

Distance-Preserving Pseudonymization for Timestamps and Spatial Data

Florian Kerschbaum
SAP Research
Karlsruhe, Germany
florian.kerschbaum@sap.com

ABSTRACT

The need for privacy in intrusion detection data, such as audit logs is widely recognized. The prevalent method for privacy protection in audit logs is pseudonymization (and suppression). There is a clear trade-off between the privacy of a pseudonymization technique and its utility for intrusion detection. E.g., for IP addresses a method for prefix-preserving pseudonymization has been developed, that allows pseudonymized IP addresses to be still grouped into subnets. This paper describes a pseudonymization technique for timestamps that is distance preserving. I.e. given two pseudonymized timestamps one can compute the distance δ , if δ is below or equal to an agreed threshold d and one cannot compute δ if $\delta \geq 2d$. We extend our technique for two-dimensional spatial data, e.g. location of objects or persons. We also evaluate the privacy any such distance-preserving technique can provide for timestamps theoretically and on real-world log data.

Categories and Subject Descriptors

C.2.4 [Computer-Communication Networks]: Distributed Systems—*distributed applications*; D.4.6 [Operating Systems]: Security and Protection—*Cryptographic controls*

General Terms

Algorithms, Security

Keywords

Pseudonymization, Distributed Intrusion Detection, Privacy, Hash Functions

1. INTRODUCTION

In intrusion detection it is often necessary to detect if two events occurred within a certain time span of each other. This is done by comparing the timestamps of the two events and computing a simple arithmetic difference (distance). In

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WPES'07, October 29, 2007, Alexandria, Virginia, USA.

Copyright 2007 ACM 978-1-59593-883-1/07/0010 ...\$5.00.

B2B scenarios privacy is of the utmost importance. If two companies want to do joint intrusion detection, they want to detect intrusions (which they might not be able to do alone), but do not want to share more information than is absolutely necessary to achieve the task. E.g. sending the unencrypted logs (and with it the associated plain-text timestamps) to the other party is unacceptable.

In this paper we present a solution, such that the two parties (Alice and Bob) can send their timestamps to a third party (Trudy), such that

1. Trudy does not get to know the value of the timestamp.
2. Trudy can compute the distance between two timestamps, if the distance is below or equal to a threshold d .
3. Trudy cannot compute the distance between two timestamps, if the distance is above or equal to $2d$.

This solution allows Alice and Bob to send their logs to Trudy which can then perform the intrusion detection and in this process detect events that are timely related.

2. RELATED WORK

The necessity for and benefits of collaborative intrusion detection have been discovered early [1, 6]. The need for privacy of the log data in intrusion detection system has been discussed even earlier [10, 13].

Several proposals for secure and private collaborative intrusion detection have been made since, e.g. [9, 17, 18]. In [9] light-weight practical data sanitization techniques geared towards detection capabilities are presented in an overall framework. No new pseudonymizations are introduced, and suppressing sensitive information entirely is suggested. In [17] this suppression is structured in a concept hierarchy, such that it can be controlled better. An actual tool that allows multiple levels of pseudonymization is described in [18].

All these methods use pseudonymization as the means for privacy protection, but the level of privacy they provide is not always clear. An analysis of this has been suggested in [14], as well as the search for new techniques and frameworks. An analysis of the requirements for distributed intrusion detection and the possible privacy implications is given in [5]. It particularly states the need for distance comparison for timestamps and concludes that no current pseudonymization method meets those needs.

An overview of the current timestamp pseudonymization techniques is given in [18]. It includes enumeration, time

shifting and field suppression. Except for time shifting none is distance-preserving and time-shifting does not provide the same level of privacy as our scheme, especially in the case of continuous or real-time detection.

Several other pseudonymization techniques have been proposed that allow specific operations. Most importantly prefix-preserving pseudonymization for IP addresses in [16]. For an active attack on any IP address pseudonymization see [3]. A number of pseudonymization techniques including for partial ordering within a set have been proposed in [8]. A programming framework for packet trace sanitization, in particular packet contents, is described in [12]. When using pseudonymization reidentification can be important and [2] proposes a method for automatic re-identification using Shamir’s threshold secret sharing scheme.

