ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Protecting Streaming Data Using Provenance Spread Spectrum Watermarking

Ramya K P[1], Revathi M K[2]

Information Technology, Anna University, Dr.Sivanthi Aditanar College of Engineering, Tuticorin-628215, Tamilnadu, India[1, 2]

ABSTRACT: Large number of application areas, like location-based services, transaction logs, sensor networks is qualified by uninterrupted data stream from many. Chasing of data provenance in extremely active circumstance is a crucial requirement, because data provenance is a key component in appraising data trustiness which is important for lots of application. Provenance handling of continuous data needs to cover various issues, admitting the storage efficiency, processing throughput, bandwidth conception and secure transmission. This paper addresses the challenges by providing secure and efficient transmission of provenance along with sensor data by embedding it over the inter packet delays (IPDs). The embedding of provenance within a host medium makes this technique reminiscent of watermarking. Spread-spectrum based watermarking technique is proposed, that avoids data degradation due to traditional watermarking. Provenance is extracted effectively based on an optimal threshold mechanism that minimizes the probability of provenance decoding error. The outcome of the observation depicts that this system is scalable and highly resilient in provenance recovery versus several attacks up to specific level.

KEYWORDS: Streaming Data, Water Marking, Provenance Security, Sensor Network, Malicious Attack, Spread Spectrum Watermarking.

I. INTRODUCTION

Many applications process high volumes of streaming data. Examples include Internet traffic analysis, sensor networks, Web server and error log mining, financial tickets and on-line trading, real-time mining of telephone call records or credit card transactions, tracking the GPS coordinates of moving objects, and analysing the result of scientific experiments. In general, a data stream is a data set that is produced incrementally over time, rather than being available in full before its processing begins. Of course, completely static data are not practical, and even traditional databases may be updated overtime. A large network contains thousands of routers and links, and its core links may carry many thousands of packets per second; in fact, optical links i the Internet backbone a reach speeds of over 100 million packets per second. The traffic flowing through the network is itself a high-speed data stream, with each data packet containing fields such as a timestamp, the source and destination IP addresses, and ports. Other network monitoring data streams include real-time system and alert logs produced by routers, routing and configuration updates, and periodic performance measurements. Examples of performance measurements are the average router CPU utilization over the large five minutes and the number of inbound and outbound packets of various types over the last five minutes. Understanding these data stream is crucial for managing and troubleshooting a large network. However, it is not feasible to perform complex operations on high-speed streams or to keep transmitting Terabytes of raw data to a data management system. Instead, to need scalable and flexible end-to-end data stream management solutions, ranging from real-time low latency alerting and monitoring, ad-hoc analysis and early data reduction on raw streaming data, to long-term analysis of processed data.

A digital watermark is a digital signal or pattern inserted into a digital image. Since this signal or pattern is present in each unaltered copy of the original image, the digital watermark may also serve as a digital signature for the copies. A given watermark may be unique to each copy (e.g. to identify the intended recipient), or be common to multiple copies (e.g. to identify the document source). In either case, the watermarking of the document involves the transformation of the original into another form. This distinguishes digital watermarking from digital fingerprinting, where the original file remains intact and a new created file 'describes' the original file's content.

Digital watermarking is also to be contrasted with public-key encryption, which also transform original files into another form. It is a common practice nowadays to encrypt digital documents so that they become un-viewable without

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

the decryption key. Unlike encryption, however, digital watermarking leaves the original image (or file) basically intact and recognizable. In addition, digital watermarks, as signatures, may not be validated without special software. Further, decrypted documents are free of any residual effects of encryption, whereas digital watermarks are designed to be persistent in viewing, printing, or subsequent re-transmission or dissemination.

