A Network Flow Correlation Method Based on Chaos Theory and Principal Component Analysis

Yang Chen and Yonghong Chen
(Corresponding author: Yang Chen)

College of Computer Science, Technology, Huaqiao University
No. 668 Jimei Avenue, Xiamen, Fujian, China 361021
(Email: 13476863090@163.com)

(Received Sept. 18, 2018; Revised and Accepted Mar. 23, 2019; First Online June 9, 2019)

Abstract

Detection of the related incoming and outgoing flows helps to expose the attackers hiding behind stepping stones. Currently, the network flow watermarking scheme is used for the detection of network flow correlation, due to the watermarking schemes introduce large delays to the target flows and often make it impossible to achieve robustness and invisibility. In this paper, we propose a novel flow correlation scheme based on Chaos Theory and Principal Component Analysis. In this method, the network traffic is preprocessed by phase space reconstruction of chaos. Then, traffic traits are extracted by Principal Component Analysis, which are used later to calculate similarity between sender and receiver based on cosine similarity. Experimental results show that the scheme can resist packet insertions, network jitter and losses.

Keywords: Chaos Theory; Flow Correlation; Network Security; Principal Component Analysis

1 Introduction

As the Internet is more and more used in various aspects of everyday life, people are realizing that computer systems are suffering more threats than ever before. Timely response and active defense has become an important guarantee to maintain the continuous dynamic network security. However, most network security mechanisms deal with these network attacks in a passive way. Intrusion detection system is an important part of network security. But current intrusion detection mechanisms are still difficult to effectively track and detect network attack sources [16]. In fact, because network attackers rarely launch attack through their own computers directly, they are more likely to hide their origin by connecting across multiple stepping stones [2, 13, 14, 19] or use anonymous communication systems (such as Tor [11]) before attacking the final targets, it makes intrusion tracing complex and difficult.

Currently, various network flow correlation methods have been proposed efficiently to link packet flows in a network in order to thwart various attacks such as stepping stones intrusion. Traditionally, passive network flow analysis methods [4, 8, 15, 20] have many shortcomings, such as poor real-time, high space costs, poor flexibility, low accuracy, and the inability to handle encrypted traffic, etc. Recently, the network flow watermarking provides a better way to track the intrusion source. However, the robustness and invisibility of network flow watermarks is very important, which are difficult to achieve at the same time [7]. This is because that the robustness requires the injected watermark always robust living in the network flow, while the invisibility prevents the active attackers to see the watermark in network flow. For instance, in the interval-based schemes, the duration of each intercepted flow is partitioned into short time intervals, and all packets within selected intervals are intentionally modulated to form a watermark pattern. Given that a few packets would not greatly affect the pattern created in the entire interval, these schemes are robust against network artifacts such as packet drops and inserts. However, one problem with such schemes is the lack of invisibility. Shifting packets in batches produce noticeable traces of the embedded watermarks, which can expose the watermark positions [5]. This enables the attackers to remove or modify the watermarks embedded in a network stream and even transfer them to another unrelated stream, which will make any linking techniques be meaningless.

In this paper, we proposed a new scheme for linking flows, which aims at designing a similarity degree to effectively link network flows without changing and forwarding the primitive traffic models [3]. Firstly, the embedding dimension and time delay of the network time series are calculated, then the chaotic phase space reconstruction is used to reconstruct the time series to obtain the space characteristics of the network flow. Second, we use the Principal Component Analysis (PCA) algorithm to extract the most important characteristics of the above obtained traffic. In brief, we compare our scheme with the

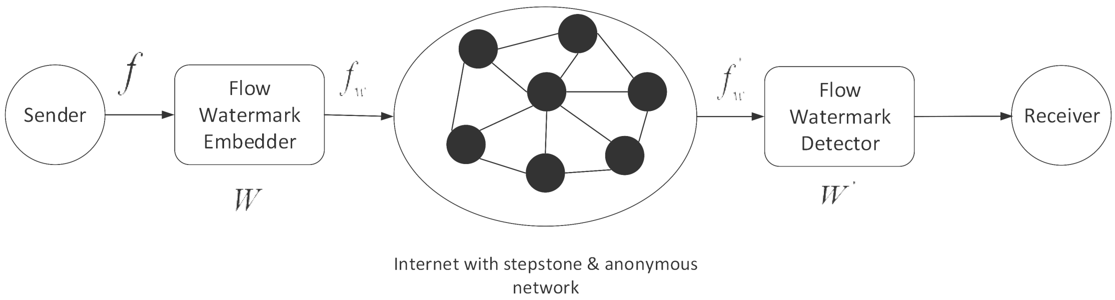

Figure 1: The universal model of network flow watermarking

classical existing methods through experiment and show that our approach can achieve better overall performance under network jitter, packet losses and insertions.

