On the Impact of Cross-Layer Information Leakage on Anonymity in Crowds

Andriy Panchenko
Interdisciplinary Center for Security, Reliability and Trust (SnT)
University of Luxembourg
andriy.panchenko@uni.lu

ABSTRACT

Common anonymizers focus only on a part of the users' personal identification information, namely on the network addresses of the communicating parties. In the light of the entire communication stack, even if the network addresses are perfectly anonymized, information leakage at one of the other layers can completely wipe out the entire effort. No popular anonymization network follows a holistic approach; all neglect the other layers. For example, at the application layer, they neither filter out nor even warn about information that may look innocent to the end-user, though it may be revealing.

Security analysis of anonymizing networks usually also focuses only on a single layer. It has been shown that in theory taking more layers into account may help to enhance attacks. In this paper, we show how innocent-looking application layer data can be practically used to speedup the network-layer attack in the Crowds anonymization system, which is often applied in wireless and mobile networks. To this end, we define two new attacks – the cross-layer and the combined attack – to facilitate the process and show their superiority compared to the earlier *predecessor attack*. The attacks we propose allow not only building extensive user profiles at low cost, but also speeding up traditional network layer attacks, which are targeted at the identification of users' peer partners. Our analysis uncovers the consequences of ignoring the consideration of information that is available to the attacker. Without a holistic approach to analysis, it is not possible to perform a realistic threat assessment.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection*; C.2.3 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Operations—*Public networks*

Q2SWinet'15, November 2–6, 2015, Cancun, Mexico.

ACM 978-1-4503-3757-1/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2815317.2815325.

Keywords

Privacy; Anonymity; Crowds; Predecessor Attack; Cross-Layer Attack

1. INTRODUCTION

Anonymous communication deals with the privacy protection of network addresses for communicating parties. All popular anonymization networks provide defenses only at the network layer. If there is no such an explicit defense, an eavesdropper can directly see the IP addresses of the sender and recipient of any message in the system. Moreover, other meta-information such as time, duration, and volume of the communication is directly visible to an eavesdropper. By tracking communication over time, it is often possible to collect enough evidence to uniquely determine a person's identity, habits, health, profession, and leisure activities.

Due to a massive deployment of anonymization networks over recent decades, it has become possible to analyze their properties "in the wild". Many attacks were developed, targeting not only the network layer, but also the application layer [6] or side channels [9]. A growing number of emerging attacks target layers of the network stack that were not the primary focus of research in the past. In addition to this, the consequences of combining data on different layers and making use of its synergy (cross-layer attack) has not yet been studied in-depth. However, the anonymization networks were not designed to work holistically and network protocols at the application layer were not originally conceptualized in a privacy-friendly manner. Hence, it is possible to infer revealing information about users at the application layer (e.g., in HTTP headers there are browser and OS versions, accepted language tags, etc.) and apply them for traffic analysis with the goal of boosting the speed and accuracy of the attack. Even very common settings, e.g., English as the accepted language at the HTTP header can significantly reduce anonymity at the network layer.

In this manner, application layer profiles can be refined and enriched with the information obtained on the network layer and the other way around as well. Regardless of the layer in the communication stack, if the adversary gets any information, he can also apply this information at other layers in order to reduce anonymity. Nevertheless, he needs to make sure that the same entity is observed on both layers.

It is erroneous to assume that the attacker will not make use of all the information that is available/visible to him. Thus, when assessing the degree of anonymity, it is not acceptable to neglect information from the other layers, such

as the application layer, even if the communicated information looks innocent at first glance.

In this paper we define and evaluate two novel attacks using the cross-layer information leakage and evaluate them practically in the Crowds anonymization system, which is – because of its lightweight protocol – often applied in wireless and mobile networks. We show the superiority of our attacks compared to the earlier predecessor attack. We urge the research community to apply holistic analysis when evaluating attacks, as only then it is possible to assess the real magnitude of the threat.

2. RELATED WORKS

To the best of our knowledge, only a small number of research papers in the area of anonymization networks consider information leakage from the application layer, even though failing to consider this leakage and the impact of synergy effects resulting from joining information from several layers is a serious drawback in the security evaluation of the resulting privacy-preserving techniques. Ignoring some of the available information gives an incorrect perception of the anonymity properties. As our evaluation shows, settings and information that are harmless from a perspective of a single layer may lead to complete deanonymization when mounting an attack on several layers. It is naïve and unacceptable to assume that an attacker will not make use of all the information at his disposal.