II. SPREAD SPECTRUM WATERMARKING

Spread spectrum is a transmission technique by which a narrowband data signal is spread over a much larger bandwidth so that the signal energy present in any single frequency is undetectable. In our context, the sequence of inter packet delays is the communication channel and the provenance is the signal transmitted through it. Provenance is spread over many IPDs such that the information present in one IPD (i.e., container of information) is small. Consequently, an attacker needs to add high amplitude noise to all of the containers in order to destroy the provenance. Thus, the use of the spread spectrum technique for watermarking provides strong security against different attacks. To have adopted the direct sequence spread spectrum (DSSS) technique which is widely used for enabling multiple users to transmit simultaneously on the same frequency range by utilizing distinct pseudo noise sequences [9]? The intended receiver can extract the desired user's signal by regarding the other signals as noise-like interferences. The components of a DSSS system are as follows:

Input:

- The original data signal $d(t)$, as a series of $+1, -1$.
- A PN sequence$px(t)$, encoded like the data signal. Nc is the number of bits per symbol and is called PN length.

Spreading. The transmitter multiplies the data with the PN code to produce spreaded signal as $s(t) = d(t)px(t)$.
Despreading. The received signal $r(t)$ is a combination of the transmitted signal and noise in the communication channel. Thus $r(t) = s(t) + n(t)$, where $n(t)$ is a white Gaussian noise. To retrieve the original signal, the correlation between $r(t)$ and the PN sequence $pr(t)$ at the receiver is computed as $R(\tau)\frac{1}{N_c}\sum_{t=T}^{T+Nc} r(t)pr(t+\tau)$. $px(t) = pr(t) and\ \tau = 0$, i.e., $px(t)$ is synchronized with $pr(t)$, then the original signal can be retrieved. Otherwise, the data signal cannot be recovered. So, a receiver without having the PN sequence of the transmitter cannot reproduce the originally transmitted data. This fact is the basis for allowing multiple transmitters to share a channel. In this paper, to refer to $R(O)$ as *cross correlation*.
To retrieve the signal for jth user, the cross-correlation between $r(t)$ and $pxj(t)$ is computed. Multi-user communications introduces noise to the signal of interest and interfere with the desired signal in proportion to the number of users. The condition for error free communication in DSSS can be derived from Shannon's channel-capacity theorem

$$C = B\log_2\left(1+\frac{S}{N}\right),$$

where C is the amount of information allowed by the communication channel, B is the channel bandwidth, and S/N is the signal-to-noise ratio. As $S/N\ is\ usually\ \ll 1$ for spread-spectrum applications, the expression becomes

$$\frac{C}{B}\approx\frac{S}{N}$$

III. PROVENANCE WATERMARKING

There are two main steps in our algorithm, which are described as follows. Provenance Encoding: This step works in three phases: Generation of Delay Perturbations, Selection of a Delay Perturbation and Provenance Embedding. Provenance Decoding: This step works in two phases: Reordering IPDs, Threshold-Based Decoding.

3.1 Provenance Encoding

Fig.1. represents an overview of our approach for provenance encoding at a sensor node in the data path and decoding at the BS. The process a node n_i follows to encode a bit of PN sequence over an IPD is summarized below

3.1.1 Generation of Delay Perturbation

As the first step to embed provenance, a node ni generates a delay sequence that is used for watermarking. The PN sequence pni and impact factor α_i are used for this purpose. The PN sequence, consisting of a sequence of $+1$ and -

