Anonymity Networks - Onion Routing and Alternatives

Student Paper for *184.269 VU Advanced Internet Computing*, WS2014

Clemens Heller, Georg Haaser, Martin Imre, Jasmin Thöner
Vienna University of Technology
1040 Vienna, Austria
E-mail: {e0953347, e0801709, e0853761, e1125020}@student.tuwien.ac.at

***Abstract*—Recent scandals and attacks have made it clear, that keeping our privacy is important. One approach to achieve privacy is to stay anonymous on the Internet. Therefore, several technologies like anonymity networks have been invented. This paper gives an overview of several different anonymity networks. It especially discusses onion routing with it's most popular implementation: Tor. Since Cloud Computing has become very popular in recent years, approaches to enhance anonymity in Cloud Computing are also being reviewed.**

***Keywords*—Anonymity networks, Low-latency systems, PET, Onion Routing, Cloud Computing**

I. Introduction

Nowadays, the intensive use of the Internet and thriving technologies like Cloud Computing make it necessary for us to be concerned about our privacy. Online banking, social networking and other applications use sensitive data and letting this data get into the wrong hands can cause you much harm. The argument that someone who wants privacy must have something to hide does not hold anymore. To give an example: Organizations can use your personal data to show you advertisements that may be so subtle, you don't even know they are personalized. For example, they could take photos of your best friends and blend them together to use the resulting image in an advertisement. You will not recognize your friends in the person that advertisement the advertisement shows anymore, but you will definitely be more likely to buy the advertised product[1]. But also criminals could use your personal information to trick you into giving them e.g. bank account data or passwords. There are certain attacks, like Context-aware Spam [2] or Social Phishing [6], that use data from social networks to make phishing or spam mails more authentic. Those methods are significantly more successful than methods that don't use personal data. Such techniques as well as the recent NSA scandal and other incidents make it clear, that privacy is important for all of us. Also, certain applications like e-voting or witness protection would not have become possible without having privacy enhancing technologies (PET).

Before discussing PET, some terms concerning privacy need do be defined. *Privacy* refers to the right of an individual or a group to decide which information about themselves can be shared with others and which should not. *PET* describe certain ICT measures to protect one's privacy (mostly by hiding or eliminating personal data) without losing functionality of the services one wants to use [1]. *Anonymity* means that actions cannot be collated with a person. Someone can only be anonymous within an *anonymity set*, that is a set of subjects with potentially the same attributes as the anonymous person. The anonymity set of a sender and that of a receiver of an action may or may not overlap, and it might vary over time [11]. *Unlinkability* means that multiple actions of a user cannot be linked together. *Unobservability* means that the states of communicating parties are not distinguishable, e.g., it is not possible to decide if a message is a real one or just random noise, or if someone is a sender, a receiver or not participating in the communication at all.

In the past, many different solutions for protecting a user's privacy and/or anonymity have been proposed. They basically can be divided into two groups: high-latency systems and low-latency systems. High-latency systems are designed for message-based communications like e-mail or Usenet postings, where it is okay to have slow transmission speed. Examples of such systems are e.g. Remailer like Mixminion [4] or the original Mix-nets proposed by Chaum (Chaum Mixes) [3]. The transmission might take some hours or even days, which makes high-latency systems not suitable for most Internet applications (e.g. HTTP, SSH,...). Because of this fact, we will deal only with low-latency systems, which provide anonymity with only a minimum of performance loss.

This paper shines light on the current research on low-latency anonymity networks with an emphasis on the currently most used technique: onion routing. Section II describes onion routing and points out its benefits and pitfalls. It also discusses Tor, the most popular and wide used implementation of onion routing. Section III will then offer insight into other anonymity network techniques. Finally, Section IV will rise the question of how to provide anonymity in the Cloud and discuss some solutions that have been proposed so far.

II. Onion Routing

Onion routing is a mixed-based system. Mixes or mix-nets were proposed by Chaum in 1981 and originally were a high-latency system. The basic idea is to use several intermediate relay servers (mixes) between a sender and receiver that mix and forward packets so that communications paths cannot be retraced [11].

Onion routing is used to anonymize TCP-based communications and uses the principle of Chaum mixes [11]. There are several relay servers (onion routers, OR) chained between the

[1] http://www.ted.com/talks/alessandro_acquisti_why_privacy_matters (last access: 19.12.2014)

sender and receiver, each of them owning a public/private key pair. If the sender wants to send a message to the receiver, he wraps the payload with several layers of encryption: He first encrypts the message with the public key of the last OR in the chain, then he encrypts the resulting ciphertext with the public key of the OR before the last server and so on. The result is a so-called onion (because the layers of encryption are wrapped around the payload like the layers of an onion) that is sent to the first OR in the chain. This OR decrypts the received message with its private key and forwards it to the next OR which will also decrypt the resulting message with his private key and so on. The last OR then forwards the cleartext message to the receiver.