Juels and Wattenberg in [7] extend a scheme from [4] to compare hashed values with Hamming distance. There schemes allow the efficient comparison of a pseudonym with a timestamp while our scheme allows for the efficient comparison of pseudonyms. As a consequence, we allow for bulk comparisons by a third party with linear time and communication cost, as opposed to quadratic time and communication cost in their schemes where the distance for each timestamp to each grid point would need to be transmitted.

The privacy of audit data can also be preserved using private database and secure computation techniques. A searchable, but encrypted audit log has been described in [15]. How these mechanisms can be used in distributed intrusion detection is subject of future research.

3. NOTATION

We denote the message authentication code of x with key s as by $MAC(x, s)$ and the concatenation of strings x, y by $x.y$.

4. ALGORITHM

4.1 Setup

Alice and Bob agree on a common shared secret s which is sufficiently hard to guess for Trudy. Then, Alice and Bob agree on a threshold value d for the maximum distance comparisons. Furthermore, they commonly choose a random value r between $0 \leq r < d$.

4.2 Timestamp Preparation

Alice and Bob perform the following steps for each of their timestamps t :

1. Compute a lower grid point $l = d \cdot \lfloor \frac{t-r}{d} \rfloor + r$.
2. Compute an upper grid point $u = d \cdot \lceil \frac{t-r+1}{d} \rceil + r$.
3. Compute the distance m to l as $m = t - l$.
4. Compute the distance v to u as $v = t - u$.
5. Then send the timestamp tuple $\mathbf{t} = \langle MAC(l, s), m, MAC(u, s), v \rangle$ to Trudy.

We refer to l and u as well as their hashed counterparts as grid points.

4.3 Distance Computation

Trudy can compute the distance $\delta = |t - t'|$ of two timestamps t and t' from the timestamp (tuples) $\mathbf{t} = \langle g_1, h_1, g_2, h_2 \rangle$ and $\mathbf{t}' = \langle g'_1, h'_1, g'_2, h'_2 \rangle$ with the following algorithm:

Case 1: $g_i \neq g'_j \forall i, j: \delta > d$

Case 2: $\exists g_c = g_i = g'_j: \delta = |h_i - h'_j|$

4.4 Visualization

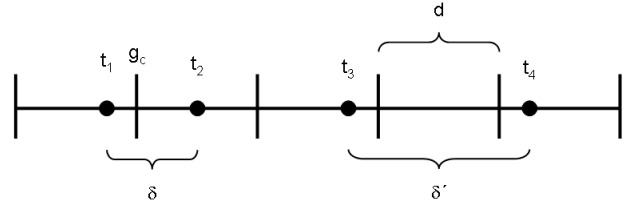


Figure 1: Distances of 4 timestamps

One can imagine the timestamps on a scale from left to right. Then the grid points divide the scale into equal-sized sections. The preparation algorithm computes the two grid points closest to the timestamp: l is the lower one and u is the upper one. The difference from the grid points is sent in plain-text to Trudy, i.e. in some sense the lower bits are leaked (but their exact values are protected by r). Figure 1 shows the timestamps t_1 and t_2 (as dots on the scale) with distance $\delta < d$ and common grid point g_c (grid points are shown as line markers on the scale) and the timestamps t_3 and t_4 with distance $\delta' > d$ and without any common grid point.