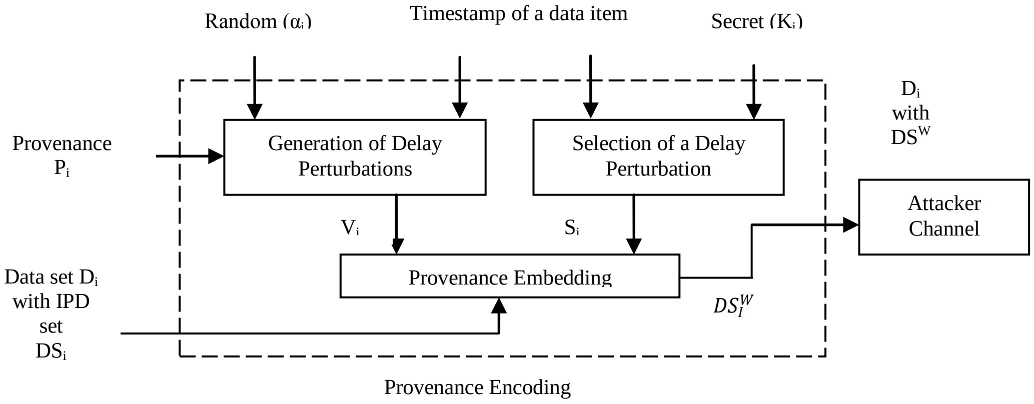


Fig.1. Provenance encoding at a sensor node in the data path

1's, is characterized by a zero mean. The zero mean property is required to ensure successful information decoding at the BS in the context of DSSS-supported multiuser communication. α_i is a random (real) number generated according to a normal distribution $N(\mu, \sigma)$. μ and σ are predetermined and known to the BS and all the nodes. Thus, the BS only knows the distribution of i's, but not their exact values. However, ni generates the set of delay perturbations Vi as a sequence of real numbers as follows:

$$\begin{aligned} Vi &= \alpha_i \times pni \\ &= \alpha_i \times \{pni\,[1], pni\,[2], \dots, pni\,[L_P]\} \\ &= \{(\alpha_i \times pni\,[1]), \dots, (\alpha_i \times pni\,[L_P])\} \\ &= \{v_i[1], v_i[2], \dots, v_i[L_P]\}. \end{aligned}$$

3.1.2 Selection of Delay Perturbation

To present the algorithm that a node applies to select the delay perturbation (from Vi) corresponding to an IPD. If to sequentially assign the delays to the IPDs (which implies that the provenance bits are embedded sequentially), it will be much easier for the attackers to infer information about the provenance or to corrupt the provenance. Hence, to randomize the embedding positions using a different permutation of the elements in Vi. On the arrival of any $(j+1)$th data packet, the jth IPD $\Delta[j]$ is considered for watermarking and the information to watermark is picked out from Vi using a selection algorithm. Thus, instead of watermarking $vi[j]$ over the IPD $\Delta(j)$, to select a delay $vi[kj]$ for this purpose, where kj is an index within $[0, Lp-1]$.the algorithm uses the secret Ki and the packet timestamp, and selects a delay perturbation for the IPD according to the following formula:

$$selection(\Delta[j]) = H(ts[j+\ 1]||Ki\,)mod\ Lp.$$

Here, H is a lightweight, secure hash function, k is the concatenation operator, and $ts[j+1]$ represents the packet timestamp. Since secure hash functions generate uniformly distributed message digests, each execution of the selection mechanism will result in a unique integer in the range $[0, Lp-1]$. The resulting integer can be used to index a distinct element in Vi. The indices are used to point the elements in Vi. Thus; the order according to which each node embeds the delays from Vi over the IPDs forms a permutation of the elements different from the sequential order. This sequence is denoted as $Si = \{si[1], si[2], \dots\ si[L_p]\} = \{vi[k1], vi[k2], \dots, vi[kLp]$.

3.1.3 Provenance Embedding

The simple provenance is represented as a simple path. Each node in the path watermarks its PN sequence over a set of Lp IPDs, i.e., $(Lp\ +\ 1)$ packets is utilized. Intuitively, the first packet in a data flow does not experience any delay due to provenance embedding. For any other $(j\ +\ 1)$th data packet (sent/forwarded), each node in the path hides a provenance bit over the associated IPD $\Delta[j]$. interchangeably, a node ni uses the IPD $\Delta[j]$ to accommodate a delay perturbation$(vi[k_j] =\ si[j])$. Using $si[j]$, the delay to be added to $\Delta[j]$ is computed as $\lambda[j] =\ si[j]\ \times T$;where T is the value of a time unit. If $si[j]\ >\ 0$, the resulting $\lambda[j]\ >\ 0$ and then to can perform watermarking by simply adding $\lambda[j]\ to\ \Delta[j]$. But if $si[j]\ <\ 0$, the delay to be added to an IPD is negative. To avoid this situation, we introduce a constant offset when calculating $\lambda i[j]$, which ensures that $\lambda i[j]$ is always positive. The offset may be any constant