Onion routing has many advantages. It can be used to provide anonymity for a variety of services (it is application independent) and it also provides real-time and bidirectional communications. Also, the client (sender) itself chooses the path through which its messages should be sent and the ORs only know their immediate predecessor and successor. Especially, intermediary ORs do not know the origin (except the first OR, called entry node) as well as the destination and content (except the last OR, called exit node). In order to make it impossible for an adversary performing traffic analysis to recognize the route by inspecting the length of the message (due to the decryption, the message would become shorter on every hop), a random bit string that has the same size as the peeled off encryption layer is appended before forwarding [11]. Onion routing provides strong unlinkabiliy, because no single OR knows everything about the route and routes are changing for every message that is being sent.

On the other hand, onion routing also has its pitfalls. It cannot provide perfect sender or receiver anonymity, since it is possible for a local eavesdropper to see all the incoming and outgoing messages of an OR. If an adversary has enough ORs under control, he could retrace the routes the messages are undergoing. Therefore, privacy in onion routing is determined by the relationship of the number of participating ORs vs. the number of compromised ORs [11].

1) The Tor Network: The Tor project[2] was called into life to eliminate the problems of the classical onion routing approach. Tor is the most popular implementation of onion routing, but it is more than that. Tor is an overlay network that uses regular network architecture by volunteers providing servers that act as relay nodes or directory nodes. A communication path in Tor consists of three relay nodes, which are known by directory servers. A client can obtain information about relay nodes from the directory server and then build a path to send a message to the receiver. Tor provides perfect forward secrecy, which means that the client can connect to services without revealing his location to those sites or an observer [11]. To provide anonymity to service providers, Tor also includes what is called hidden services: A hidden service can be accessed only through the Tor network via special onion URLs that consist of random characters (.onion).

Tor is a distributed-trust system, which means that an adversary compromising a single relay node would not cause much harm but that Tor needs to prevent an attacker from owning too many compromised nodes. It protects against traffic analysis attacks by non-global adversaries (such that only compromised a single or just a few nodes) [9], but it cannot provide anonymity in presence of a passive global observer [11]. Such an observer could monitor the whole (Tor) network traffic and therefore would know which message was sent by whom and to whom it was sent.

[2]https://www.torproject.org/ (last access: 20.12.2014)

III. Other Low-Latency Systems

In the following, we give an overview of other mixnet-based schemes, DC-net systems, network routing-based schemes and peer-to-peer communication systems.

A. Other Mixnet-based Schemes

1) Garlic Routing: Garlic routing also uses the basic idea of onion routing and forms the basis of the Invisible Internet Project (I2P)[3]. In addition to usual onion routing, it ties multiple messages together into a so-called garlic clove. Those bundle of messages is then encrypted together for transfer to the relay nodes.

2) Web MIXes: WebMIXes consist of three parts: the JonDonym anonymous proxy (JAP)[4] on the sender side, mixes between sender and receiver, and a cache-proxy on the receiver side. In contrast to onion or garlic routing, in WebMIXes, cover traffic is used. This means, that all senders constantly send something (either a real message or some random dummy traffic that is indistinguishable from real encrypted messages) and all receivers constantly receive something [11]. This is done in order to make it harder for an adversary to perform traffic analysis.

B. DC-net Systems

DC-nets have also been proposed by Chaum and they use a secure multi-party computation protocol. The benefit is that DC-nets provide provable sender an receiver anonymity and do not rely on a third-party. Another benefit over mix-based systems is non-interactivity: There does not need to be a direct party to party communication, a sender can publish its message in a single broadcast round. The requirement for DC-nets is to have a reliable broadcast network, which is often not suitable (e.g. Internet). There are some other problems too, like transmission collision or the detection of dishonest parties. An example of a DC-net is Herbivore [5].

C. Network routing-based Schemes

A popular representative of network routing-based schemes is Crowds. Its purpose is to secure against internal attackers or a corrupt receiver [11]. First, the user has to join a crowd (i.e. a set of similar communication partners). Then the messages of this user are routed randomly within the group. Each member of the group can decide if it wants to pass the message directly to the receiver or if it forwards it to another random member. It is more likely ($p > 0.5$) that a member will forward the message to another random member. An adversary impersonating a group member cannot tell whether someone is the real sender or is just forwarding the message, since the group members are indistinguishable. But again, also Crowds

[3]https://geti2p.net/en/ (last access: 20.12.2014)

[4]http://anon.inf.tu-dresden.de/index_en.html (last access: 20.12.2014)

cannot provide anonymity in case of a global attacker. It can also be compromised with a predecessor attack.