5. ATTACK

Assume Trudy has access to a black-box device that tells her for any two tuples the distance δ , if $\delta < d$ or indicates otherwise. This device is a stricter abstraction of our algorithm, which actually computes the difference (and not just the distance) and allows the computation of some differences $\delta > d$ (but $\delta < 2d$), i.e. everything an attacker can do with this black-box device he can do with our pseudonym tuples. Given this device and a dense data set T of tuples t_1, \dots, t_n , Trudy can align the timestamps on a linear scale. For that she picks two tuples \mathbf{t} and \mathbf{t}' by repeatedly querying the black-box device, until $|t - t'| = \delta \leq d$. She then searches the remainder of the timestamp for a timestamp t'' (again by querying the black-box device), such that $|t - t''| = \delta' \leq d$. Now, she asks the device for $\delta'' = |t' - t''|$ and $\delta''' = |t - t''|$. If $\delta'' \leq d$ which is known to her, then if $\delta = \delta' - \delta''$, she can conclude that $t < t' < t''$ or, if $\delta''' = \delta' - \delta''$, then she concludes $t < t'' < t'$. If $t' - t'' > d$, she concludes that either $t' < t < t''$ or $t' > t > t''$ and that $\delta'' = \delta + \delta'$, i.e. she has computed a distance $\delta'' > d$ by inference over two other distances $\delta < d$ and $\delta' < d$. Given enough data points Trudy can order all the timestamps along the scale. The direction ($<$ or $>$) is unknown to Trudy, if only distances can be computed. We can achieve the same for difference computation, if we flip a coin and accordingly multiply each timestamp with -1 or not before preparing it.

The problem is even harsher on our algorithm due to the grid points. Trudy only needs to align the grid points along

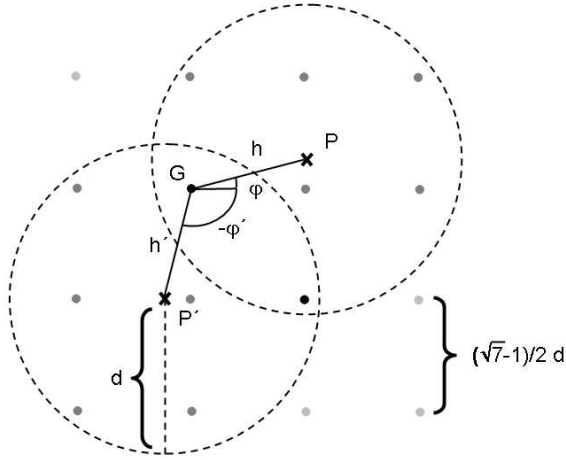


Figure 2: 2D distance computation

the scale and the timestamps will follow, but we showed above that it is unavoidable by any solution to the problem.

6. EXTENSION TO 2D

This section outlines the extension to two dimensions. This might be useful for intrusion detection or analytics on two-dimensional data. The basic properties remain: one can always compute the distance if the distance is below or equal to a threshold d and one cannot compute the distance if the distance is above $2d$.

Let $\alpha = \frac{\sqrt{7}-1}{2} \approx .82$. First, Alice and Bob agree on three random numbers $0 \leq r_x < \alpha d$, $0 \leq r_y < \alpha d$, $0 \leq r_\phi < 2\pi$. Similar to the grid points on the line, imagine the plane full of grid points with distance αd along the axis shifted by (r_x, r_y) . Let $G_{x,y}$ denote the grid point at coordinates (x, y) .

Let P be a point in the plane with coordinates (x, y) . The pseudonymization of P is the set of grid points G_1, \dots, G_n that are within distance d along with their distance h_i to P and the angle ϕ_i between line $\overline{G_i P}$ and the x-axis rotated by r_ϕ . For the pseudonymization $\{(MAC(x_1, y_1, s), h_1, \phi_1), \dots, (MAC(x_n, y_n, s), h_n, \phi_n)\}$ of P we write \mathbf{P} . To pad the pseudonymization \mathbf{P} to an equal length one can add random grid points (distances, and angles) that do not match any other grid point (with very high probability) up to a maximum of 7, i.e. $n \leq 7$ (proof omitted).

The distance between two pseudonymized points \mathbf{P} and \mathbf{P}' can then be found with this algorithm.

1. Find a common grid point G in \mathbf{P} and \mathbf{P}' by comparing the hashes.
2. Compute the distance δ

$$\delta^2 = h^2 + h'^2 - 2hh' \cos(\phi - \phi')$$

Figure 2 shows an example with P and P' which have two grid points in common including G . One can easily see that the distance computation follows from the cosine law.