Irwin et al. [7] claim that inferences allow users' privacy be compromised, even when powerful anonymization systems are used. To improve the situation, the authors point out the necessity for dynamically customizable privacy policies to protect users' privacy. These policies should depend on sensitivity and identifiability of data that is being transmitted. Irwin et al. suggest that sensitivity should be based on the user's own judgement, while the identifiability of data should be a function of the anonymity set. In particular, the user's judgement about the sensitivity of the data is very subjective, as data that seems to be innocent from a user's perspective may be revealing to an extent that is enough to completely deanonymize him.

Similar results are shown in [13] and [8], where, despite the use of strong cryptographic primitives such as zero-knowledge proofs, users of the system can be profiled and identified.

Clauß and Schiffner [3] suggest a statistical attacker model for anonymity assessment at the application layer. The authors propose to apply it to measure anonymity when publishing data that may include personal identifiable information. Before publishing a set of data, a user can estimate to which extent this data can be applied to decrease his anonymity or even to deanonymize him. The authors distinguish between attributes while dealing with data. They call a subset of attributes either a *profile* or a *partial identity*. The authors distinguish these two terms in the following way: if there is no evident binding of a set of attributes to an entity, it is called a "profile"; if it is known that the originator of the set of attributes is a single entity, it is called a "partial identity". However, the work of Clauß and Schiffner is rather theoretical and too abstract to be directly applied in a practical attack or as a part of an actual attack.

In their other work on structuring anonymity metrics [4], Clauß and Schiffner present models for anonymity metrics for both network and application layers. In addition, the authors highlight the need to join models from both layers in order to provide a usable unified metric for user-centric identity management systems. However, the practical realization of the unified model and corresponding metrics is left for future work.

Diaz et al. [5] were the first to show a counterexample, where the anonymity – contrary to a common belief – increases when the adversary combines information from several layers. In their application scenario, these were the network and the application layers and anonymity was measured in terms of entropy. In our work on this topic [12], we identified and derived a necessary condition for the decrease of anonymity. The availability of such a condition helps users to assess to what extent their behavior on the Internet contributes to the speed-up and increase of accuracy of their deanonymization. However, how exactly the attack works and performs if the condition is met was set aside for future work. The current paper aims to close this gap.

3. BACKGROUND

This section gives a background on the Crowds anonymization system, followed by an introduction to the earlier predecessor attack.

3.1 Crowds

Crowds [14] is a peer-to-peer system for anonymous web browsing. It is based on an idea to provide anonymity by a simple randomized protocol for routing. In this protocoll, all participants forward not only their own messages, but also messages on behalf of other users. Thus, the anonymity is achived by means of an increased path length. This is the price for a trade-off between performance and privacy. The main idea of Crowds is to hide users' communications by routing messages randomly within a group of similar users ("blending into a crowd," where the name comes from). When a user wants anonymously request a website, he first sends his request to a randomly chosen crowd member (called a *jondo*). After receiving such a request, this jondo (and all subsequent jondos, if any) decides whether to forward the message to its final destination or to another randomly chosen jondo in the crowd. Practically, this is realized by performing a *biased coin toss*. More formally, the message is forwarded to another randomly-chosen network participant with probability p_f, or to its final destination with probability $1 - p_f$. p_f – the forwarding probability – is a system parameter. Communication within the crowd, i.e., between jondos is encrypted at the link layer. This implies that, by design, each of the jondos on the path sees the content of messages passing by, including the application layer data and addresses of final destinations. The ultimate request to the server is usually sent in plaintext. Though the system is not new itself, many approaches build on this idea to provide lightweight anonymization (e.g., in mobile settings [2, 16]).

3.2 Predecessor Attack in Crowds

Because the initiator of a communication always forwards messages to a randomly-chosen node, but all subsequent nodes do so only with a certain probability that is smaller than one, there is information leakage in the system. Indeed, the one who forwards a message to a colluding jondo is more likely to be the message originator than any other non-colluding jondo (note that there is no simple strategy

