CIRCUMVENTING IP-ADDRESS PSEUDONYMIZATION

Tønnes Brekne and André Årnes
Centre for Quantifiable Quality of Service in Communication Systems*
Norwegian University of Science and Technology
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway
tonnes|andrearn@q2s.ntnu.no

ABSTRACT
This paper presents an attack that circumvents anonymization of IP addresses in IP network traffic data in $\mathcal{O}(n^2)$ time, or $\mathcal{O}(n)$ time under certain circumstances. The attack is based on packet injection, and circumvents *all* anonymization techniques that assign a static and unique pseudonym to an IP address. It turns out that the packet injection itself, as well as the extraction of the corresponding anonymized header data, are the most time-consuming steps.

KEY WORDS
Network Security, Network Monitoring, Anonymity, Traffic Analysis

1 Introduction

This paper presents an attack against anonymized IP addresses in passive monitoring data. The attack works against all anonymization systems where each IP address has a constant and unique anonymized value (pseudonym). This work was done while examining candidate solutions for anonymization of passive monitoring data in the context of the LOBSTER[1] and SCAMPI[2] projects.

Passive measurement of IP networks collect real traffic data containing private and confidential information. Since such data can reveal corporate or personal habits, they should ideally be anonymized as far as possible. In many jurisdictions, such protection is required by law. Effective anonymization, however, tends to render information on network structures unusable for most analysis applications. Thus there is a case for providing configurable anonymization, as an adjustable compromise between two extremes. Furthermore, law enforcement applications may impose a requirement that anonymization schemes be revocable.

In [3] we demonstrated how prefix-preserving pseudonymization of IP addresses in passive IP traffic measurements could be attacked efficiently in the presence of an active adversary. We proceeded to strengthen prefix-preserving schemes against such attacks and presented a method for strengthening hash-based IP address anonymization. We subsequently developed the attack presented in this paper, which is a more general attack that also compromises our strengthened anonymization techniques. The attack is a special case of the cryptographic chosen-plaintext attack, as applied to network monitoring pseudonymization schemes.

2 Background on Anonymization and Pseudonymization

There is a fine distinction between *anonymization* and *pseudonymization*. In this section, we present some common primitives for achieving anonymity and pseudonymity.

2.1 Anonymization

Anonymization tries to achieve "the state of being not identifiable within a set of subjects, the anonymity set" [9]. There are several ways of achieving this goal.

Data removal is the irreversible deletion of data, often done through replacing data with a constant or random value. One special case, known as *truncation*, is to replace part of a value by a constant.

Randomization is the substitution of sensitive information with random information. This provides unlinkability[3] between observations.

Generalization is the substitution of identifying data with less specific data, so that identifying individuals becomes harder. One example is the substitution of IP-addresses with their respective AS-numbers[4]. This preserves network topology, but the anonymity provided is

*The Centre for Quantifiable Quality of Service in Communication Systems, is a Centre of Excellence appointed by the Research Council of Norway, and funded by the Research Council, NTNU and UNINETT.

[1]LOBSTER is a pilot European Infrastructure for large-scale monitoring of broadband Internet infrastructure, see http://www.ist-lobster.org/.

[2]SCAMPI is a EU project for creating a scalable and programmable monitoring platform for the Internet, see http://www.ist-scampi.org/.

[3]Unlinkability means that "two or more items within a system are no more and no less related than they are related concerning a priori knowledge" [9].

[4]An Autonomous System (AS) is a collection of IP networks registered by a single entity. A unique AS-number is associated with each AS for routing purposes.

limited by the number of users associated with the AS-numbers in question.

2.2 Pseudonymization

Pseudonymization is the replacement of the actual identity by an alternate identity (a pseudonym). The use of pseudonymous network monitoring traces is discussed by Biskup and Flegel in [2] and by Sobirey, Fischer-Hübner, and Rannenberg in [13]. Some common primitives for achieving pseudonymization are given below.