The rest of this paper is arranged as follows: Background on flow watermarking appears in Section II. In Section III we introduce our proposed intrusion detection scheme. Section IV presents the experimental results and discussion. This paper is concluded in Section V along with some future research directions.

2 Background

In this section, we review the problem of detecting stepping stones and then introduce the framework, universal model and typical characteristics of flow watermarks, respectively.

2.1 Stepping Stone Detection

Stepping stone attacks are a common way for network intruders to hide their identity. In a stepping-stone attack, the attacker compromises multiple hosts as relay machines, uses remote login such as Telnet or SSH to construct a chain of connections through these hosts, and then sends attacking commands to the victim through this chain [22]. Because each connection is made through a separate remote login, the next host in the chain can only see the identity of its immediate upstream neighbor, and the victim can only see the identity of the last host. Therefore, we must trace back the chain to find the origin of an attack.

2.2 Universal Model of Network Flow Watermarking

Flow watermarking technologies, embed watermark by changing or modulating traffic characteristics such as inter-packet delay (IPD) and interval centroid, at the sender side, nd the watermark will be identified and extracted at the receiver side to correlate the communication relationship of the sender and the receiver, as shown in Figure 1.

Figure 1 shows the universal model of network flow watermarking. The watermark embedder collects network traffic flow f, then selects a feature of the stream f (such as inter-packet delay (IPD), interval centroid, etc.) as the carrier of watermark w. Watermark detector captures the network traffic flow f_w and extract the watermark $w^{'}$, with watermarking detection algorithm, comparing w with $w^{'}$ to judge whether f_w is correlated with f.

In order to accurately trace the flow, flow watermarking technologies must have the following characteristics [24].

First of all, robustness is needed to ensure that watermark information survive to be correctly detected after malicious attacks or network transmission damages. Secondly, a successful watermark pattern should stay "invisible" to avoid possible attacks, if the intruder found that incoming flow is marked, he might command the stepping stone to take precautionary actions(for example, remove the watermarks).

At present, many flow watermarking approaches have been proposed. Houmansadr *et al.* [7] proposed an interval time-delay based watermarking scheme(RAINBOW), which first calculates the interpacket delays (lPDs) and saves them into the IPD database, and further increases or decreases the value of IPDs to embed the watermark information. In detection process, all IPDs are computed with the IPDs in the database to judge whether the watermark information existed. However, the demand and difficulty of network deployment have become much higher than before. In [21], ICBW method based on interval centroid was proposed, it embeds watermarking signals by adjusting the centroids of the intervals, However, its mechanism is built on a prerequisite that the interval centroid is stable when the count of packets is large enough.

Existing watermarking scheme has more or less modify the network traffic patterns, increasing the possibility of being discovered by the attacker.So this paper aims at designing a similarity degree to efficiently link network flows without forwarding and distorting the original traffic patterns.

3 Correlation Model

The correlation model proposed in this paper can be divided into three parts: Traffic preprocessing, feature extraction, and correlation detection. In the part A of this section, the network traffic is preprocessed with using chaos theory, which to get chaotic characteristics of network flow. The method restores the hidden characteristics of network flow by reconstructing the time series of network traffic, and can grasp the inherent nature and regularity of chaotic time series. However, the chaotic characteristics obtained in the part A use in practical applications directly will cause a large amount of calculations. So in the part B of this section, we use the Principal Component Analysis (PCA) algorithm to process the chaotic characteristics obtained in the Part A to get more robust traffic characteristics. In the part C of this section, we calculate the similarity between the sender and receiver based on the cosine similarity using the characteristics obtained in the part B.

The algorithm proposed in this paper is as in Table 1.

Table 1: Algorithm steps

Step 1	Collect network flow.
Step 2	Get a time series is $\{x_1, x_2, \cdots, x_n\}$
Step 3	Calculate the time delay and embedding dimension of the time series obtained in the second step using the methods in the part A.
Step 4	The phase space is reconstructed to obtain the space characteristics of the network flow based on the embedding dimension and time delay acquired in the third step.
Step 5	Process the space characteristics obtained in the fourth step to get the final required traffic characteristics using the Principal Component Analysis (PCA) algorithm in the part B.
Step 6	According to the traffic characteristics obtained in the fifth step, we use cosine similarity in the part C to detect the linked flows. If the similarity is within the range of $\eta = \left(\frac{\sqrt{2}}{2}, 1\right]$, the both sides are considered to be correlated and recognized successfully. Otherwise, it is likely that the received flow is uncorrelated.

In the following part of this section, we start with a brief review of phase space reconstruction and Principal Component Analysis (PCA) algorithm and then each component of our scheme is described in details.