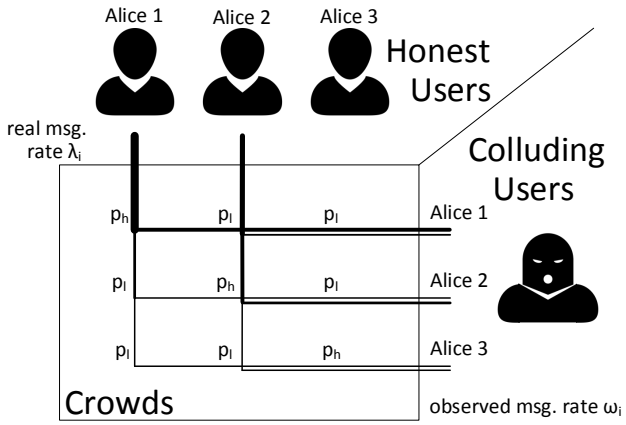


Figure 1: Information flow for the attacker: matrix M

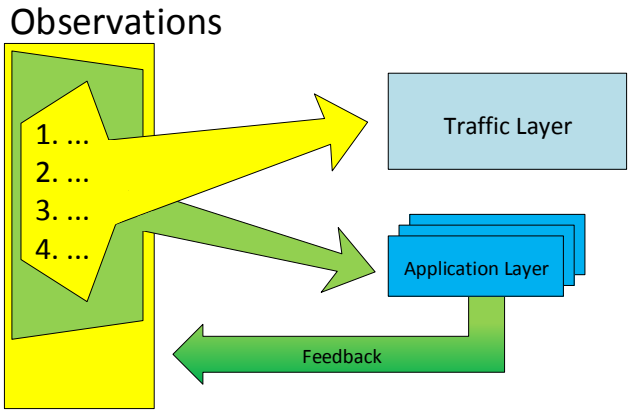


Figure 2: Cross-layer information flow

to improve the situation by, e.g., letting the message originator to act in the same way as any other jondo receiving a message: in this case the information leakage will be directed to the destination of a message). In the following we will show how to exploit the above mentioned information leakage to reveal communication peers of users in Crowds. In the literature, this is known as the predecessor attack [17, 15, 11].

Let $n+c$ be the size of the crowd (anonymization system), where c is the number of colluding jondos, n the number of honest jondos, and p_f the probability of message forwarding within the system (for more detailed definition, we refer to [14]). $n+c$ and p_f are public system parameters known to everyone, while c is only known to the adversary (colluding nodes).

Furthermore, let p_h denote the probability that a colluding jondo receives a message directly from path initiator[1]. Let p_l be the probability for one honest jondo, which is not the message originator, to forward the message to a colluding jondo (note that $p_h + (n-1) \cdot p_l = 1$). The interested reader is referred to [11] for details of how to calculate the probabilities p_h and p_l.

Because of the information leakage mentioned above, $p_h > p_l$ for all admissible parameter values [11]. This fact can be exploited by an attacker to make precise statements about the users' peers. We assume that the attacker contributes c jondos ($c \geq 1$) to the crowd. Without loss of generality, we further assume that the attacker is only interested in deanonymizing connections to a single external entity, which we call *Bob*. In the following, based on our previous work on this topic, we will summarize our approach on how to not only find out whether there is a communication taking place between a user and Bob, but also to estimate the communication rates that each crowds user has with Bob. This estimation can be performed with an arbitrary precision [11].

From the point of view of an attacker, there are n jondos that are communicating with Bob, each with its own average sending rate $\lambda_i \geq 0$ per time interval. We model these as Poisson processes $A_i = \mathcal{P}_{\lambda_i}$ for $i \in 1 \ldots n$. In our opinion, the arrival distribution in a form of Poisson processes is a decent trade-off between realism and analytical complexity. In this way, we can provide an initial approximation that can be further refined by modelling arrivals in a more sophisticated but also complex manner. The colluding entities (attacker) observe on average the following number of messages from the i-th system member and destined to Bob per one time interval:

$$E[\text{msgs to Bob from } i] = p_h \cdot \lambda_i + \frac{1-p_h}{n-1} \cdot \sum_{j=1, j\neq i}^{n} \lambda_j \quad (1)$$

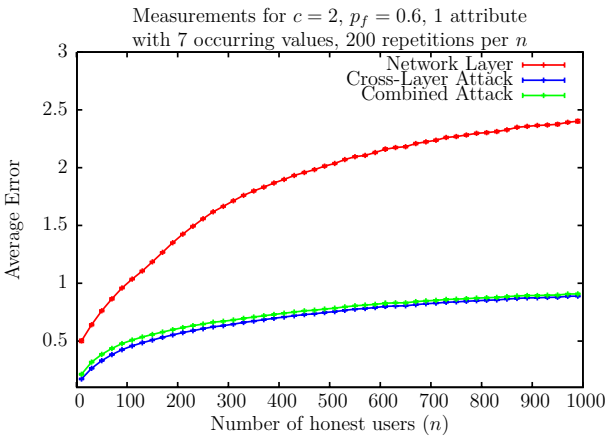
By means of the following matrix M, we can model and estimate the number of messages O_i arriving at the attacker from jondo i, that are directed to Bob:

$$M = \begin{pmatrix} p_h & p_l & \cdots & p_l \\ p_l & p_h & \cdots & p_l \\ \vdots & \vdots & & \vdots \\ p_l & p_l & \cdots & p_h \end{pmatrix} \quad (2)$$

