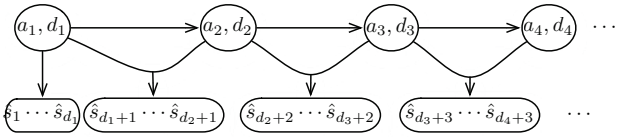


Fig. 4. The HMM of the watermark-over-network channel.

Figure 4 depicts the HMM model of our channel. The hidden state are defined as (a_i, d_i), where $i = 1, 2, \cdots M$. d_i is the *drift* of packet i in the received flow. If k packets were dropped before packet i, then d_i equals to '$-k$'. For example, in Figure 3, Packet 2 has drift of '-1' due to the loss of Packet 1, and Packet 5 has drift of '-3' because of the loss of three previous packets. a_i is the *accumulated* bit when sending Packet i. Again in Figure 3, before transmitting Packet 2, the bit in the current IPD is a merger of s_1 and s_2, that is $a_2 = s_1 \oplus s_2$. Similarly, $a_5 = s_3 \oplus s_4 \oplus s_5$. In general, a_i equals to $\sum_{j=r+1}^{i} s_j$, where r is the index of the last successfully received packet before Packet i. The observed states of Figure 4 are the received bits $\hat{\mathbf{s}}$. Note that the watermark extractor receives $\max\{d_i + i - 1, 0\}$ bits in total before Packet i is sent.

Posterior probabilities of this HMM model are required for ML decoder that extracts the watermark estimate $\hat{\mathbf{w}}$. In interest of space and brevity of presentation, we use an example to illustrate how this quantity is computed. In Figure 3, when sending Packet 3, the hidden state is $(d_3 = -1, a_3 = s_3)$. If Packet 3 is lost (with probability P_d), no new bit is transmitted, i.e., $\hat{s}_{d_3+3}^{d_4+3}$ is the empty sequence, $\emptyset$. The next state is $(d_4 = d_3 - 1, a_4 = a_3 \oplus k_4 \oplus w_4^s)$ (since $s_4 = k_4 \oplus w_4^s$). When the sparsified sequence density f is small, w_4^s may be modeled as a Bernoulli with parameter f. Therefore, the one-step transition $P(\emptyset, a_3 \oplus k_4 \oplus 1, d_3 - 1 | a_3, d_3)$ is given by fP_d, and $P(\emptyset, a_3 \oplus k_4 \oplus 0, d_3 - 1 | a_3, d_3)$ equals to $(1-f)P_d$. Similarly, transition probabilities to other states can be calculated in terms of density f, deletion probability P_d and the substitution probability P_s of (2) (we need to consider the possibility of occurrence of a substitution event when the current packet is successfully received).

Once the complete HMM model is in place, the standard forward-backward algorithm may be used to calculate the posterior probabilities $P(\hat{\mathbf{s}}|w_j)$ for $j = 1, 2, \cdots, N$ which are fed to ML decoder to extract the watermark estimate $\hat{\mathbf{w}}$. Finally, the distance between $\hat{\mathbf{w}}$ and the original watermark $\mathbf{w}$ is compared to a threshold to decide whether the watermark is present.

3. EVALUATION

We evaluated our watermarking scheme on network traffic flows generated from independent Poisson processes of rate of $\lambda = 3.3$ packets per second and length of 2000 packets. Network jitters was modeled as Laplacian with zero mean and a standard deviation of 10 ms.

3.1. Robustness to Packet Losses

We first evaluated the robustness of our scheme against packet deletions by considering varying packet deletion probabilities $P_d = \{0.01, 0.02, 0.03, 0.1\}$. We embedded randomly generated watermarks into 4000 flows, from which true positive rates were calculated. In addition, we employed another 4000 unmarked flows to obtain the false positive rates. The watermark parameters were chosen as $N = 50$, $n = 10$ and $\Delta = 100$ ms. The detection threshold was chosen such that the false positive rate was kept below 1% for all deletion probabilities.

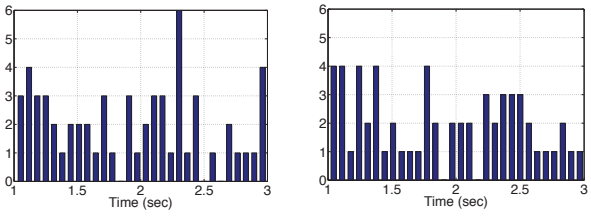
P_d	1%	2%	3%	10%
TP	0.9998	0.9998	0.9998	0.9942

Table 1. True Positive (TP) watermark detection rates for various deletion ratios P_d. All False Positive (FP) rates are restricted under 1%.