3.1 Traffic Preprocessing

Chaos theory is widely used in chemistry, physics, mechanics, mathematics as well as economic system, and it has been proved to be an important and effective theoretical method to solve nonlinear problems. In this paper, chaos theory provides a good means and methods for analyzing network traffic [17]. The collected network traffic time series can be extended from the low dimensional space to high dimensional space through phase space reconstruction technique of the chaos theory. In high dimensional space, it can recover regular characteristic of the network traffic from the seemingly irregular network traffic. In a word, phase space reconstruction technique of the chaos can restore the hidden nature of the original system, analyzing and extracting fixed characteristic value under the original rules and nature of the system accurately.

Given a network flow, its time series $\{x(n), n = 1, 2, \cdots, N\}$, a phase space reconstruction $X_{m,\tau}$ is defined as

$$X_{m,\tau} = \left(x(n), x(n+\tau), \cdots, x(n+(m-1)\tau)\right), \quad n = 1, 2, \cdots, N_m \tag{1}$$

Where τ is time delay, the number of vectors in the point set $N_m = N - (m-1)\tau$, and it is the total number of reconstituted by the time sequence of the status point. m is embedding dimension. The embedding dimension refers to the number of variables needed to describe the motion of a system. The appropriate m and τ can deeply explain the space-time characteristics of traffic and reveal the movement rule of the dynamic system.

From the above detailed description of the embedding dimension m and time delay τ, we know that it is important to calculate the two parameters of the network traffic time series. As far as we know, the famous Takens theorem implies that an appropriate time delay τ and a good embedding dimension m play an important role in reconstruction state space, among which the trajectories may maintain the diffeomorphism with original dynamic system. In other words, the dynamic system can be analyzed through phase space reconstruction from certain a time series. As previous work [10] has proved the network traffic is chaotic, we can use the phase space reconstruct technique to get the optimal parameter. The calculation steps are as follows.

Firstly, we get a timestamp of network traffic to get time sequence $\{x_1, x_2, \cdots, x_n\}$ and reconstruct a phase space with the time delay τ and the embedding dimension m describing in above. Secondly, we determine the time delay by using the C-C method [18]. According to the formula (1), we can get the reconstruction of points in space $X_i = \left(x_i, x_{i+1}, \cdots, x_{i+(m-1)\tau}\right)$, the correlation integral of embedding time series is defined as

$$C(m, N, \gamma, t) = \tfrac{2}{M(M-1)} \textstyle\sum_{1 \le i \le j \le M} \theta(\gamma - d_{ij}), \gamma > 0 \tag{2}$$

$$\theta(z) = \begin{cases} 0, z < 0 \\ 1, z \ge 0 \end{cases} \tag{3}$$

Where $d_{ij} = \| X_i - Y_j \|$ is the distance between X_i and Y_j, $\| \bullet \|$ denotes maximal norm in this paper for convenience, $\theta(\bullet)$ is Heaviside function. The correlation integral measures the fraction of the pairs of points X_i, whose maximal norm separation is no greater than γ.

For the time series $\{x_i\}, i = 1, 2, \cdots, N$, it will be divided into t subsequence which do not overlap each other. We define each subsequence $S(m, N, \gamma, t)$ as

$$S\left(m, N, \gamma, t\right) = \frac{1}{t}\sum_{s=1}^{t}\left[C_s\left(m, \frac{N}{t}, \gamma, t\right) - C_s^m\left(1, \frac{N}{t}, \gamma, t\right)\right]$$

We choose a value corresponding to maximum (respectively minimum) radius γ, delta is define by

$$\begin{aligned} \Delta S\left(m, t\right) &= max\left\{S(m, \gamma_j, t)\right\} - min\left\{S\left(m, \gamma_j, t\right)\right\} \\ \Delta \bar{S}\left(t\right) &= \frac{1}{4}\sum_{m=2}^{s}\Delta S(m, t) \end{aligned}$$

The first minimum of $\Delta\overline{S}(t)$ is we need the optimal time delay τ.

In the end, we determine the best embedding dimension by using the Cao method [9]. The relative length of a point in a phase space is defined as

$$L\left(i, m\right) = \frac{\left\|X_{m+1}\left(i\right) - \ X_m^{NN}(i)\right\|}{\left\|X_m\left(i\right) - \ X_m^{NN}(i)\right\|}$$

Where $X_m(i)$ and $X_m^{NN}(i)$ are for the m dimension space of the ith vector and its nearest point. Then,

$$\begin{aligned} E\left(m\right) &= \frac{1}{N - m\tau}\sum_{i=1}^{N-m\tau} L\left(i, m\right) \\ E_1\left(m\right) &= {}^{E(m+1)}/_{E(m)} \end{aligned}$$