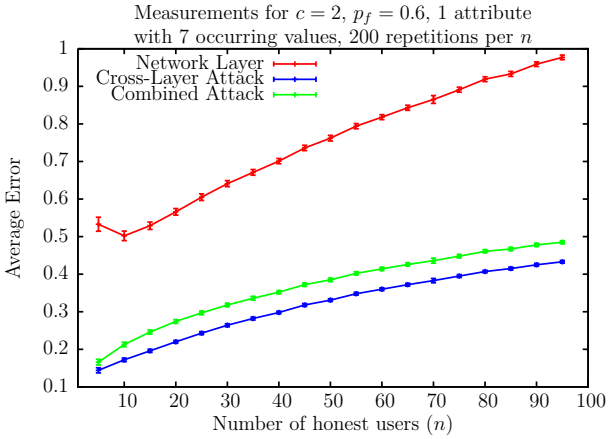
$$O_i = \sum_{j=1}^{n} \mathcal{P}_{m_{i,j}\lambda_j} \quad (3)$$

These observations can be considered as a vector of messages (where each element represents the number of messages received from the corresponding jondo by colluding members and addressed to Bob). Each element is a product of the vector of messages actually sent $(\lambda_1, \ldots, \lambda_n)$ with the matrix M (see Figure 1). The thickness of the lines represents the sending rate of the corresponding user (the thicker the line, the higher the sending rate of the user). Please note that the nonexisting curve from one of the jondos on the right-hand side is not a mistake, but simply means that the user does not send their own messages. Instead, this user only forwards some messages on behalf of the others (this is represented by the thin line pointing to colluding users).

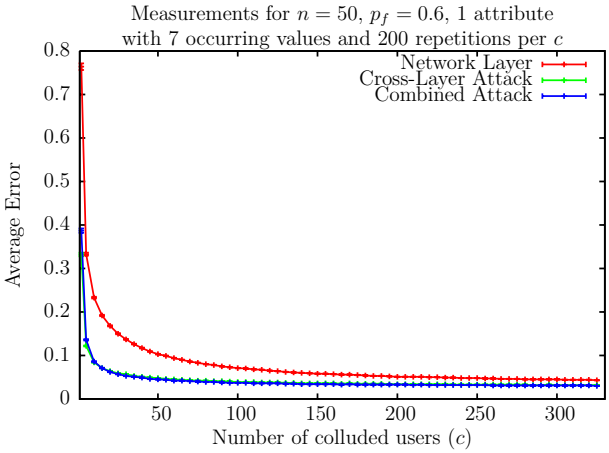
[1] h stands for "high", l for "low"



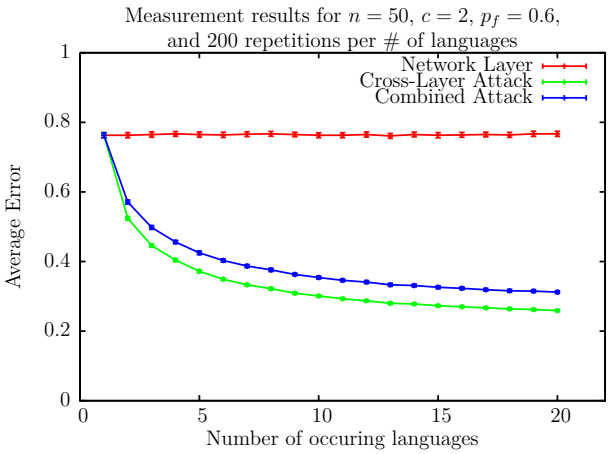
(a) Increasing no. of honest users



(b) Increasing no. of honest users (zoomed in)



(c) Increasing no. of colluding users



(d) Different no. of application layer attributes

Figure 3: Cross-layer attack: simulation results

Because O_i is the sum of Poisson processes, on its own it is also a Poisson process with the following characteristics:

$$O_i = \mathcal{P}_{\omega_i} \quad \text{for} \quad \omega_i = \sum_{j=1}^{n} m_{i,j}\lambda_j \tag{4}$$

$$E[O_i] = V[O_i] = \omega_i \tag{5}$$

Because the cardinality of ω_i depends on the values of the λ_i, from the observations of O_i it is possible to draw conclusions about the λ_i (for $t \to \infty$):

$$(\omega_1, \ldots, \omega_n) = M(\lambda_1, \ldots, \lambda_n) \tag{6}$$

$$\iff \quad (\lambda_1, \ldots, \lambda_n) = M^{-1}(\omega_1, \ldots, \omega_n) \tag{7}$$

