TASP: Towards Anonymity Sets that Persist

Jamie Hayes
University College London
j.hayes@cs.ucl.ac.uk

Carmela Troncoso
IMDEA Software Institute
carmela.troncoso@imdea.org

George Danezis
University College London
g.danezis@ucl.ac.uk

ABSTRACT

Anonymous communication systems are vulnerable to long term passive "intersection attacks". Not all users of an anonymous communication system will be online at the same time, this leaks some information about who is talking to who. A global passive adversary observing all communications can learn the set of potential recipients of a message with more and more confidence over time. Nearly all deployed anonymous communication tools offer no protection against such attacks. In this work, we introduce TASP, a protocol used by an anonymous communication system that mitigates intersection attacks by intelligently grouping clients together into anonymity sets. We find that with a bandwidth overhead of just 8% we can dramatically extend the time necessary to perform a successful intersection attack.

1. INTRODUCTION & MOTIVATION

Anonymous communication systems (ACS) safeguard against the ability to link a sender and receiver of a communication. ACS route messages through a series of relays. Mix networks require only a small number of relays to be honest to provide unlikability, usually at the expense of latency, whereas low-latency designs such as the Tor network [5] require a larger fraction of honest relays to assure a clients anonymity.

Recently proposed ACS, such as Vuvuzela [11], Riffle [9], AnonPoP [7] and cMix [1], offer strong security against a powerful adversary, as all clients communicate in rounds, either by choice (sending a real message) or by sending a cover message that is indistiguishable from a real message. We argue that security based on the assumption of constant client participation is impracticle and weakens the usability of such a system, which is the ultimate goal for any communication tool.

Today, Tor is by far the most widely used ACS and uses Onion Routing to deliver messages to clients. Tor does not operate in mixing rounds and does not require clients to send cover messages but is vulnerable to statisical disclosure attacks [3] and long term intersection attacks [2, 8, 4].

WPES'16, October 24 2016, Vienna, Austria
© 2016 ACM. ISBN 978-1-4503-4569-9/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2994620.2994635

Intersection attacks allow a global passive adversary, who observes all incoming and outgoing messages of an ACS, to link a targeted sender and receiver. To perform such an attack, when a client sends a message, the adversary records the recipients who were online and could potentially have been the intended target. This is done repeatedly until the intersection of all the potential recipients reveals a single client. Such an attack is not purely theory, recently the FBI were able to de-anonymize a person of interest by interesecting the number of Tor users within a specific geographic location [6]. While intersection attacks are deterministic, statistical disclosure attacks allow an adversary to estimate, among a group of clients, the likelihood that a target recipient was the receiver of a communication. Inevitably, with the churn of an ACS user base, messages will become linkable via statistical disclosure and intersection attacks eventually.

In this work, we consider simple long term intersection attacks, when the rate at which a client can send messages is fixed. We investigate how well an ACS can resist an intersection attack when an adversary has complete view of the system. To this end, take the following scenario: a service offers clients the ability to anonymously post a blog. For example, this could be via Tor where each client hosts their own blog on a Tor hidden service. We explore the frequency that a service, protecting against intersection attacks, will allow a client to post against the utility of the service. For example, if a client is posting a blog they may be able to tolerate delays in the order of hours or days, whereas a live tweeting service reporting on current events will require shorter delays.

Over time, natural churn will occur in the user base of an ACS, some new clients may join whereas other clients may stop using the service altogether. Clients may start to post at different times to others within an anonymity set, or stop posting, thereby shrinking the anonymity set, and weakening resistance to an intersection attack. The aim of an ACS that resists intersection attacks is to delay this anonymity degradation process. Leakage is inevitable as we do not want to force all clients to send cover messages each round; rather we would like to intelligently group clients in to anonymity sets with clients who share similar communication patterns, thus providing a natural safeguard against rapidly shrinking anonymity sets.