When $E_1(m)$ changes slowly or even is unchanged, the corresponding m will be the best embedding dimension. After obtaining two important parameters of delay time and embedding dimension, reconstructing phase space of original flow sequence, in m-dimensional space, the trajectory of n points of one-dimensional space can be expressed as

$$\begin{aligned} X &= [X(1), X(2), \cdots, X(N_m)]^T \\ &= \begin{pmatrix} x(1) & x(1+\tau) & \cdots & x(1+(m-1)\tau) \\ x(2) & x(2+\tau) & \cdots & x(2+(m-1)\tau) \\ \vdots & \vdots & \ddots & \vdots \\ x(N_m) & x(N_m+\tau) & \cdots & x(N_m+(m-1)\tau) \end{pmatrix} \end{aligned}$$

3.2 The Extraction of Traffic Traits

In order to reduce the amount of calculations in practical applications, we cannot directly use the chaotic characteristics obtained in the part A to calculate similarity. So in this part, we use Principal Component Analysis (PCA) [6,12,23] to analyze this higher dimensional matrix and obtain the major component of the reconstructed the original data. With the proposed approach, the traffic characteristics we obtain is more than an isolated subsequence, but contains all the information about the sequence. In this paper, the Principal Component Analysis (PCA) method is not only used as a tool to reduce the dimension of datasets, but also as a data processing method to extract the most important information from the original data set and use it as the extracted traffic characteristics.

For convenience that (10) will be recorded as

$$\begin{aligned} X &= \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{N_m 1} & \cdots & a_{N_m m} \end{pmatrix} \\ &= [X(1), X(2), \cdots, X(m)] \end{aligned}$$

In order to balance the weight of each element in each row, the data matrix is normalized and make its mean value of each row is 0. As shown by the following formula:

$$\begin{aligned} G &= \begin{pmatrix} \mathrm{g}_{11} & \cdots & \mathrm{g}_{1m} \\ \vdots & \ddots & \vdots \\ \mathrm{g}_{N_m 1} & \cdots & \mathrm{g}_{N_m m} \end{pmatrix} \\ \mathrm{g}_{ik} &= a_{ik} - \overline{a_i}; \forall i, k; 0 < i \leq N_m, 0 < k \leq m \\ \overline{a_i} &= \frac{1}{m}\sum_{j=1}^{m} a_{ij}\ ; 0 < i \leq N_m \end{aligned}$$

After that, every $\mathrm{g}_{ik}\,(0 < i \leq N_m + 1; 0 < j \leq m + 1)$ is used to calculate the covariance matrix $\boldsymbol{C}\,(m \times m)$ using the following formula:

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mm} \end{pmatrix}$$

$$\mathrm{Cov}\left(\mathrm{g}_i, \mathrm{g}_j\right) = c_{ij} = \frac{\sum_{k=1}^{L}(a_{ik} - \overline{a_i})(a_{jk} - \overline{a_j})}{m-1}$$

$$0 < i \leq m, 0 < j \leq m$$

And then we calculate the eigenvalues $\lambda = (\lambda_1,\ \lambda_2,\ \lambda_3,\ \cdots,\ \lambda_m)$ of the matrix $\boldsymbol{C}$ through the characteristic polynomial $\mid \boldsymbol{C} - \lambda\boldsymbol{I} \mid = 0$ ($\boldsymbol{I}$ denotes identity matrix, $\boldsymbol{C}$ represents covariance matrix mentioned above), and get the corresponding eigenvectors $\boldsymbol{V} = (V_1, V_2, V_3, \cdots, V_m)$.

We rearrange the eigenvalues in a descendant order and the eigenvectors correspondingly. Based on the aggregation of the eigenvalues, the sufficient corresponding eigenvectors are selected to calculate the principle components for the reconstruction of the original data matrix. For example, we choose s eigenvectors to reflect the information of the original data. So the front principle eigenvectors from v_1 to v_s comprise a new matrix $\boldsymbol{V}' = (v_1, v_2, v_3, \cdots, v_s)$. The principle matrix is constructed using the following formula:

$$\begin{aligned} P &= (P_1, P_2, P_3, \cdots, P_s) = G \times \boldsymbol{V}' \\ \boldsymbol{V}' &= (v_1, v_2, v_3, \cdots, v_s) \end{aligned}$$

According to the above formula, the energy of the original data $\boldsymbol{G}$ has been mapped to a new s-dimensional space. Meanwhile, some insignificant information has been ignored through the mapping process. Then we can calculate the principle components information in original data space.