Recall that Crowds is susceptible to the information leakage because of different forwarding probabilities for message originator vs. any other node ($p_h \neq p_l$). Hence, the matrix M is invertible to M^{-1} and, thus:

$$M^{-1} = \begin{pmatrix} \tilde{p_h} & \tilde{p_l} & \cdots & \tilde{p_l} \\ \tilde{p_l} & \tilde{p_h} & \cdots & \tilde{p_l} \\ \vdots & \vdots & & \vdots \\ \tilde{p_l} & \tilde{p_l} & \cdots & \tilde{p_h} \end{pmatrix} \tag{8}$$

$$\tilde{p_h} = \frac{n + p_h - 2}{np_h - 1} \tag{9}$$

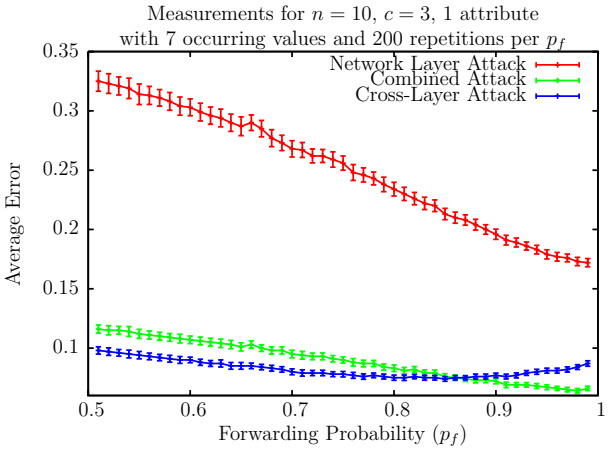
$$\tilde{p_l} = \frac{p_h - 1}{np_h - 1} \tag{10}$$

Under the common values for p_f, n, and c ($0 < p_f < 1$, $n \gg c$) the following inequalities hold: $\tilde{p_h} > 0$ and $\tilde{p_l} < 0$.

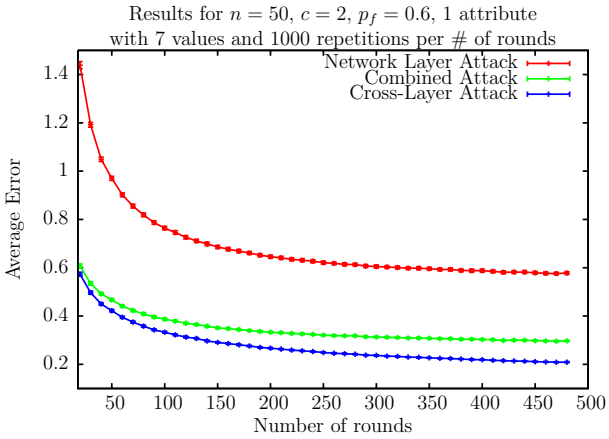
Having observed ω_i, it is therefore possible to establish an estimation for the λ_i, namely $\tilde{\lambda_i}$, as follows:

$$\tilde{\lambda_i} = \sum_{j=1}^{n} {m^{-1}}_{i,j}\omega_j \tag{11}$$

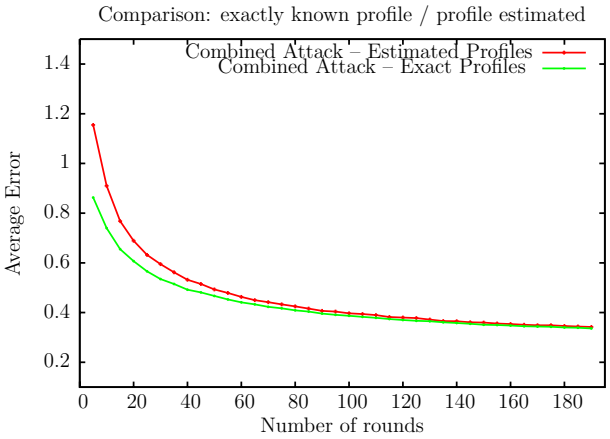
Now it is possible to calculate the number of observations needed in order to estimate sending rates λ_i with any specified precision. This can be done by applying a Chernoff-style bound for the probability that a Poisson process deviates from its mean. For further details, we refer the interested reader to [11].