Table 1 presents the detection results. We see that the detector achieves rather high true positive rates, even when up to 10% of packets were deleted while maintaining false positive rates under 1%. Further tests show that the true positive rate would drop to 57% when packet deletion ratio is at

[4] $sgn(\cdot)$ denotes the sign function.

[5] A scenario when the first packet is lost can easily be dealt with by repeating the watermark in the network flow.



(a) Unwatermarked flows (b) Watermarked flows

Fig. 5. Histogram of empty intervals in an aggregate of 10 flows

20%, which is a rare occurrence in a network system. Hence, unlike other IPD-based designs which suffer from desynchronization, our scheme is robust against both network jitters and packet losses.

3.2. Watermark Visibility

To examine the visibility of our scheme, we performed two experiments: Kolmogorov-Smirnov test and multi-flow attack. The Kolmogorov-Smirnov (K-S) test evaluates the similarity between two sequences, by finding the maximum distance in their empirical distribution functions [8]. In our case, the K-S distance between two flows is measured as $\sup_x(|F_A(x) - F_B(x)|)$, where $F_A(x)$ and $F_B(x)$ are the empirical distribution functions of IPDs in two flows. We performed the K-S test on 1000 watermarked flows against 1000 unwatermarked flows. Results in Table 2 show that our watermark stays invisible within 99% confidence intervals corresponding to K-S distances below 0.036, a reference threshold suggested in [8].

It has been shown that a multi-flow attack can often detect and even remove the watermark added using interval-based techniques [5]. If a few flows all contain the watermark, then in their aggregated flow, an unusually high number of empty intervals could be observed. To test the visibility under the multi-flow attack, we aggregated 10 different flows with the same watermark embedded in the same position (a disadvantageous setup for our scheme). The embedding parameters used were $N = 50$, $n = 10$ and $\Delta = 100$ ms. The histogram of empty intervals in the aggregated flow is depicted in Figure 5. Compared with the non-watermark case, no clearly abnormal empty interval patterns are observed in the watermarked flow. The exact statistics of empty intervals for the two cases are given in Table 3. Again there is not a significant difference between watermarked and unwatermarked flows.

4. CONCLUSION

An invisible flow watermarking scheme is presented for network forensic application. Experimental results show that the embedded watermark can be retrieved with high probability in presence of both network jitter and high rate of packet drop. Moreover, we verified the transparency of the scheme against the K-S test and the multi-flow attack.

N \ Δ (ms)	100	80	60
30	0.0177	0.0138	0.0101
40	0.0233	0.0181	0.0133
50	0.0284	0.0223	0.0160

Table 2. Average K-S distances for varying watermark lengths and step-sizes.

	Marked	Unmarked
Mean	24.07	25.96
Standard Deviation	5.246	5.187

Table 3. Empty intervals over the first 500 packets for watermarked and unwatermarked flows.

5. ACKNOWLEDGEMENT

This work was supported in part by AFOSR under Grant FA9550-11-1-0016, MURI under AFOSR Grant FA9550-10-1-0573, and NSF CCF 10-54937 CAR.

6. REFERENCES

[1] Yin Zhang and Vern Paxson, "Detecting stepping stones," in *USENIX Security Symposium*, 2000, pp. 171–184.

[2] Xinyuan Wang and Douglas S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *ACM Conference on Computer and Communications Security*, 2003, pp. 20–29.

[3] Young June Pyun, Young Hee Park, Xinyuan Wang, Douglas S. Reeves, and Peng Ning, "Tracing traffic through intermediate hosts that repacketize flows," in *Infocom*, 2007, pp. 634–642.

[4] Amir Houmansadr, Negar Kiyavash, and Nikita Borisov, "Rainbow: A robust and invisible non-blind watermark for network flows," in *Network and Distributed System Security Symposium*, 2009.

[5] Negar Kiyavash, Amir Houmansadr, and Nikita Borisov, "Multi-flow attacks against network flow watermarking schemes," in *USENIX Security Symposium*, 2008, pp. 307–320.

[6] Brian Chen and Gregory W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, pp. 1423–1443, 2001.

[7] Matthew C. Davey and David J. C. Mackay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Transactions on Information Theory*, vol. 47, pp. 687–698, 2001.

[8] Jr Frank J. Massey, "The Kolmogorov-Smirnov Test for Goodness of Fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.