Bijective mappings make pseudonymity possible. A pseudonym must be uniquely identifiable. This identifiability is the feature that makes the attack presented in this paper possible.

Cryptographic methods for anonymization of network traces are discussed in [14, 15, 8]. Note that any cryptographic anonymization scheme is subject to attacks on the cryptographic algorithms and the key management system.

Hashing employs surjective functions to produce data with constant length. In practice they can be applied to provide a pseudonymization scheme as defined above. Strictly speaking, they cannot in general be considered pseudonymous, since there is the possibility for collisions. However in the context of IP addresses, this possibility is usually considered negligible, if the hash function is preimage resistant, 2nd-preimage resistant, and collision resistant (see pages 323-324 in [7]).

Keyed hashing addresses a weakness with unkeyed hash functions, where any adversary can perform the same computations and build a dictionary for all possible IP addresses. In an experiment, we computed MD5 hashes for the entire IPv4 address space in a matter of hours on a regular PC. Such an attack is prevented by using a keyed hashing scheme.

2.3 Related Work

Much of the early work in anonymization was related to solving the problem of traffic analysis. Two solutions to this problem was published by Chaum in 1981 [4] and 1988 [5], called mix networks and dc networks respectively. Similarly, there has been an ongoing effort to improve traffic analysis methodologies in order to compromise such networks. Raymond [11] has provided an overview of existing traffic analysis research. Another overview, with a proposal for terminology for the field of anonymity, was published by Pfitzmann and Koehntopp [9].

The problem of anonymizing IP traffic monitoring data differs from the above mentioned problem of designing traffic analysis resistant networks in that the underlying network traffic in question generally is not protected against traffic analysis. As a consequence, the anonymization method of the monitoring system has to provide the necessary protection, while still keeping the necessary data for the monitoring applications. In [3] we studied prefix-preserving pseudonymization, as this is a solution specifically designed for monitoring data. This is further discussed below.

3 Anonymization of IP Traffic Monitoring Data

As discussed in the introduction, anonymization of IP traffic monitoring data calls for specialized anonymization schemes. In this section, we discuss anonymization schemes that have been designed for this purpose.

3.1 Prefix-preserving Pseudonymization

An anonymization scheme is prefix-preserving if, for any two original IP addresses sharing a k-bit prefix, their anonymized mappings will also share a k-bit prefix. The tools TCPdpriv, wide-tcpdpriv, and Crypto-PAn are examples of prefix-preserving schemes, as discussed in [14, 15]. Prefix-preserving pseudonymization is particularly suitable for anonymizing IP traffic monitoring data, as it preserves information about the network topology. As an initial example, we will provide a brief description of TCPdpriv.

TCPdpriv, written by Greg Minshall, stores a set of original and anonymized IP address pairs. When a new IP address arrives, it is compared with previous original IP addresses in order to identify the longest prefix match. The new IP address is anonymized by using the same anonymized prefix as that of its match, whereas the remaining part of the address is anonymized with a random value. Since new pseudonyms are generated using random values, TCPdpriv is not deterministic. The pseudonym for a given IP address will differ between TCPdpriv sessions.

3.2 Cryptographic Prefix-preserving Pseudonymization

Cryptographic prefix-preserving pseudonymization was proposed in [14, 15] as an improvement of TCPdpriv. Cryptographic prefix-preserving pseudonymization uses a cryptographic algorithm rather than a random value. In this way, the pseudonymization is uniquely determined by an encryption key. As a result, this method is deterministic, and allows consistent prefix-preserving pseudonymization in distributed environments and across sessions. This scheme has been implemented in the tool Crypto-PAn. Some improvements on Crypto-PAn were proposed in [12].