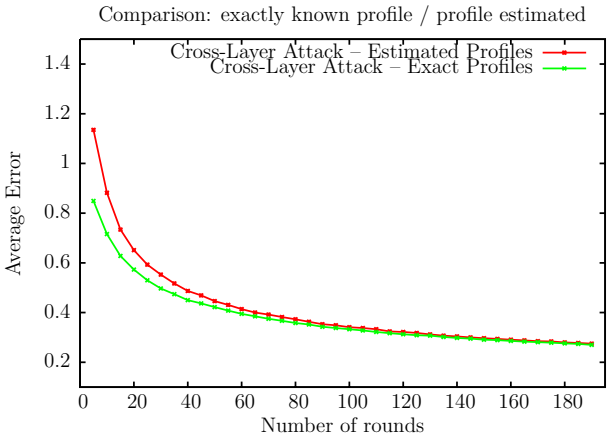
(a) Different values of forwarding probability



(b) Increasing no. of observation rounds



(c) Influence of inaccuracy in profile estimation (combined attack)



(d) Influence of inaccuracy in profile estimation (cross-layer attack)

Figure 4: Cross-layer attack: simulation results (cont'd)

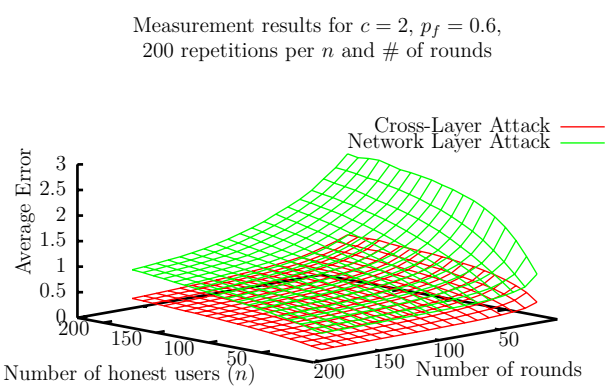
4. NETWORK LAYER ATTACK SPEED-UP

The classic predecessor attack attempts to identify users' peer partners at the network layer [17, 15, 11]. In this section, we illustrate that finding a user's peer partners (deanonymization) by the means of this attack can be significantly sped up. Thus, the previously published results of the attack give an incorrect perception of the quality of protection. In our approach, we build an extensive user profile at the application layer in parallel with the network layer attack, i.e., we enrich data at the network layer by identifying the values of different communication attributes from the application layer.

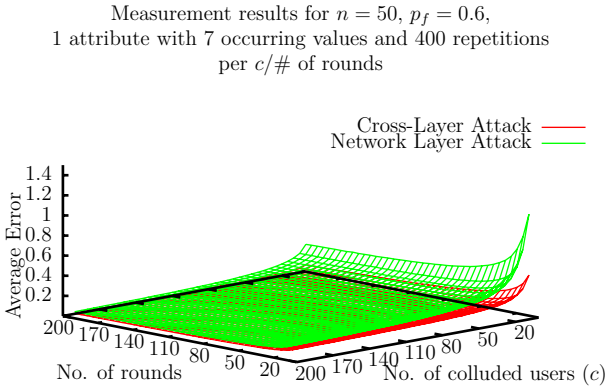
The following observation is responsible for a significant increase of the attack's speed: while a user typically has many different communication peers, the attributes of his application layer profile (the version of the Internet browser, the set of accepted languages, etc.) usually remain the same. In our previous work on this topic, we have shown that the needed number of observations to identify and confirm user A_i's peer partner B with arbitrary precision depends proportionally on the message rate from A_i to B observed by colluding users [11, 10]. The same is valid for the application layer profile. Hence, the attacker will be able to identify the application layer profile of his victim considerably faster than the communication profile. If the attacker is making use a statistical analysis on the network layer, he can then use information from the application layer to decide whether the data belongs to the same communicating peer on the network layer attack or not, i.e., by filtering out improbable combinations of attributes or, in general, messages that do not match the victim's profile.

Figure 2 demonstrates how the attack works in practice: firstly, a user's profile at the application layer is built performing the predecessor attack on the application layer data. This profile (information about the attributes at the application layer) is then used to refine and speed up the predecessor attack on the network layer. The feedback from application to network layer allows the observations from the network layer to be accepted or rejected (depending on whether they match the application layer profile or not). These "refined" observations are further used as an input to the classic predecessor attack on the network layer.