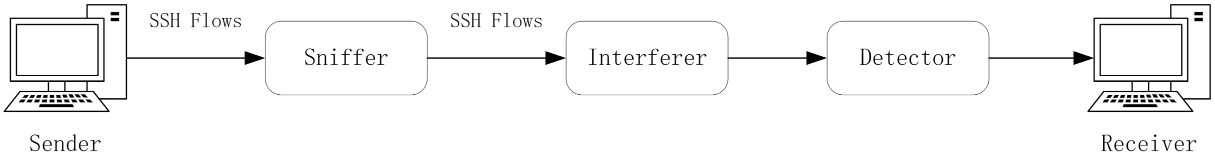

Figure 2: Experimental simulation environment

Finally, the mean value of each column of the new data matrix P is calculated, and the mean sequence is used as the traffic traits extracted from the algorithm:

$$F = \frac{1}{N_m}\sum\nolimits_{j=1}^{N_m} P_{ij}$$

3.3 Detecting the Correlation Traffic Flows

The stream of packets passes through a noisy channel that may include all kinds of interferences. Finally, the flow arrives at the detector. Assuming that the flow to the detector is disturbed relative to the original flow, it is necessary to processed that based on the above scheme firstly. The detector intercept the received network flow and obtains the digital summary $F^{'}$ of the data stream that may have encountered network noise. Next, the detector reads the digital digest $F^{'}$ stored in a third-party database. Calculate the cosine similarity of F and $F^{'}$:

$$\cos\theta = \frac{\boldsymbol{F}\cdot\boldsymbol{F}^{'}}{\left|\boldsymbol{F}\right|\left|\boldsymbol{F}^{'}\right|} = \frac{\sum_{i=1}^{s}(F_i \times F_i^{'})}{\sqrt{\sum_{i=1}^{s}(F_i)^2}\times\sqrt{\sum_{i=1}^{s}(F_i^{'})^2}}$$

Cosine similarity is a common method of determining the correlation between two n-dimensional vectors. It mapped individual indicator data to vector space and calculates the cosine of the angle between two vectors as a measure of the similarity between two variables. The closer the cosine of the angle to 1, the more similar [1]. In this paper, we pay more attention to the degree of similarity of variation trend between the internal components of the vector, so we use cosine similarity to measure the similarity between the sender and the receiver. If the similarity is within the range of $\eta = \left(\frac{\sqrt{2}}{2}, 1\right]$, the flows are considered to be correlated and recognized successfully. Otherwise, the traffic received is probably not related.

In fact, according to the scheme proposed in this paper, no additional communication overhead is required between the sender and the receiver, except for the shared digital summaries.

4 Simulation Results

The experiment simulation environment design is shown in Figure 2. A network flow passing through the Sniffer gets monitored in real-time, and then generate a digital digest according to traffic traits, and the generated digital digest is passed to the detector secretly through a secure channel, the Interferer implement the potential network jitter and countermeasures intentionally introduced by the adversaries (such as packet insertions and packet losses), once receipt of a disturbed version of the primitive traffic, the detector tries to confirm whether the two flows are linked based on the comparison of digital digest of source and sink ends. In addition, the SSH flows used in the experiment came from CAIDA anonymous network traces, and each SSH stream has a length of 2000. These real SSH streams reflect some of the typical behavior of people in the network.

4.1 Detection Rate of Packet Insertion and Deletion

The detection rate of the presented scheme in this paper is also evaluated in a networked environment where not only packet deletions but also packet insertions occur. Tests have shown that the insertion of chaff packets are conformed to a Pareto distribution [7], and the removal of the packets is independently and randomly. The experimental results shown in Figure 3.

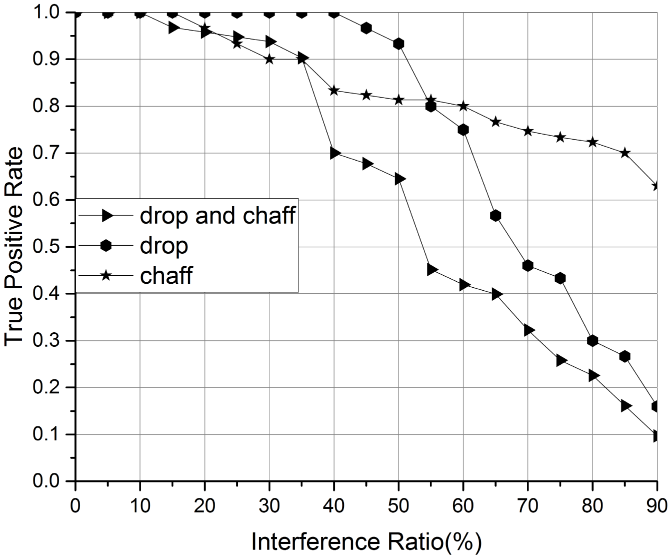

Figure 3: True positive rates comparison under Packet Insertion and deletion

Encouragingly, even if packets injection rate and loss rate are up to 10%, the correlated traffic can still be accurately detected. In addition, from the experimental results, it can be observed that the detection correctness rate is fewer affected by different number of chaff packets than the packet loss. There is the strong possibility that the digital digest designed in this scheme is based on the inherent characteristics of the flows. Packet deletions will cause the inherent characteristics of the stream to be severely damaged, so that the reconstructed phase space can not fully reflect the original flow characteristics, which leads to the decrease of detection efficiency. However, packet insertions may only influence the inherent characteristics of a portion of the source stream.