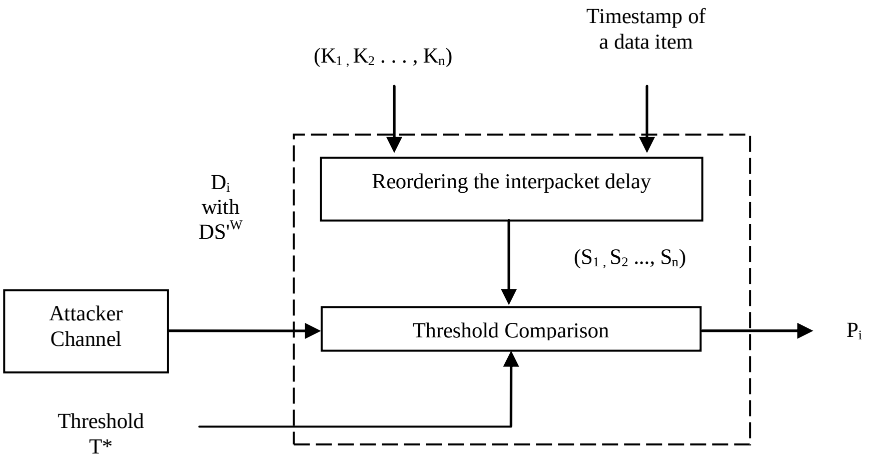


Fig.2. Stages of Provenance Decoding at the receiver

leading to $\lambda i[j] > 0$. To use $\mu + const * \sigma$ in our scheme, where $const$ is any constant that makes $\lambda i[j]$ greater than 0, i.e.,

$$const > \frac{-(si[j] + \mu)}{\sigma}$$

3.1 Provenance Decoding

An overview of our approach for provenance decoding at the receiver is shown in Fig.2. The process a node n_i follows to decode a bit of PN sequence over an IPD is summarized below:

3.2.1 Reordering the IPDs

The watermarked version DS_w is collected by received node with the interpacket delays {Δ[1], Δ[2],......, Δ[L_p]}. On the arrival of any (j+1)th data packet, ni records the IPD Δ[j] and assigns a delay perturbation $v_i[k_j] \in V$ to it. To ensure the robustness of the scheme, the delay perturbations are not assigned sequentially to the IPDs, i.e., $v_i[j]$ is not assigned to Δ[j]. Instead, a delay perturbation $v_i[k_j]$ is selected using the secret K_i and the packet timestamp. On the arrival of any $(j + 1)$th data packet, the j^{th} IPD $\Delta[j]$ is considered for watermarking and the information to watermark is picked out from Vi using a selection algorithm. Thus, instead of watermarking $vi[j]$ over the IPD $\Delta(j)$, to select a delay $vi[kj]$ for this purpose, where kj is an index within $[0, Lp - 1]$.the algorithm uses the secret Ki and the packet timestamp, and selects a delay perturbation for the IPD according to the following formula:

$$selection(\Delta[j]) = H(ts[j + 1]||Ki\,)mod\ Lp.$$

Here, H is a lightweight, secure hash function, k is the concatenation operator, and $ts[j + 1]$ represents the packet timestamp. Since secure hash functions generate uniformly distributed message digests, each execution of the selection mechanism will result in a unique integer in the range$[0, Lp - 1]$. The resulting integer can be used to index a distinct element in Vi. The indices are used to point the elements in Vi. Thus, the order according to which each node embeds the delays from Vi over the IPDs forms a permutation of the elements different from the sequential order. This sequence is denoted as $Si = \{si[1], si[2], \ldots\ si[L_p]\} = \{vi[k1], vi[k2], \ldots, vi[kLp]$.