D. Peer-to-Peer Communication Systems

An example of a P2P communication system is MorphMix. In MorphMix, the intermediate nodes choose the communication path, not the initiator itself. Intermediary nodes are application-level mixes which use TCP for communication. Since it is a P2p system, each participant is also a mix. Any participant can join or leave at any time. As in Crowds, a participant doesn't know if its predecessor is the sender of a message or just a relay. Therefore, plausible deniability is provided. MorphMix is vulnerable to colluding attacks [11].

IV. Cloud-based Solutions

Cloud Computing has become very popular in the past view years. When it comes to privacy and anonymity, Cloud Computing becomes very complex and most people would not attribute those two characteristics to a typical Cloud. We have to distinguish between two target groups that may want to stay anonymous: The users of Cloud services as well as the providers of services. The big challenge with service provider anonymity is that the service provider must authenticate itself and that the usage of Cloud resources must be accounted and charged for at an on demand basis [10]. This may seem like a contradiction to the anonymity concept, but several solutions have been proposed that make it possible for a service provider to stay anonymous in the eyes of the Cloud resources provider.

One solution was proposed by Pacheco and Puttini [10]. They introduced a broker entity (called Third Party Broker, TPB) that intermediates between the Cloud resource provider and der service provider. The TPB provides the service provider with anonymous credentials to be used for authenticating to the Cloud provider. It also handles contracting, accounting and billing. Anonymity in [10] is achieved through a multi-layer design in order to hide details about the service provider's ID, personal data, behavior and location. Contracts (and including personal data) are anonymized by introducing indirect contracts between the TPB and the service provider and between the TPB and the Cloud provider respectively. Only the TPB (which should of course be administrated by a third party and not the Cloud resource provider itself) knows about the service provider's personal data. On the other hand, the TPB is only used for administrative purposes and has no knowledge about service consumption messages sent between the Cloud provider and the service provider. The Cloud provider sends resource usage information along with the anonymous credentials to the TPB, so that the TPB can handle accounting and billing. Those anonymous credentials can either be traceable or non-traceable. While non-traceable credentials provide more privacy, they must be used together with pre-payment or credit systems, which limits flexibility. Traceable credentials provide less privacy (the service provider could be identified, e.g. in case of misbehavior) but can be used with pay on demand systems. With regards to data anonymity (e.g. the messages being sent to Cloud servers and between them) it can be very challenging to provide anonymity, since there are often many different services with different data structures and business logic involved. In this case, each service needs to be analyzed and handled accordingly. Of course, onion routing solutions like Tor or any other anonymity network described before can also be included to provide sender and receiver anonymity between communication partners in the Cloud. A problem with the approach proposed in [10] is that the TPB must be trusted, in fact both by the service provider as well as by the Cloud resource provider. Therefore, the TPB is a single point of failure.

To provide a user's anonymity to the computing nodes in a Cloud, Khan and Hamlen proposed a similar concept as the TPB together with an onion routing approach using Tor[8]. The idea is to use Tor inside the Cloud by utilizing so-called slave nodes (SN) as Tor relay servers that forward a user's message to a dedicated master node (MN) which will itself forward the message to the Cloud nodes that do the calculation on the data. The length of the Tor path can be varied to find the best balance between performance and the required degree of privacy. As with the approach from [10], the MN is used for accounting and billing, except that the method of relating service usage to a certain user is different. Instead of using anonymous credentials, a public-key-cryptography-based approach is used. Of course, MN is also a single point of failure and needs to be trusted.

A third proposal which eliminates the need to trust a single entity can be used for both Cloud services as well as usual anonymous surfing and was provided by Jones et al. [7]. It can be seen as an enhancement to Tor. One drawback of Tor is that performance is limited because relay servers often have limited bandwidth. Another disadvantage is that because Tor relay nodes are publicly known (due to the directory nodes), they can easily be blocked by e.g. governments to enforce censorship (e.g. China does this). To eliminate those two pitfalls Jones et al. created a Tor-like service that is spread across different Clouds. The advantage is that this service is hosted by several anonymization providers that use several Cloud hosting providers. Therefore, one has not to trust a single entity but the trust is divided and it is more easy to circumvent malicious entities. Additionally, since Cloud resources can be added on demand, performance issues due to bandwidth shortage can easily be dealt with. Blocking of nodes also becomes hard, because if a node is blocked, it is easy to spawn a new one with different IPs. Blocking an IP range that the Cloud hosting service uses is also not sufficient, because one would also block other services hosted on this Cloud [7]. One problem that still remains is the fundamental bootstrapping problem (meaning that the start of usage of such a service cannot be anonymized).