4.2 Accuracy Under Various Interference

As far as we know, existing network flow watermark designs can be roughly divided into two categories: Interval-based and IPD-based [3]. Non-blind watermark (RAINBOW) is a good example based on IPD schemes. RAINBOW embeds watermarking by fine-tuning the packet delay in network traffic, after recording the delay sequence information between packets [7]. In interval-based schemes, for example, an interval center-based watermarking (ICBW) presented by Wang *et al.* [21] randomly selects time interval as two different subsets and performs an operation on the centroid of the selected entire time interval pairs to embed a watermark.

In this section, packet deletions and chaff packets are introduced in the case of network jitter. Following the observation of previous work that shows jitter (difference of two delays) is approximately i.i.d. zero-mean Laplace distributed [7], we vary the standard deviation of jitter σ over $\{10, 20, 30, 40\}ms$. Evaluate the performance of three scheme in the same network environment: Our solution, ICBW and RAINBOW. For each solution, 6000 different network flows are used to test their performance under various interference. In addition there are 6,000 network flows acting as control groups that are passed directly to the detector without any interference. Figure 4, Figure 5 and Figure 6 depict the average true positive rates and false positive rates for these 6000 network flows respectively.

As shown in Figure 4 and Table 2, where P_i and P_d denote packet loss interference rate and packet insertions interference rate respectively and σ represents the number of network jitters added. With the increase of the number of available packets, the detection rate will be greatly improved. Compared with RAINBOW and ICBW, actually, the fewer number of packets required by our scheme in achieving the same level of accuracy. Moreover, as the number of available packets increases, the detection rate will be greatly increased. The figure has shown that if the number of packets is the same as the number of packets in the original stream, the detection rate of our solution is over 90% even if as many as 30% packets simultaneously deleted and inserted besides jitter as high as $40ms$, yet ICBW and RAINBOW basically keep smaller than 80% detection rate under these conditions.It may be that, network traffic has self-similarity and chaotic, and reconstructed phase space based on more data packets, which reflects the inherent law of data flow implicitly, so the detection rate will be higher. As shown in Figure 5 and Table 3, the overall detection rate of our program is still higher than that of ICBW and RAINBOW.

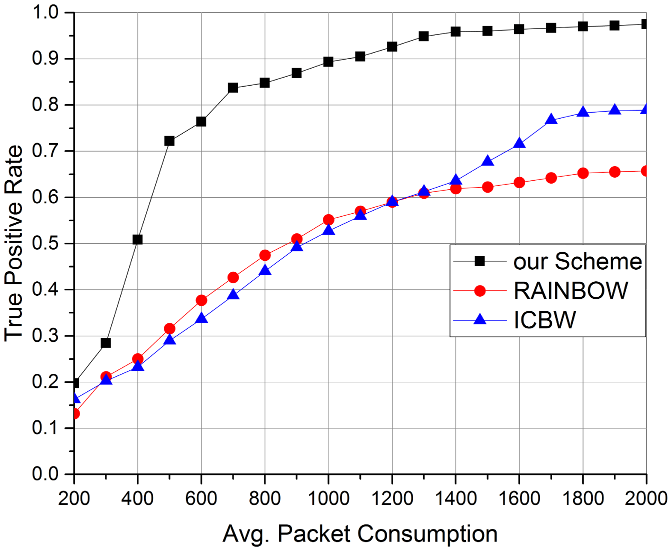

Figure 4: True positive rates comparison under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

Table 2: True detection rate under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

Average Packet Consumption	400	800	1200	1600	2000
Our Scheme	0.512	0.849	0.922	0.972	0.974
RAINBOW	0.251	0.486	0.598	0.632	0.665
ICBW	0.242	0.442	0.606	0.724	0.778

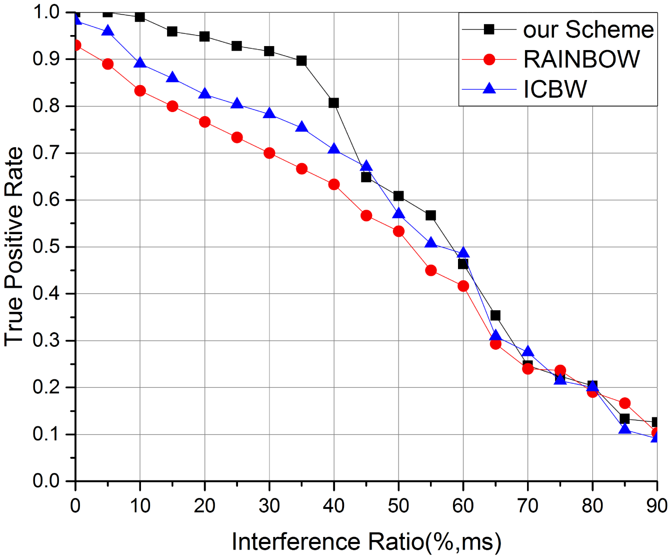

Figure 5: True positive rates comparison under Packet Insertion, Removal and Network Jitter