In this work we attempt to answer the following: how does an ACS intelligently group clients in to anonymity sets? What utility can a client expect to gain by doing so? To answer the first question we introduce TASP, a protocol that

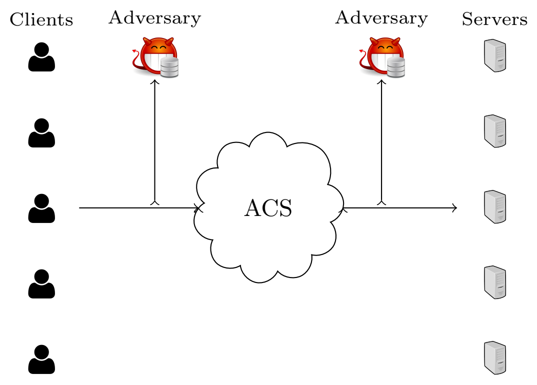

Figure 1: Clients post messages to servers through an ACS. The attacker eavesdrops on all incoming and outgoing messages.

can be used by an ACS to form anonymity sets. TASP introduces a bootstrapping phase that learns the posting patterns of each client, and then groups clients with similar communication patterns together into anonymity sets.

2. SYSTEM MODEL

We consider an ACS that operates in rounds. Each round is of some length of time, t. Messages are packaged in such a way that, to an external observer, they are indistinguishable from one another. All messages received by an ACS are kept in the system until the end of a round, they are then flushed and sent to their intended destination. Each client may send a maximum of one message per round. The length of a round is a parameter choice of the ACS and will be domain specific.

3. THREAT MODEL

Figure 1 shows the threat model we work under. We consider a global passive adversary with the ability to monitor, but not alter, all incoming and outgoing messages that pass through the ACS. Over time it is highly probably that each client will exhibit a unique communication pattern, the adversary will record for each client, the possible recipients, and repeatedly intersect this set to recover the intended receivers of the clients communications.

Since we rate-limit clients to sending at most one message per round, we degrade the ability for an adversary to probabilistically profile a client based on rates of sending. That is, a statistical disclosure attack in which an adversary attempts to assign probabilities of recipients of a communications will be no better than assigning a uniform probability to each client within their anonymity set.

4. PROTOCOL

We introduce two methods by which clients may be assembled into anonymity sets by an ACS.

4.1 Possibilistic Anonymity Sets

Buddies [12]. The Buddies architecture provides k-anonymity[1] to a client publishing to a shared board. The authors do this by denying a clients request to publish a message if the system deems them to be susceptible at that time to an intersection attack. Buddies comes at the expense of latency, essentially blocking a client from using the service once their anonymity set has dropped below some threshold. This is particularly problematic for usability and availability of low-latency publishing tools such as a live tweeting service.

Possibilistic Anonymity. We form *possibilistic anonymity sets* in the following way: when a client sends their first message their anonymity set is the group of clients that send in the same round. In subsequent rounds, if the original client sends a message and clients within their anonymity set do not, they are removed from the anonymity set. In the Buddies variant a threshold is specified, if the size of the anonymity set drops below this threshold the client will no longer be allowed to send messages as they are vulnerable to intersection attacks. In this work, we set the threshold to zero, essentially allowing a client to send messages until they can be uniquely identified. We do this because we do not want to inhibit the ability for a client to use the system until they are completely vulnerable to an attack.

4.2 TASP

TASP is run by an ACS and operates in two phases, a learning phase, and an online phase. The learning phase runs at the inception of the system, requiring all clients to send a message each round while learning which clients sent a real message and which sent cover messages. Clients are then assigned to anonymity sets based on their communication profiles. After this, the online phase runs, the system operates as normal with clients no longer expected to send cover messages. Since clients in anonymity sets share similar communication profiles, the adversary will require a greater amount of time until a successful intersection attack can be performed, when compared with no protection against such attacks.

Learning. Initially, in the learning phase, all clients must send a message in each round, either a real or cover message. During this phase, an external observer is unable to distinguish which clients are actively taking part in the protocol and which are sending cover messages. TASP is able to distinguish real messages from cover messages, and actively learns the posting patterns of all clients. It then groups clients with similar posting patterns together into anonymity sets. The number of rounds in which the learning phase runs is a parameter choice of the ACS.

Online. After the learning phase, TASP initiates the online phase, in which clients are no longer required to send cover messages. For each anonymity set, messages are flushed when either all clients within that set have sent a message, or at the end of a round.