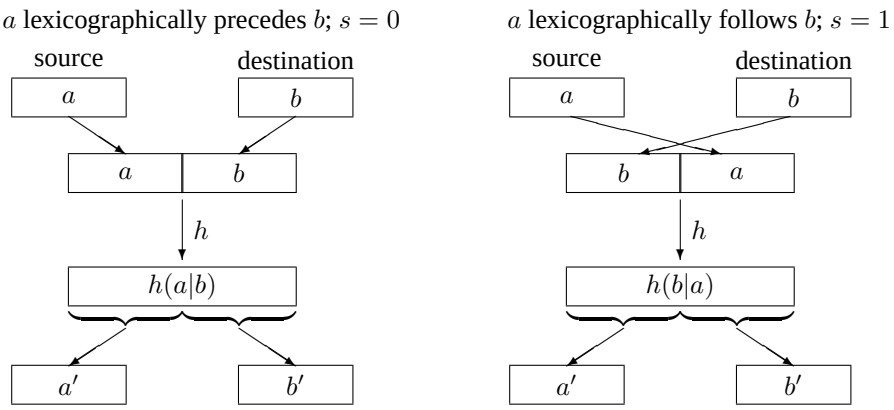

Figure 1. Illustration of block anonymization shows how it provides bidirectional traffic with a unique hashed identifier, which is equal for both directions.

3.3 Strengthened Hashing of IP Addresses

Another common technique for anonymizing IP addresses in traffic data, is to apply a cryptographically strong hash function to the plaintext IP address. This provides a plaintext search space containing 2^n elements, where n is the length in bits of each IP address[5].

In [3] we presented a scheme for constructing longer hashes by hashing *pairs of IP addresses*, in order to increase resistance to cryptographic attacks, as well as attacks employing packet injection. With this scheme, all traffic between two fixed parties A and B have the same pseudonyms, regardless of packet direction. Information about the packet direction is retained in a separate bit s. This scheme is illustrated in Figure 1 and described in Pseudocode 1.

PSEUDOCODE 1 *block-anonymization*(n, a, b, h)

IN: address length in bits n, source address a, destination address b, cryptographically strong hash function h generating output at least $2n$ bits long, or keyed encryption function h with blocklength $2n$

OUT: two n-bit blocks a' and b' replacing the plaintext addresses a and b, respectively. One bit s indicating whether a lexicographically precedes b or not.

if a *lexicographically precedes* b
    return last $2n$ *bits of* $h(a|b)$ *split into two*
        n*-bit bitstrings, along with* $s = 0$
else
    return last $2n$ *bits of* $h(b|a)$ *split into two*
        n*-bit bitstrings, along with* $s = 1$
end if

[5] 32 bits for IPv4 and 128 bits for IPv6.

3.4 Strengthened Prefix-Preserving Pseudonymization

It is possible to strengthen prefix-preserving pseudonymization with a technique similar to the one discussed above.

IP addresses pseudonymized with prefix-preserving pseudonymization are split into a series of l blocks, each block w_i bits in length. w_1 is the length of the most significant block, and w_l the length of the least significant block. Block l from source and destination are concatenated and encrypted, producing r_l. Block $l - 1$ from source and destination are concatenated, and then concatenated with r_l. This is then encrypted, producing r_{l-1}. This is repeated until block 1 from source and destination are concatenated along with r_2, and all $2n$ bits are encrypted. This is the essence of the algorithm described in Pseudocode 2 below.

PSEUDOCODE 2 *hardened-pseudonymization-2*$(n, a, b, g, l, \{w_i\}_{i=1}^{l}, e_k, \{f_i\}_{i=0}^{n-1})$

IN: address length in bits n, source address a, destination address b, a permutation function $g\{1, \ldots, 2n\} \longrightarrow \{1, \ldots, 2n\}$ the number l of sub-blocks, a list $\{w_i\}_{i=1}^{l}$ of sub-block lengths such that $\sum_{i=1}^{l} w_i = n$, a keyed block encryption function e_k, that encrypts k-bit blocks, a series $\{f_i\}_{i=0}^{n-1}$ of encryption functions $f_0, f_1(a_1), \ldots, f_{n-1}(a_1, \ldots, a_{n-1})$ which return one bit each