V. Conclusion

Privacy is not a topic that only journalists, people living in a regime that enforces cencorship or criminals have to be worried about. Keeping our privacy on the Internet is very important for everyone of us because of various attacks like Social Phishing [6] or Context-aware Spam [2]. Anonymity helps to enhance privacy. In the past, research has developed many different types of privacy enhancing technologies, amongst them different types of anonymity networks. The most popular approach nowadays is onion routing, with its most popular implementation, Tor, that is used by millions of users. While onion routing is a mixnet-based approach, there are other anonymity networks like DC-nets, network-routing

based approaches or peer-to-peer systems that all use different techniques to provide anonymity to senders and receivers of messages transferred on the Internet. Which kind of anonymity network to use depends on the use-case.

Cloud Computing has become very popular in the past few years. Some research has been done about how to achieve anonymity in these very complex systems. Not only the end user wants to be anonymous, but also the service provider that uses Cloud resources. Since such a provider must authenticate himself to the Cloud hosting provider and every usage of a resource needs to be accounted and billed, introducing an anonymity service is a very difficult and complex task. Current anonymity systems for Clouds often have a limitation in that they have a single entity that is responsible for enforcing anonymity and therefore needs to be fully trusted. Further research should focus on how to distribute this trust on various different entities so that no single point of failure exists.

REFERENCES

[1] G. W. van Blarkom, J. J. Borking, and J. G. E. Olk, "Handbook of Privacy and Privacy-Enhancing Technologies: The case of Intelligent Software Agents," Privacy Incorporated Software Agent Consortium, Den Haag, Tech. Rep., 2003.

[2] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders, "Social networks and context-aware spam," in *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '08, San Diego, CA, USA: ACM, 2008, pp. 403–412, ISBN: 978-1-60558-007-4. DOI: 10.1145/1460563.1460628. [Online]. Available: http://doi.acm.org/10.1145/1460563.1460628.

[3] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981, ISSN: 0001-0782. DOI: 10.1145/358549.358563. [Online]. Available: http://doi.acm.org/10.1145/358549.358563.

[4] G. Danezis, R. Dingledine, D. Hopwood, and N. Mathewson, "Mixminion: design of a type iii anonymous remailer protocol," in *In Proceedings of the 2003 IEEE Symposium on Security and Privacy*, 2003, pp. 2–15.

[5] S. Goel, M. Robson, M. Polte, and E. G. Sirer, *Herbivore: a scalable and efficient protocol for anonymous communication*, Ithaca, NY, 2003. [Online]. Available: http://ecommons.cornell.edu/handle/1813/5606.

[6] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Commun. ACM*, vol. 50, no. 10, pp. 94–100, Oct. 2007, ISSN: 0001-0782. DOI: 10.1145/1290958.1290968. [Online]. Available: http://doi.acm.org/10.1145/1290958.1290968.

[7] N. Jones, M. Arye, J. Cesareo, and M. J. Freedman, "Hiding amongst the clouds: a proposal for cloud-based onion routing," in *In FOCI*, 2011.

[8] S. M. Khan and K. W. Hamlen, "AnonymousCloud: a data ownership privacy provider framework in cloud computing," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Liverpool, UK, 2012, pp. 170–176.

[9] D. Mccoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker, "Shining light in dark places: understanding the tor network," in *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies*, ser. PETS '08, Leuven, Belgium: Springer-Verlag, 2008, pp. 63–76, ISBN: 978-3-540-70629-8. DOI: 10.1007/978-3-540-70630-4_5. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-70630-4_5.

[10] V. M. Pacheco and R. S. Puttini, "Saas anonymous cloud service consumption structure," in *32nd International Conference on Distributed Computing Systems Workshops (ICDCS 2012 Workshops), Macau, China, June 18-21, 2012*, 2012, pp. 491–499. DOI: 10.1109/ICDCSW.2012.28. [Online]. Available: http://dx.doi.org/10.1109/ICDCSW.2012.28.

[11] J. Ren and J. Wu, "Survey on anonymous communications in computer networks," *Comput. Commun.*, vol. 33, no. 4, pp. 420–431, Mar. 2010, ISSN: 0140-3664. DOI: 10.1016/j.comcom.2009.11.009. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2009.11.009.