[1]k-anonymity is the property that a client shares a communication profile that is similar to at least $k-1$ other clients.

If a round ends and clients within an anonymity set have sent messages, clients within the same anonymity set that have not sent messages are removed from the anonymity set and are no longer allowed to use the ACS as they are now vulnerable to an intersection attack.

Grouping mechanism. TASP uses dynamic time warping (DTW) [10] to group clients in to anonymity sets. DTW is an algorithm for measuring the similarity of two time sequences, that may be of unequal length[2]. DTW works as follows: given two times series:

$$A = a_1, a_2, ..., a_n, \quad B = b_1, b_2, ..., b_m$$

construct a warped path:

$$W = w_1, w_2, ..., w_l$$

where $\max(n, m) \leq l < n + m$ and $w_k = (i, j)$ where $i \in \{1, ..., n\}$, $j \in \{1, ..., m\}$, $k \in \{1, ..., l\}$. DTW constructs the optimal warped path to minimize the distance between A and B; the warped path distance is given by:

$$d(A, B) = d(W) = \sum_{k=1}^{l} \text{dist}(w_{ki}, w_{kj})$$

where dist is Euclidean distance and $\text{dist}(w_{ki}, w_{kj})$ is the distance between a_i and b_j at the k^{th} element of W.

At the end of the learning phase, for each client, TASP produces a time sequence of the rounds in which the client sent a real message. It then computes the similarity of this time sequence compared to all other clients' time sequences. If the similarity distance is below a certain threshold then the client is included in the anonymity set. The distance threshold is another design choice for the system, a large threshold will result in large anonymity sets, but with a downside that many of the clients may not share similar posting patterns. However, for small systems this may be the best that can be done without incurring high delays.

5. ANALYSIS

Here we give a brief theoretical analysis which aims to provide some intuition behind TASP. Let n clients be grouped in to an anonymity set as described in Section 4.2. Let the probability that client c_i posts in round t_j be given by $\delta_{c_{ij}}$, for $i \in \{1, ..., n\}, j \geq 1$.

Ideally we would like all clients within an anonymity set to have the same posting patterns; for some fixed $\delta_j \in [0, 1]$, $\delta_{c_{ij}} = \delta_j \; \forall i, j$. Then, the probability that the anonymity set decreases by k between two rounds is:

$$\binom{n}{k}(1 - \delta_j)^k \delta_j^{n-k}, \quad k \in \{1, ..., n\}$$

More generally, the probability that the anonymity set decreases by at least k, is equal to:

$$\sum_{l=k}^{n} \binom{n}{l}(1 - \delta_j)^l \delta_j^{n-l}, \quad k \in \{1, ..., n\} \qquad (\star)$$

Within an anonymity set, TASP groups clients that share a similar probability of participating in a given round. TASP groups clients such that for each round t_j, $\exists \delta_j, \epsilon \in [0, 1]$, such that $\delta_j \leq \delta_{c_{ij}} \leq \delta_j + \epsilon$, $\forall i \in \{1, ..., n\}$. If TASP can sort clients in to anonymity sets such that ϵ is neglibible then we can analyze the degradation of anonymity using $(\star)$.

[2]Further explanation of DTW can be found at [10].

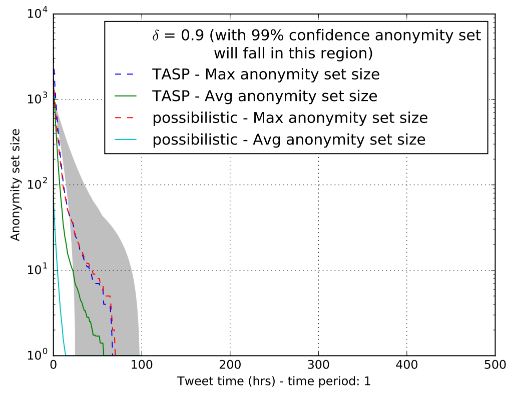

Figure 2: TASP and *possibilistic anonymity set* sizes for an online phase of 500 hours with a round length of 1 hour.