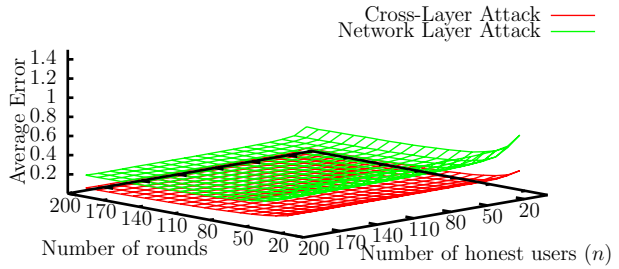
Hence, after the profiles of users have been built, it is possible to use the same observations/data in order to identify a user's peer partners. In fact, even during the process of discovering and building the profile from the application layer,



(a) Different no. of rounds/honest users. Two colluding users



(b) Different no. of rounds/colluding users



(c) Different no. of rounds/honest users. 20% colluding users

Figure 5: Cross-layer attack: simulation results (3D view)

the calculations for the attack on the network layer can be adapted on the fly. We propose and evaluate two novel attacks using application layer data that work as follows:

- *combined attack:* we mount the classic predecessor attack (described above) only for messages matching the attributes of the application layer profile;
- *cross-layer attack:* same as above, but additionally p_h^{new} and p_l^{new} are used instead of p_h and p_l. p_h^{new} is the probability that A_i is the originator of the message, given that the message is received from A_i by an attacker and that it has a corresponding application layer attribute (how to calculate this probability is described in detail in our previous work on this topic [12]). Similarly, $p_l^{new} = \frac{1-p_h^{new}}{n-1}$.

In the next section, we compare the effectiveness and precision of these attacks with the predecessor attack, which is applied at the network layer only.

5. EVALUATION

In this section we evaluate the effectiveness of the two newly-introduced attacks. To do this, we performed simulations. Our simulations were performed as follows: we assume that every honest user has exactly two random communication peers. These peers remain the same over the whole simulation run. In each round of communication, every honest user sends between zero and three messages to each of his communication peers.

While Figure 3 shows the results for different numbers of honest users, colluding users, and application layer attributes, in Figure 4 we present the evaluation for different values of the forwarding probability, a growing number of observation rounds, as well as the influence of the inaccuracy in the profile estimation. Additionally, Figure 5 presents the results in a 3D view.

Figure 3 illustrates that both of the novel attacks are significantly more precise and, hence, faster, than the original attack on the network layer. The lower the fraction of colluding nodes in the system, the greater is the advantage of our attacks. In the evaluated scenario the accuracy is improved by a factor of three. An interesting finding is about the superiority between the attacks. As we can see, up to a forwarding probability value of 0.86, the cross-layer attack is more precise than the combined. However, for $p_f \geq 0.86$ the combined attack is more accurate. Since p_f is a publicly-known system parameter, the attacker can select his strategy according to the parameter values. This behavior of the proposed attacks can can be seen in Figure 4(a).

Figure 4(d) shows, among other things, the influence of inaccuracy in profile estimation on the results of the novel attacks. While for one curve the actual profile was used, for the other one the profile was at first estimated (using a

similar method as for the network layer attack). Afterwards, the estimated profile was used for combined and cross-layer attacks. The results show that, with a growing number of observations, there is no significant difference, whether the real or estimated profile is used. A reason for this is that the inaccuracy in profile estimation decreases with the growing number of observation rounds.

The novel attacks are faster since the application layer profile is usually much more stable than the communication profile. However, the analytical quantification of the speed gain is beyond the scope of this paper and will be the subject of future research. The same applies for the practical confirmation of the attack and its realization in real-world settings. We do, however, briefly elaborate on strategies that can be used to mitigate our novel attacks.

The attack can be impeded if a filtering system is in place. Such a filtering system should target the application layer to substitute the identifying information from the web browser. One possibility would be to frequently make random changes to the browser data. However, if not all users of the system apply such a strategy, there is a possibility that an attacker could link such frequent random changes. A better approach for substitution would be to use a known statistical distribution of the browser data such as [1]. To improve the circumvention and become even more unidentifiable, we propose the following strategy: a user-side toolbox should observe the communication of the others in the network in the same way that an attacker does. This information should not be logged, but only used to collect statistics about browser information of users within the anonymization network. This is because the distribution of identifying information among users of anonymizing systems may be different from those on the Internet in general. Another way to improve privacy would be to decrease the sending rates of communication with some specific profile to a level, where insufficient information leakage exists for an attacker to mount a successful attack with a high confidence. However, this approach still requires passive network observation to collect browser statistics in order to identify tolerable communication rates.