3.2.2 Threshold Based Decoding

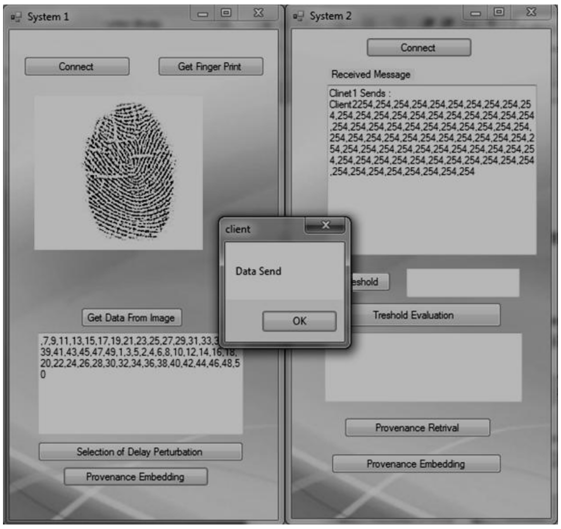
For any node n_i, the BS computes the threshold from the IPDs which is calculated by received node. If the provenance result exceeds the threshold T*, the BS decides that pn_i was embedded as a part of the provenance. The threshold is calculated as follows

$$T *= \sum_{j=0}^{lp} \frac{\Delta[j]}{ts(j + 1)}$$

After calculating the T*, that is used for provenance retrieval purpose. As already told that the fingerprint image is considered as sensor data, the matrix value that is calculated from the image is reordered. The reordered data is then converted into fingerprint image as a result.

IV. EXPERIMENTAL RESULTS

All experiments are performed on a Desktop PC with Intel Duo Core 1.7 GHz CPU, 2G Ram and Windows XP operating system. Programs and codes are implemented in VB.Net. The sensor data was gathered from the sensor device and it was taken for further process. Here the finger print device is considered as sensor device and the captured finger print image is considered as sensor data. After capturing the finger print image, it was converted into matrix format and stored in database. The nodes that participated in data transmission were connected in network. The delays are generated and it was assigned to sensor data in random. The sensor data was send from one node to another according to assigned delays. Provenance Embedding at the receiver is shown in Fig.3. In the receiver side the data was received and it is stored along with the received time. Then it is decoded to get the original sensor image. Fig.4. shows Provenance Decoding at the receiver.

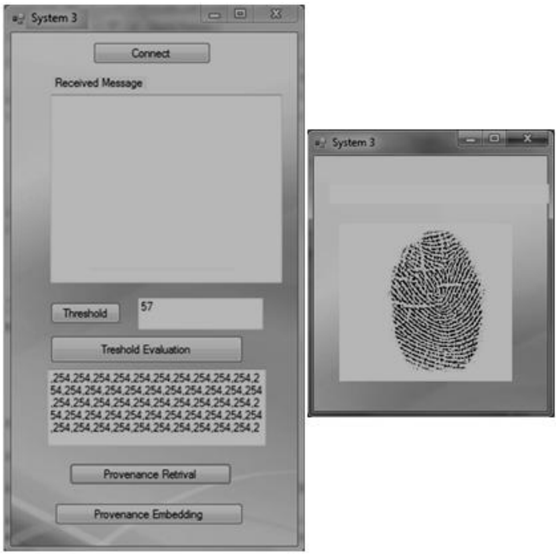


```
,7,9,11,13,15,17,19,21,23,25,27,29,31,33,3
39,41,43,45,47,49,1,3,5,2,4,6,8,10,12,14,16,18,
20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,5
0
```

```
Clinet1 Sends :
Client2254,254,254,254,254,254,254,254,254,25
4,254,254,254,254,254,254,254,254,254,254,254
,254,254,254,254,254,254,254,254,254,254,254,
254,254,254,254,254,254,254,254,254,254,254,2
54,254,254,254,254,254,254,254,254,254,254,25
4,254,254,254,254,254,254,254,254,254,254,254
,254,254,254,254,254,254,254,254
```