For correctness it remains to show that two points within distance d always have a common grid point and that points

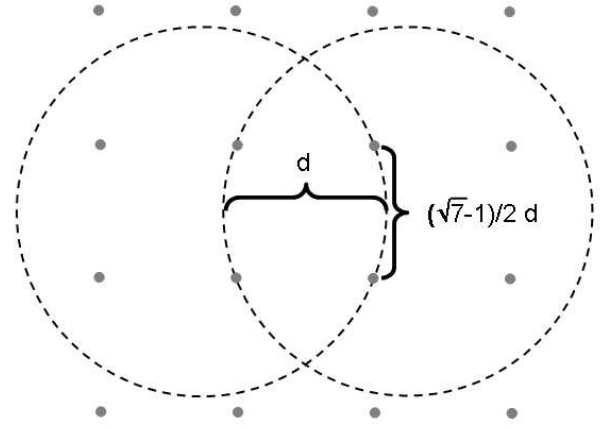


Figure 3: Intersection of two points with distance d

with distance larger than $2d$ do not have such a common grid point. The case of distances larger than $2d$ follows from the empty intersection of the circles around the two points. To get an intuition that two points P and P' within distance d always have at least one common grid point, look at figure 3. The maximal inscribed square of the intersection of the two circles around P and P' has edge length αd . Due to the continuous layout of the grid points in squares, the minimal grid point distance, such that no grid point is inside the intersection is greater than αd and therefore the intersection always contains at least one grid point.

7. DISCUSSION

7.1 Privacy vs. Utility

The larger the distances we can compute, the larger the utility for intrusion detection, but also the less privacy our scheme provides. This section aims at estimating the limits on the privacy provided by our scheme in terms of the parameters d and the arrival rate of new events.

We can model the arrival of events (and corresponding log entries with timestamps) with a constant arrival rate λ , e.g. $5 \frac{1}{min}$. Then the distribution of the time between two consecutive events is exponential with parameter λ and mean $\mu = \frac{1}{\lambda}$ (12s in the example). The distribution remains exponential in the case of multiple collaborating sources, since the distribution of a random variable $Y = \min(X_1, \dots, X_N)$ is exponential distributed, if all X_i are exponential distributed.

We can express the threshold d in terms of μ : $d = a\mu$ (e.g. $d = 24s, a = 2$). Then using the cumulative density function of the exponential distribution we can compute the probability that an event occurs within time d or less. But we must model the interval $]d, 2d[$ where the probability that the distance $d+i$ ($0 < i < d$) between two timestamps can be computed is $1 - \frac{i}{d}$ (proof omitted), so we consider the average of distances below or equal to $\frac{3}{2}d = \frac{3}{2}a\mu$ as computable. The probability p that a distance δ of an event to its previous one is computable (i.e. $\delta \leq \frac{3}{2}d$) is then:

$$p = 1 - e^{-\frac{3}{2}a}$$

We call a consecutive series of timestamps t_1, \dots, t_n where

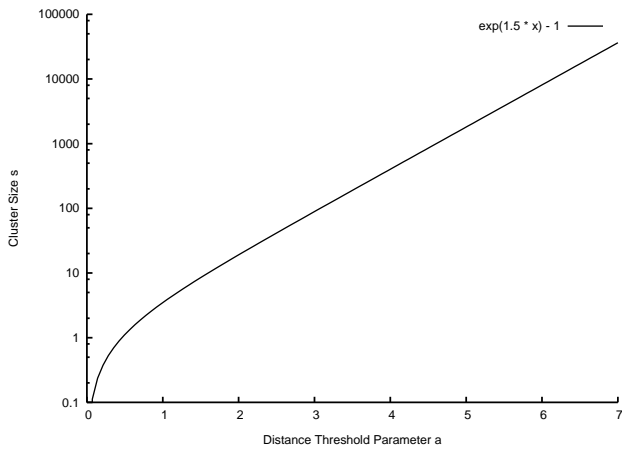


Figure 4: Expected cluster size

the distance between two consecutive timestamps t_i and t_{i+1} is less than or equal to $\frac{3}{2}d$ a cluster. Due to the attack described in Section 5 the distances between all timestamps in a cluster can be computed by inference. A cluster is broken every time an event occurs with a distance $\delta > \frac{3}{2}$ to its predecessor and therefore the expected size s of a cluster is

$$s = (1 - p) \sum_{i=0}^{\infty} ip^i = \frac{p}{1 - p} = e^{\frac{3}{2}a} - 1$$

In our example ($a = 2$) the expected cluster size is $s \approx 19$ timestamps. Figure 4 displays the expected cluster size for the threshold parameter a .