6. CONCLUSIONS

In this paper we shown that enriching network layer information with innocent-looking application layer data (e.g., browser strings such as accepted languages in HTTP requests) can be used to significantly increase accuracy and speed up traditional attacks targeted at the network layer only. To this end, we applied cross-layer information leakage to boost performance of the earlier predecessor attack in Crowds. It is naïve to assume that an adversary would not make use of all the information which is at his disposal to perform a more accurate at faster attack. Therefore, information leakage at all layers has to be taken into account in future research in order to provide realistic and, thus, usable metrics for anonymity.

Regarding the usage of the innocent-looking browser information at the application layer, it is possible to circumvent the attack speed-up. To this end, we proposed a filtering system that would dynamically replace the identifying information of the browser with the browser data according to the intrinsic statistical distribution in the considered anonymization network. The network layer attack itself, however, cannot be circumvented completely. Crowds leaks routing information under any admissible parameter values [11]. The number of observations needed to determine the sending rate of the users in the crowd precisely enough can be relatively small. Hence, users with a need for a strong anonymity in an open environment should consider using alternative anonymization techniques.

Acknowledgements

This work has been partially supported by the EU Horizon 2020 framework programme within the Privacy Flag project (reference: 653426). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the European Commission.

7. REFERENCES

[1] Browser statistics. http://www.w3schools.com/browsers/browsers_stats.asp, September 2014.

[2] C. Andersson, R. Lundin, and S. Fischer-Hübner. Privacy-enhanced wap browsing with mcrowds - anonymity properties and performance evaluation of the mcrowds system. In *Proceedings of the Fourth annual ISSA 2004 IT Security Conference*, Johannesburg, South Africa, July 2004.

[3] S. Clauß and S. Schiffner. Anonymität auf Anwendungsebene. In *Proceedings of Sicherheit 2006*, pages 171–182, Bonn, Germany, 2006. GI Lecture Notes in Informatics P-77.

[4] S. Clauß and S. Schiffner. Structuring anonymity metrics. In *DIM '06: Proceedings of the Second ACM Workshop on Digital Identity Management*, pages 55–62, New York, NY, USA, 2006. ACM Press.

[5] C. Diaz, C. Troncoso, and G. Danezis. Does additional information always reduce anonymity? In *WPES '07: Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society*, pages 72–75, New York, NY, USA, 2007. ACM.

[6] FortConsult. Practical onion hacking: Finding the real address of Tor clients. October 2006.

[7] K. Irwin and T. Yu. An identifiability-based access control model for privacy protection in open systems. In *WPES '04: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 43–43, New York, NY, USA, 2004. ACM Press.

[8] D. Kesdogan, V. Pham, and L. Pimenidis. Information disclosure in identity management. In *Proceedings of 12th Nordic Workshop on IT-Security, NordSec*, Reykjavik, Iceland, Oct 2007.

[9] S. J. Murdoch. Hot or not: Revealing hidden services by their clock skew. In *13th ACM Conference on Computer and Communications Security (ACM CCS)*, Alexandria, Virginia, USA, 2006.

[10] A. Panchenko and L. Pimenidis. Fundamental Limits of Anonymity Provided by Crowds. In *8th International Symposium on System and Information Security SSI'2006*, Sao Jose dos Campos, Sao Paulo, Brazil, 8–10 November 2006.

[11] A. Panchenko and L. Pimenidis. Crowds Revisited: Practically Effective Predecessor Attack. In *Proceedings of the 12th Nordic Workshop on Secure IT-Systems (NordSec 2007)*, Reykjavik, Iceland, October 2007.

[12] A. Panchenko and L. Pimenidis. Cross-Layer Attack on Anonymizing Networks. In *Proceedings of the 15th International Conference on Telecommunications (ICT 2008)*, St. Petersburg, Russia, June 2008. IEEE Xplore.

[13] A. Pashalidis and B. Meyer. Linking Anonymous Transactions: The Consistent View Attack. In G. Danezis and P. Golle, editors, *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, Cambridge, UK, June 2006. Springer.

[14] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, pages 66 – 92, April 1998.

[15] V. Shmatikov. Probabilistic model checking of an anonymity system. *Journal of Computer Security*, 12(3-4):355–377, 2004.

[16] A. Vaha-Sipila and T. Virtanen. BT-Crowds: Crowds-Style Anonymity with Bluetooth and Java. In *HICSS '05: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9*, page 320.1, Washington, DC, USA, 2005. IEEE Computer Society.

[17] M. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. In *ACM Transactions on Information and System Security TISSEC'04*, volume 7 (4), pages 489 – 522. ACM Press, November 2004.