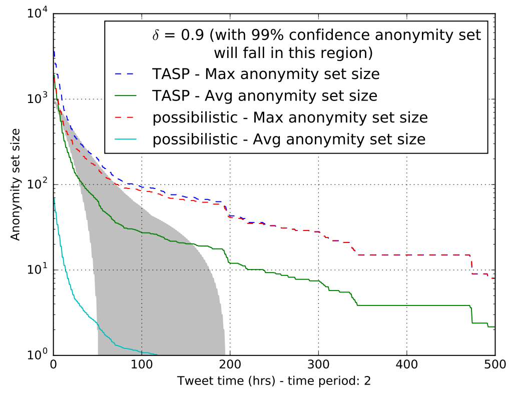

Figure 3: TASP and *possibilistic anonymity set* sizes for an online phase of 500 hours with a round length of 2 hours.

Example. Let $n = 1000, k = 100$. With $\delta_j = 0.9$, the probability that after the next round the anonymity set decreases to less than 900 clients is equal to $\sum_{l=100}^{1000} \binom{1000}{l} 0.1^l 0.9^{1000-l} = 0.52$. At $\delta_j = 0.91$, this probability becomes 0.15 and at $\delta_j = 0.95$, this probability becomes 8.41×10^{-11}. Clearly good groupings in which all members are equally likely to participate in rounds lead to persistent anonymity sets.

6. EVALUATION

We performed experiments on a dataset consisting of tweets posted by a number of clients throughout November 2012[3]. The length of a round in the learning phase was chosen to be one hour and the length of the learning phase was 24 rounds. We consider a subset of clients within this dataset that posted at least five times within the learning phase, allowing us to uncover information about their posting habits. This subset consisted of 9195 clients. Using TASP and *possibilistic anonymity sets*, we then examine the rate of degra-

[3]http://cnets.indiana.edu/groups/nan/webtraffic/websci14-data/

dation of anonymity sets in two cases, when a round in the online phase is one hour in length and when it is two hours in length[4]. For comparison, we also include a theoretical grouping of clients in which there is a 90% probability that any client within that set posts in a round. This allows us to establish if the anonymity sets created by TASP are well formed.

Figures 2 and 3 show the reduction in anonymity set sizes for online round lengths of one hour and two hours, respectively. We can see that the best performing TASP anonymity set is similar in size and rate of decrease to the best performing *possibilistic anonymity set* grouping of clients over the same period, but on average our method produces larger anonymity sets that decrease at a slower rate. If clients can tolerate a delay of two instead of one hour, their ability to withstand long term intersection attacks dramatically increase. Finally, for a grouping of 1000 clients in which each client has a 90% probability of sending a message in a round, there is a 99% chance the anonymity set will shrink to a single client within 200 hours. The average TASP anonymity set does not degrade to a single client within 500 hours, and so we can be sure that these anonymity sets do contain clients that share similar communication profiles.

Clearly our method has advantages over the *possibilistic anonymity set* approach as anonymity sets are more stable in size. Therefore it will be more costly for an attacker to mount an intersection attack. Additionally, only 8% more messages were required to be sent to the ACS to achieve this attack resistance, compared to *possibilistic anonymity sets.*

7. DISCUSSION & FUTURE WORK

TASP removes the unrealistic expectation that all clients can commit to participation in every round. However, currently it can only be applied to an ACS at its inception. After the learning phase TASP does not consider new clients joining the system, though this could be handled by requiring new clients to post at every round until the most appropriate anonymity set is found. In other words, each new joining client could have their own learning period for some amount of time after they join, after which they are not required to send cover traffic.

TASP's current method of client removal from anonymity sets is inflexible. Currently if just one client posts in a round, all other clients within that anonymity set are removed, leaving the client that posted a message vulnerable. This was an intentional decision, we did not want to inhibit clients behaviour in any way. Still, it may be beneficial to investigate how long a client can tolerate delays before posting a message. For example, TASP could flush messages from an anonymity set at the end of a round only if some threshold of clients within that set have posted message.

We also experimented with longer learning phases. We found that a longer learning phase results in more stable, long-lived anonymity sets, since there is more time to uncover the posting patterns of all clients, and so group clients in to appropriate anonymity sets. However, a longer learning phase will also require more cover messages to be sent; quantifying this trade-off is left for future work.