Table 3: True positive rates under Packet Insertion, Removal and Network Jitter

Interference Rate	**10%**	**20%**	**30%**	**40%**
Our Scheme	0.992	0.958	0.917	0.816
RAINBOW	0.839	0.787	0.708	0.648
ICBW	0.898	0.823	0.789	0.710

Figure 6 and Table 4 show that since the digital digest designed in this paper is generated according to traffic pattern and there seems to be some inherent similarity between two uncorrelated network flows, when the original traffic is severely disrupted and nine-tenths of packets are not available, the false detection rate for our solution may not be ideal, but it always does not exceed 8%. In addition, as the number of required packets increases, the false positive rate of our scheme has a very significant decrease. The figure also shows that when the number of received packets exceeds 50% of the original flow length, the error rate of our designs is lower than ICBW. When the available packet reaches 80% of the original packet flow length, the false positive rate of our scheme is smaller than RAINBOW, which is basically not exceeding 1%.

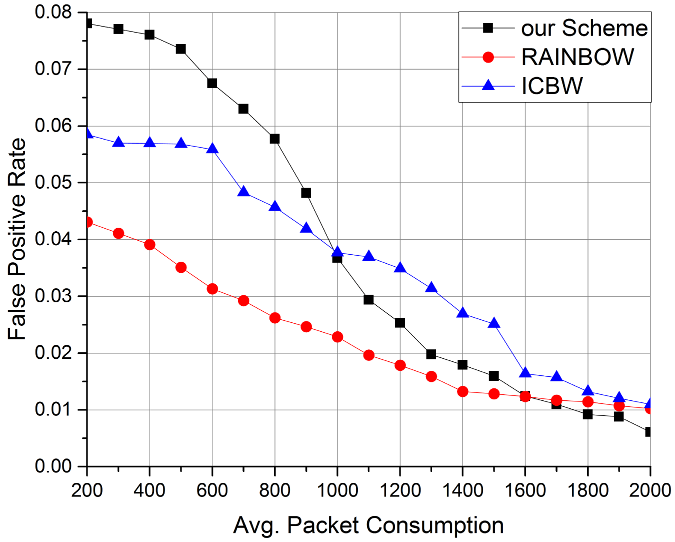

Figure 6: False positive rates comparison under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

Table 4: False positive rates comparison under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

false positive rate	**400**	**800**	**1200**	**1600**	**2000**
Our Scheme	0.072	0.058	0.026	0.013	0.007
RAINBOW	0.039	0.027	0.019	0.013	0.011
ICBW	0.057	0.041	0.036	0.016	0.011

5 Conclusion

In this paper, we proposed a novel flow correlation scheme based on Chaos Theory and Principal Component Analysis that does not rely on network watermarking. Only main part of the flow characteristics are utilized without interfering with the communication patterns of the intercepted flows, which prevents attackers from detecting the trace process. The ideal network flow watermarking technology needs to satisfy robustness and invisibility simultaneously, but it can only meet one of them in practical applications, and our solution does not have this concern. And there is no additional communication overhead between the sender and the receiver, except for the shared digital summaries. Finally, theoretical analysis and experimental results confirmed the correctness and the operability of network flow correlation model based on chaos theory and PCA despite the presence of network jitter, packet additions and removals.

There are still some limitations of our proposed method, and its false positive rate is higher than network watermarking, and this is our following work. We may be able to understand more deeply the principles of common coding techniques such as network coding, channel coding, source coding, video coding, etc., and explore the intrinsic connection of these technologies, and seek more robust and adaptable information coding techniques to optimize the feature coding module of this scheme.

Acknowledgments

This study is supported by the National Natural Science Foundation of China(NO. 61370007). The Postgraduate Scientific Research Innovation Ability Training Plan Funding Projects of Huaqiao University(1611314002).

References

[1] S. Amit, *et al.*, "Modern information retrieval: A brief overview," *IEEE Data Engineering Bulletin*, vol. 24, no. 4, pp. 35–43, 2001.

[2] M. Behi, M. Ghasemi, and H. V. Nejad, "A new approach to quantify network security by ranking of security metrics and considering their relationships," *International Journal of Network Security*, vol. 20, no. 1, pp. 141–148, 2018.

[3] Y. Chen, N. Zhang, H. Tian, T. Wang, and Y. Cai, "A novel connection correlation scheme based on threshold secret sharing," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2414–2417, 2016.