Data Send

Fig.3. Provenance Embedding



57

```
,254,254,254,254,254,254,254,254,254,254,2
54,254,254,254,254,254,254,254,254,254,254,254
,254,254,254,254,254,254,254,254,254,254,2
54,254,254,254,254,254,254,254,254,254,254,254
,254,254,254,254,254,254,254,254,254,254,2
```

Fig.4. Provenance Decoding

V. CONCLUSION

Interpacket timing based network flow watermarking has been widely used to identify the correlated traffic flows and to detect the source of attack behind the stepping stone(s). Our approach address the novel problem of securely transmitting provenance for data streams. We propose a spread-spectrum watermarking-based solution that embeds provenance over the interpacket delays. Spread spectrum technique is used so that it makes watermark delays much smaller. The decoding process does not requires the IPDs to be stored in database. The security features of the scheme make it able to survive against various sensor network or flow watermarking attacks. With the capability of capturing data packets and interpacket timing characteristics, an outside attacker may try to disrupt provenance security in different ways. In Provenance Detection and Retrieval attack, an attacker might want to identify and extract the provenance embedded by a node. Several attacks have been devised to detect and corrupt the active timing-based watermark in network flows. In our scheme, the watermarked IPDs do not follow any regular pattern. Thus our watermarking scheme show the robustness and makes the embedded provenance invisible to most of the attacks.

REFERENCES

[1] Chong S, Skalka C, and Vaughan J A, "Self-Identifying Sensor Data," Proc. Information Processing in Sensor Networks (IPSN), pp. 82-93, 2010.

[2] Cox I and Miller M, "Electronic Watermarking: The First 50 Years," Proc. IEEE Workshop Multimedia Signal Processing pp. 225- 230, 2001.

[3] Dixon R C, Spread Spectrum Systems: With Commercial Applications, third ed. John Wiley and Sons, Inc., 1994.

[4] Hasan R, Sion R, and Winslett M, "The Case of the Fake Picasso:Preventing History Forgery with Secure Provenance," Proc. Conf.File and Storage Technologies (FAST), pp. 1-14, 2009.

[5] Houmansadr A, Kiyavash N, and Borisov N, "Multi-Flow Attack Resistant Watermarks for Network Flows," Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing, pp. 1497-1500, 2009.

[6] Kiyavash N, Houmansadr A, and Borisov N, "Multi-Flow Attacks against Network Flow Watermarking Schemes," Proc. USENIX Conf. Security Symp., pp. 307-320, 2008. Cabuk S, "IP Covert Timing Channels: Design and Detection," Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 178-187, 2004.

[7] Lim H, Moon Y, and Bertino E, "Provenance-Based Trustworthiness Assessment in Sensor Networks," Proc. Workshop Data Management for Sensor Networks, pp. 2-7, 2010.

[8] National Cyber Security Research and Development Challenges, Related to Economics, Physical Infrastructure and Human Behavior, 2009.

[9] Peng P, Ning P, and Reeves D S, "On the Secrecy of Timing- Based Active Watermarking Trace-Back Techniques," Proc. IEEE Symp. Security and Privacy (SP), pp. 334-349, 2006.

[10] Simmhan Y L, Plale B, and Gannon D, "A Survey of Data Provenance in E-Science," SIGMOD Record, vol. 34, pp. 31-36,2005.

[11] Syalim A, Nishide T, and Sakurai K, "Preserving Integrity and Confidentiality of a Directed Acyclic Graph Model of Provenance," Proc. Working Conf. Data and Applications Security and Privacy, pp. 311-318, 2010.

[12] Vijayakumar N and Plale B, "Towards Low Overhead Provenance Tracking in Near Real-Time Stream Filtering," Provenance and Annotation of Data, vol. 4145, pp. 46-54, 2006.

[13] Wang X and Reeves D S, "Robust Correlation of Encrypted Attack Traffic Through Stepping Stones by Manipulation of Interpacket Delays," Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 20-29, 2003.

[14] Wang X, Chen S, and Jajodia S, "Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems," Proc. IEEE Symp. Security and Privacy (SP), pp. 116-130, 2007.