OUT: two n-bit blocks a' and b' replacing the plaintext addresses a and b, one bit s indicating whether a lexicographically precedes b or not

if a *lexicographically precedes* b
    apply prefix-preserving pseudonymization to a *to get* c
    apply prefix-preserving pseudonymization to b *to get* d
    $s \leftarrow 0$
else

apply prefix-preserving pseudonymization to a *to get* d
apply prefix-preserving pseudonymization to b *to get* c
$s \leftarrow 1$
end if
$i \leftarrow l$
$p \leftarrow 0$
while $i > 0$ *do:*
$p \leftarrow p - w_i$
encrypt the concatenation of bits $p+1, \ldots, p+w_i$ *of* c *and* d *with the last* $n-p$ *bits from any previous encryption, if any with* e_{n-p}
$i \leftarrow i - 1$
end for
call the resulting cryptotext block r
$a' \leftarrow$ *first* n *bits of* r
$b' \leftarrow$ *last* n *bits of* r
return a', b', s

Pseudocode 2 encrypts successively longer concatenations of corresponding blocks from source and destination addresses. Thus each header is now coupled to *both* addresses in a communication. An adversary now sees all pseudonymized pairs.

4 The Attack

We propose a new attack based on IP packet injection. There are two variations of the proposed attack: one for the strengthened pseudonymization algorithms presented in [3], and one for individually pseudonymized addresses. This section provides a description of the context and an overview of injection attacks, as well as a detailed description of the two attack variations.

4.1 Context and Threat Model

It is important to be aware of the circumstances which make the attack possible. The underlying scenario is that an organization (such as a telecommunications operator or a non-profit organization) releases IP traffic monitoring data in a pseudonymized form. The IP traffic data is typically captured from publicly available backbone networks using programmable passive network monitoring cards capable of capturing high-bandwidth traffic while performing on-board data anonymization[6]. The pseudonymized data is made available to third parties for analysis.

The main threat is that an adversary is able to compromise the anonymization scheme and reidentify anonymized network traces. This will enable the adversary to obtain private or confidential information through the analysis of traffic patterns. Given the circumstances in which traffic data is made available, the following is assumed:

Assumption 1 *The adversary is capable of ensuring that injected packets are captured by at least one passive sensor.*

[6]Examples of such cards are SCAMPI cards and Endace DAG cards.

If the adversary is capable of using more than one sensor or even has direct access to monitoring interfaces, one can assume that the adversary's efficiency will be further increased. This does, however, not appear to impact the complexity of the attack presented in this paper.

Assumption 2 *The adversary may send forged network traffic with arbitrary source and destination IP addresses.*

In other words, the adversary is capable of performing an attack similar to a cryptographic *chosen plaintext attack*.

4.2 On Injection Attacks

Given the threat model in section 4.1, an adversary can send an IP packet with arbitrary source and destination IP addresses, either through IP spoofing or by sending packets from a variety of locations. By forging the packet header in such a way that it is recognizable in its anonymized form, an adversary is able to find an exact match between an original and an anonymized IP address.

As already noted, the injection attacks described in this paper are special cases of cryptographic chosen-plaintext attacks. See [1] for a general treatment of such attacks.

The use of repeated messages for revealing the correspondence between original and anonymized data is discussed by Chaum in [4] and referred to as *flush attacks* by Raymond in [11]. The forging of packet headers for reidentification purposes is related to the *message tagging* attack described by Raymond in [11]. It is further discussed in the context of IP traffic monitoring data in [3, 10].

In the case of prefix-preserving pseudonymization, a successful attack also reveals information about the prefix for all other addresses with identical prefixes. Using this, an adversary can build a binary tree mapping pseudonymized addresses to original IP addresses.