[4]The entire dataset consists of 27.8M tweets from many more clients over 720 hours.

8. REFERENCES

[1] David Chaum and Debajyoti Das and Farid Javani and Aniket Kate and Anna Krasnova and Joeri de Ruiter and Alan T. Sherman . cMix: Anonymization by High-Performance Scalable Mixing. Cryptology ePrint Archive, Report 2016/008, 2016. http://eprint.iacr.org/2016/008.

[2] O. Berthold and H. Langos. Dummy Traffic Against Long Term Intersection Attacks. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 110–128, Berlin, Heidelberg, 2003. Springer-Verlag.

[3] G. Danezis. Statistical Disclosure Attacks: Traffic Confirmation in Open Environments. In *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)*, pages 421–426. Kluwer, 2003.

[4] G. Danezis and A. Serjantov. *Statistical Disclosure or Intersection Attacks on Anonymity Systems*, pages 293–308. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[5] R. Dingledine, N. Mathewson, and P. F. Syverson. "Tor: The Second-Generation Onion Router". In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320, 2004.

[6] Gallagher, Kevin. "How Tor helped catch the Harvard bomb threat suspect". http://www.dailydot.com/crime/tor-harvard-bomb-suspect/. Accessed: June 10, 2016,.

[7] N. Gelernter, A. Herzberg, and H. Leibowitz. Two Cents for Strong Anonymity: The Anonymous Post-office Protocol. Cryptology ePrint Archive, Report 2016/489, 2016. http://eprint.iacr.org/.

[8] D. Kedogan, D. Agrawal, and S. Penz. *Limits of Anonymity in Open Environments*, pages 53–69. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[9] A. Kwon, D. Lazar, S. Devadas, and B. Ford. Riffle. In *Proceedings on Privacy Enhancing Technologies*, volume 2016, pages 115–134, 2015.

[10] S. Salvador and P. Chan. Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intell. Data Anal.*, 11(5):561–580, Oct. 2007.

[11] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*, SOSP '15, pages 137–152, New York, NY, USA, 2015. ACM.

[12] Wolinsky, David Isaac and Syta, Ewa and Ford, Bryan. Hang with your buddies to resist intersection attacks. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & 38; communications security*, CCS '13, pages 1153–1166, New York, NY, USA, 2013. ACM.

APPENDIX

A. TASP PROTOCOL ALGORITHMS

```
**Data**: $n$ clients $\{c_0, .., c_{n-1}\}$, round start time $t_0$,
        round stop time $t_{k-1}$, DTW distance function
        $f : P \times P \to \mathbb{R}$ where $P$ is the message profile of
        a client, distance threshold $d$.
**Result**: Anonymity set for each client.
for $i \leftarrow 0$ to $n-1$ do
    create message profile $P_i = [\ ]$ for client $c_i$.
    for $j \leftarrow 0$ to $k-1$ do
        receive $m_{ji}$, a message from client $c_i$ at time $t_j$.
        if $m_{ji}$ is not a cover message then
            Add $m_{ji}$ to $P_i$.
for $i \leftarrow 0$ to $n-1$ do
    $P_i = [m_{0i}, ..., m_{li}]$ where $l \leq k-1$.
    AnonSet$_i$ = [ ]
    for $h \leftarrow 0$ to $n-1$ do
        if $f(P_i, P_h) < d$ then
            Add client $c_h$ to AnonSet$_i$.
    return AnonSet$_i$
```

Algorithm 1: TASP learning protocol.

```
**Data**: $n$ AnonSets $\{\text{AnonSet}_0, ..., \text{AnonSet}_{n-1}\}$, round
        time $t_m$ $(m > k-1)$.
**Result**: Anonymity set for each client.
for $i \leftarrow 0$ to $n-1$ do
    for $c_j \in AnonSet_i$ do
        if $c_j$ does not send message between $t_m$ and
        $< t_{m+1}$ and $c_i$ does then
            Remove $c_j$ from AnonSet$_i$
return Flush messages at time $t_{m+1}$
```

Algorithm 2: TASP online protocol.