8. CONCLUSIONS

We showed a pseudonymization technique which allows Alice and Bob to hide their timestamps (in log data), but a third party Trudy can still compute their distance as long as the distance is below an agreed threshold. This allows Trudy to perform collaborative private intrusion detection for events that are timely related. The method has been extended to two dimensions for spatial data. We showed that any such mechanism as ours is limited in the privacy it can provide, but we believe that it has good applications for sparse data sets. Future work is to allow Trudy even more complex operations efficiently other than distance computations.

9. REFERENCES

- [1] T. Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4), 2000.
- [2] J. Biskup, and U. Flegel. Threshold-based Identity Recovery for Privacy Enhanced Applications. *Proceedings of the 7th International ACM Conference on Computer and Communications Security*, 2000.
- [3] T. Brekne, and A. Årnes. Circumventing IP-Address Pseudonymization. *Proceedings of the 3rd IASTED International Conference on Communications and Computer Networks*, 2005.
- [4] G. Davida, Y. Frankel, and B. Matt. On Enabling Secure Applications Through Off-Line Biometric Identification. *Proceedings of the IEEE Symposium on Security and Privacy*, 1998.
- [5] U. Flegel, and J. Biskup. Requirements of Information Reductions for Cooperating Intrusion Detection Agents. *Proceedings of the International Conference on Emerging Trends in Information and Communication Security*, 2006.
- [6] M. Huang, R. Jasper, and T. Wicks. A large scale distributed intrusion detection framework based on attack strategy analysis. *Computer Networks*, 31(23-24), 1999.
- [7] A. Juels, and M. Wattenberg. A fuzzy commitment scheme. *Proceedings of the 6th ACM conference on Computer and communications security*, 1999.
- [8] A. Lee, P. Tabriz, and N. Borisov. A Privacy-Preserving Interdomain Audit Framework. *Proceedings of the Workshop On Privacy In The Electronic Society*, 2006.
- [9] P. Lincoln, P. Porras, and V. Shmatikov. Privacy-Preserving Sharing and Correlation of Security Alerts. *Proceedings of the 13th USENIX Security Symposium*, 2004.
- [10] E. Lundin, and E. Jonnson. Privacy vs. Intrusion Detection Analysis. *Proceedings of International Symposium on Recent Advances in Intrusion Detection*, 1999.
- [11] A. Menezes, P. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography. *CRC Press*, 1996.
- [12] R. Pang, and V. Paxson. A high-level programming environment for packet trace anonymization and transformation. *Proceedings of the ACM Conference on Applications, technologies, architectures, and protocols for computer communications*, 2003.
- [13] M. Sobirey, S. Fischer-Hübner, and K. Rannenburg. Pseudonymous audit for privacy enhanced intrusion detection. *Proceedings of the 13th IFIP International Conference on Information Security (SEC)*, 1997.
- [14] A. Slagell, and W. Yurcik. Sharing Computer Network Logs for Security and Privacy: A Motivation for New Methodologies of Anonymization. *Proceedings of the Workshop on the Value of Security through Collaboration*, 2005.
- [15] B. Waters, D. Balfanz, G. Durfee, and D.K. Smetters. Building an Encrypted and Searchable Audit Log. *Proceedings of the Internet Society Network Distributed Systems Symposium*, 2004.
- [16] J. Xu, J. Fan, M. Ammar, and S. Moon. Prefix-Preserving IP Address Anonymization: Measurement-Based Security Evaluation and a New Cryptography-Based Scheme. *Proceedings of the 10th IEEE International Conference on Network Protocols*, 2002.
- [17] D. Xu, and P. Ning. Privacy-Preserving Alert Correlation: A Concept Hierarchy Based Approach. *Proceedings of the 21st Annual Computer Security Applications Conference*, 2005.
- [18] J. Zhang, N. Borisov, and W. Yurcik. Outsourcing Security Analysis with Anonymized Logs. *Proceedings of the Workshop on the Value of Security through Collaboration*, 2006.