[4] G. Danezis, "The traffic analysis of continuous-time mixes," in *International Workshop on Privacy Enhancing Technologies*, pp. 35–50, May 2004.

[5] X. Gong, M. Rodrigues, and N. Kiyavash, "Invisible flow watermarks for channels with dependent substitution, deletion, and bursty insertion errors," *IEEE*

Transactions on Information Forensics and Security, vol. 8, no. 11, pp. 1850–1859, 2013.

[6] H. Harold, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417, 1933.

[7] A. Houmansadr, N. Kiyavash, and N. Borisov, "Non-blind watermarking of network flows," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1232–1244, 2014.

[8] C. T. Li, M. S. Hwang, and S. Chen, "A batch verifying and detecting the illegal signatures," *International Journal of Innovative Computing, Information and Control*, vol. 6, no, 12, pp. 5311–5320, 2010.

[9] Z. Li, H. Zheng, and C. Pei, "A modified cao method with delay embedded," in *The 2nd International Conference on Signal Processing Systems (ICSPS'10)*, pp. V3–458–V3–460, July 2010.

[10] X. Ma and Y. Chen, "Ddos detection method based on chaos analysis of network traffic entropy," *IEEE Communications Letters*, vol. 18, no. 1, pp. 114–117, 2014.

[11] A. Montieri, D. Ciuonzo, G. Aceto, and A. Pescapé, "Anonymity services tor, I2P, JonDonym: Classifying in the dark," in *The 29th International Teletraffic Congress*, pp. 81–89, Sep. 2017.

[12] C. Mu, X. Huang, J. Wu, and Y. Ma, "Network traffic signature generation mechanism using principal component analysis," *China Communications*, vol. 10, no. 11, pp. 95–106, 2013.

[13] F. Nabi and M. M. Nabi, "A process of security assurance properties unification for application logic," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 40–48, 2017.

[14] E. U. Opara and O. A. Soluade, "Straddling the next cyber frontier: The empirical analysis on network security, exploits, and vulnerabilities," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 10–18, 2015.

[15] J. Qin, R. Sun, X. Xiang, H. Li, and H. Huang, "Anti-fake digital watermarking algorithm based on QR codes and DWT," *International Journal Network Security*, vol. 18, no. 6, pp. 1102–1108, 2016.

[16] T. Shi, W. Shi, C. Wang, and Z. Wang, "Compressed sensing based intrusion detection system for hybrid wireless mesh networks," in *International Conference on Computing, Networking and Communications (ICNC'18)*, pp. 11–15, Mar. 2018.

[17] J. Song, D. Meng, and Y. Wang, "Analysis of chaotic behavior based on phase space reconstruction methods," in *The Sixth International Symposium on Computational Intelligence and Design (ISCID'13)*, pp. 414–417, Oct. 2013.

[18] L. Tang and J. Liang, "CC method to phase space reconstruction based on multivariate time series," in *The 2nd International Conference on Intelligent Control and Information Processing (ICICIP'11)*, pp. 438–441, July 2011.

[19] J. Wang, Y. Yu, and K. Zhou, "A regular expression matching approach to distributed wireless network security system," *International Journal Network Security*, vol. 16, no. 5, pp. 382–388, 2014.

[20] X. Wang, D. S. Reeves, and S. F. Wu, "Inter-packet delay based correlation for tracing encrypted connections through stepping stones," in *European Symposium on Research in Computer Security*, pp. 244–263, Oct. 2002.

[21] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *IEEE Symposium on Security and Privacy (SP'07)*, pp. 116–130, May 2007.

[22] X. Xu, J. Zhang, and Q. Li, "Equalized interval centroid based watermarking scheme for stepping stone traceback," in *IEEE International Conference on Data Science in Cyberspace (DSC'16)*, pp. 109–117, June 2016.

[23] K. Yoshiki, F. Kensuke, and S. Toshiharu, "Evaluation of anomaly detection based on sketch and PCA," in *IEEE of Global Telecommunications Conference (GLOBECOM'10)*, pp. 1–5, Dec. 2010.

[24] L. Zhang, Y. Kong, Y. Guo, J. Yan, and Z. Wang, "Survey on network flow watermarking: Model, interferences, applications, technologies and security," *IET Communications*, vol. 12, no. 14, pp. 1639–1648, 2018.

Biography

Yang Chen was born in Chongqing, China in 1994. She received the B.S. Degree from Wuhan Textile University, Hubei, China in 2016. She is currently pursuing the M.S. Degree in Huaqiao University. Her research interests include Digital Watermarking and Property Protection and Blockchain and Application.

Yonghong Chen received the Ph.D. degree from Chongqing University, Chongqing, China, in 2005. He is a Professor in Huaqiao University of China. His current interests include Network and Information Security, Network intrusion detection, Digital Watermarking and Property Protection and Blockchain and Application.