4.3 Attacking Strengthened Pseudonymization

The strengthened algorithms in [3] pseudonymize *pairs* of IP addresses instead of pseudonymizing the addresses individually. Because the addresses in each pair are sorted prior to pseudonymization, and an extra order bit stored, it is easy to identify packets belonging to the same session, as well as the direction of the packet.

In [3] the following assumption about the adversary's intentions was made:

Assumption 3 *The adversary wants to pick out all pseudonymized packets containing the IP address* a *in their headers.*

The attack is enabled by relaxing assumption 3 to assumption 4.

Assumption 4 *The adversary wants to pick out all pseudonymized packets containing the IP address pairs (c, d) in their headers such that either $c = a$ and $d \in B$ or $c \in B$ and $d = a$, where a is a fixed IP address and B is a fixed set of IP addresses.*

Based on assumption 4, we have a "set of interest" with $|B|$ pairs of addresses. Assign unique positive integer weights to all address pairs, and inject this number of packets into the network. Doing this so as to minimize the number of injected packets required, takes at least $\sum_{j=1}^{|B|} j = |B|(|B|+1)/2$ packets, which is order $\mathcal{O}(n^2)$. For each pseudonymized pair, record the number of times it shows up in the traffic data. Then compare with the plaintext pairs to match them. This can be done in $\mathcal{O}(|B| \log |B|)$ time by sorting both lists of pairs by their frequencies of occurrence in the traffic data.

4.4 Attacking Individually Pseudonymized Addresses

The relaxation represented by assumption 4 is only necessary when attacking the strengthened pseudonymization schemes presented in [3]. It is not necessary if IP addresses are pseudonymized individually. IP addresses that have been pseudonymized without the strengthening techniques can be compromised in a similar manner. The adversary assigns to each address of interest an integer weight. Since two individual addresses can be put into each packet header, at least $\frac{1}{2}\sum_{j=1}^{|B|+1} j = (|B|+1)(|B|+2)/4$ packets are needed, which is still $\mathcal{O}(n^2)$.

4.5 Analysis

Thus circumventing conventional pseudonymization techniques, as well as the strengthened pseudonymization scheme, requires $\mathcal{O}(n^2)$ packets and thus $\mathcal{O}(n^2)$ time.

If, however, packet injection can be injected and extracted in the same order without packet loss or reordering, an adversary can perform the attacks in $\mathcal{O}(n)$ time using $\mathcal{O}(n)$ packets. Note that such an approach requires that packets are injected with a minimal separation in time in order to minimize the amount of packet reordering. As a special case of the attacks described in this paper, *any single* pseudonymized IP address or IP address pair can be reidentified in $\mathcal{O}(1)$ time.

Finally it is important to keep in mind that these attacks apply to *all* types of anonymized traces of IP traffic, as long as IP addresses are pseudonymized with static pseudonyms. The attacks are *not* limited to prefix-preserving pseudonymization techniques, nor conventional hashing techniques. They apply to all static pseudonymizations of IP addresses, and to all static anonymization techniques that are "almost" pseudonymous, such as hash functions. As mentioned above, the probability of collisions in hashes of IP addresses is negligible. Thus the hashing scheme can for practical purposes be considered a pseudonymization scheme, which means that this attack strategy should succeed with a very high probability.

5 Countermeasures

We have identified three possible ways of attempting to counter the attack presented in section 4.

Employ non-static pseudonyms for IP addresses. This is a potential research topic, due to the functional requirements for traffic data. Traffic data should ideally be efficiently searchable and indexable, as well as effectively anonymous, and thus resistant to our attacks.

Employ mandatory sampling at the monitoring sensors. This will increase the cost of performing a successful injection attack. This necessitates the use of redundant injected packets to ensure capture of the relevant packets in the trace, and it also increases the probability that the injected packets will not have correct relative weighting in the traffic data. In other words it increases the cost of the attack, as well as the probability of detecting it.

Detect and prevent packet injection attempts. This can for instance be done through the detection and removal of malformed packets. However, this could impact measurements, such as measurements designed to capture network errors. Also, a resourceful adversary would most likely be able to circumvent such a protection system.

6 Conclusions

We have presented two variants of what is essentially the same attack, employing packet injection, that can compromise any form of static pseudonymization of IP addresses. This attack demonstrates that static pseudonymization of IP addresses does not provide sufficient privacy in traffic data released for analysis purposes. This is a disquieting conclusion to say the least. There is a very real possibility that such attacks may already have taken place.

A corollary of this conclusion is that extreme care is required when implementing anonymization schemes for IP traffic monitoring data. Failure to understand the efficiency of traffic analysis, in particular if packet injection is possible, may result in very weak anonymity for the users of the monitored networks.

An alternate class of pseudonymization techniques is in urgent need of research to enable the secure release of pseudonymized and anonymized IP traffic data.

Acknowledgements

We would like to thank our colleagues at the Centre for Quantifiable Quality of Service in Communication Sys-

tems, Svein J. Knapskog and Karin Sallhammar in particular, for feedback on our paper.

References

[1] M. Bellare, and P. Rogaway, Introduction to Modern Cryptography, *course notes, University of California, San Diego*, 2004.

[2] J. Biskup and U. Flegel, On Pseudonymization of Audit Data for Intrusion Detection, *Workshop on Design Issues in Anonymity and Unobservability*, Springer-Verlag, LNCS 2009, 2000.

[3] T.Brekne, A. Øslebø, and A. Årnes, Anonymization of IP Traffic Monitoring Data—Attacks on Two Prefix-preserving Anonymization Schemes and Some Proposed Remedies, *Privacy Enhancing Technologies 2005*.

[4] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, *Communications of the ACM*, *4*(2), 1981.

[5] D. Chaum, The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability, *Journal of Cryptology*, Vol. 1, Pages 56–75, 1988.

[6] K. Cho, K. Mitsuya, and A. Kato, Traffic Data Repository at the WIDE Project, *Proceedings of FREENIX Track: 2000 USENIX Annual Technical Conference*, 2000.

[7] A. J. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[8] M. Peuhkuri, A Method to Compress and Anonymize Packet Traces, *Internet Measurement Workshop 2001*, pages 257–261, 2001.

[9] A. Pfitzmann and M. Koehntopp, Anonymity, unobservability, and pseudonymity—a proposal for terminology, *Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[10] R.Ramaswamy, N. Weng, and T. Wolf, An IXA-Based Network Measurement Node, *Proc. of Intel IXA University Summit*, 2004.

[11] J. F. Raymond, Traffic Analysis: Protocols, Attacks, Design Issues and Open Problems, *Workshop on Design Issues in Anonymity and Unobservability*, LNCS 2009, Springer-Verlag, 2000.

[12] A. Slagell, J. Wang, and W. Yurick, Network Log Anonymization: Application of Crypto-PAn to Cisco Netflows, *IEEE Workshop on Secure Knowledge Management (SKM)*, 2004.

[13] M. Sobirey, S. Fischer-Hübner, and K. Rannenberg, Pseudonymous audit for privacy enhanced intrusion detection, *IFIP TC11 13th International Information Security Conference (SEC'97)*, page 151 – 163, 1997.

[14] J. Xu, J. Fan, M. Ammar, and S. B. Moon, On the Design and Performance of Prefix-preserving IP Traffic Trace Anonymization, *Proceedings of the ACM SIGCOMM Internet Measurement Workshop 2001*.

[15] J. Xu, J. Fan, M. Ammar, and S. B. Moon, Prefix-Preserving IP Address Anonymization: Measurement-Based Security Evaluation and a New Cryptography-Based Scheme, *Proceedings of the IEEE International Conference on Network Protocols*, 2002.