Anonymity Online – Current Solutions and Challenges

Matthias Marx, Erik Sy, Christian Burkert, Hannes Federrath

Anonymity Online – Current Solutions and Challenges

Matthias Marx, Erik Sy, Christian Burkert, and Hannes Federrath

University of Hamburg, Germany

Abstract. Internet communication, regardless whether it is encrypted or not, comes with an abundance of protocol metadata. Web browsers reveal plenty of information to web applications. Additionally, web service operators have a great interest in their users' preferences and behaviour, leading to the development and deployment of several sophisticated tracking mechanisms. Therefore, the protection of the user's privacy on the Internet becomes increasingly difficult. Helpful privacy enhancing tools and techniques, which are often free of charge, are available to everyone, although have not reached widespread adoption yet. In this paper, we discuss different techniques of tracking as a challenge to online anonymity. Furthermore, we present current solutions on the application level as well as on the network level to provide anonymity, and finally we point out avenues for future research in the field of online anonymity. We find security-hardened operating systems promising to protect personal data against relatively strong adversaries on the user side. On the network side we consider lightweight network-based techniques like IPv6 pseudonymisation as promising technologies for future practical and usable anonymity on the Internet.

Keywords: privacy, anonymity, tracking, fingerprinting

1 Introduction to Anonymity

Anonymity online is the effort to communicate over the Internet while disclosing no or as little as possible of directly or indirectly identifying information unless explicitly wanted. Identification happens directly if someone discloses his or her own name accidentally or deliberately, e.g., when registering an account or signing a message. Indirect ways of identification emerge when disclosed information can be combined with additional knowledge containing sufficiently identifying information. When communicating over the Internet, being literally anonymous could mean that neither a communication partner, any intermediary, nor an outside observer is able to identify a communicating person.

This paper provides an overview of past and contemporary efforts to provide anonymity throughout different technology layers involved in online communication. Sect. 1 provides the legal background on online anonymity, the definition of important terms and a motivation for doing research in the field of online privacy. Sect. 2 discusses the problem of and countermeasures against application-level

tracking, followed by a discussion of network-level anonymisation techniques in Sect. 3. Sect. 4 concludes with an overview of the presented challenges regarding anonymity online.

1.1 Anonymity from a Legal Perspective

From a legal perspective, the term anonymity is known in association with privacy and data protection regulations. In this section, we focus on EU regulations and court decisions, i.e. the General Data Protection Regulation (GDPR), the ePrivacy Regulation, and a fundamental decision by the European Court of Justice regarding the identifiability of personal data.

EU General Data Protection Regulation (GDPR) The GDPR mentions anonymity only in its recitals. Recital 26 defines data as anonymous when it is either not related to an identified or identifiable natural person or it has been rendered anonymous such that the data subject is not or no longer identifiable. Therefore, anonymous data is not considered as personal data. Its processing does not fall within the material scope of the GDPR as defined in Article 2.

Whether given data is anonymous or not comes down to the question of how much effort is needed by an attacker to identify a person and how likely it is that the attacker undergoes such efforts, e.g., by linking several attributes of an anonymized dataset with a priori information held by the attacker. Recital 26 lists aspects that should be taken into consideration if one assess which means are likely to be used by an attacker. Aspects to be considered are the costs and the amount of time required for identification as well as the available technology at the time of the processing and technological developments.

EU ePrivacy Regulation As part of the amendment of EU data protection regulations, the ePrivacy Regulation (ePR) is developed to repeal the current ePrivacy Directive and to complement the GDPR as a lex specialis for electronic communication services.

Article 10 of the Commission's draft [1] proposes the obligation for software manufacturers to provide privacy settings that allow end-users to prevent third parties from storing or processing information on their devices. Third parties are components and services that are used by the first party services to fulfill a specific task like customer research or advertisement. Recital 22 of the Commission's proposal describes the proposed privacy settings as a more user-friendly approach to express consent to website cookies compared to the overwhelming numbers of consent requests currently prompted by individual websites. Recital 23 sees such privacy settings as an implementation of the principles of data protection by design and by default as defined in Article 25 of the GDPR and suggests differentiated cookie settings that should be prominently and intelligibly presented to end-users.

Regardless of the explicit reference to cookies in the recitals, Article 10's opt-out of processing information already stored on end-user devices could also

be interpreted as a prohibition of techniques that utilise stored device information, e.g. device fingerprinting. Such a wider interpretation of Article 10 would require software manufacturers to further confine the capabilities of third parties to query device or user-specific information. As a consequence, web browser manufacturers could be obliged to put code from third parties into a sandbox environment and deny unredacted access to senstive APIs.

Identification through IP Addresses In October 2016, the European Court of Justice (ECJ) ruled in case C-582/14 [2] with regard to the nature of dynamically assigned IP addresses, that such addresses, when registered by an online media service provider, constitute personal data, provided that the online media service provider has the legal means to identify a user behind the IP address by consulting additional data stored e.g. by the Internet Service Provider (ISP) of that person. This decision clarifies the fundamental dispute between the relative and the absolute approach to the identification of data subjects [3]. While the relative theory states that the question of identification depends on the individual processing context and the knowledge of the processor, the absolute theory considers data as identifiable as long as anyone has the means to attribute that data to a natural person [3]. ECJ followed the relative approach, but additionally used a wide interpretation of means of identification such that also information and capabilities of third parties should be taken into consideration if the processor has the legal means to utilise them.

1.2 Formalising Anonymity

Legal definitions of anonymity and identifiability are designed to be adaptable to future technological developments. Additionally, there is also a demand for more formalised and objective notions to facilitate compliance or to objectify scientific efforts. This section presents various formal definitions of anonymity and of algorithmic properties, which limit the disclosure of potentially personal data.

Anonymity Set Pfitzmann and Hansen [4] define anonymity as not being identifiable within a set of subjects, the anonymity set. Not being identifiable means that the subject is not distinguishable from other subjects within the anonymity set. Pfitzmann and Hansen describe the ability to distinguish subjects as a function of the attacker, its knowledge and capabilities. Thus, they state anonymity as a property relative to an attacker. A subject can be considered as being identified, if an attacker can attribute a given action to that subject with a probability exceeding a certain threshold. Pfitzmann and Hansen differentiate between anonymity of individual users of a system and global anonymity provided by the system. The level of global anonymity increases by a more even distribution of attribution probabilities. Consequently, a high global anonymity does not guarantee a high anonymity of each individual subject. Ensuring individual anonymity would require enforcing a uniform behaviour within a set of

subjects, which Pfitzmann and Hansen consider both very difficult to enforce and not desirable from a human rights perspective.

If a subject is known to be part of multiple anonymity sets, e.g. due to repeated interactions over time, the effective anonymity set is reduced to the intersection of all known anonymity sets. This is known as an *intersection attack*.

k-Anonymity Sweeney [5] defines a set of data records as k-anonymous, if every record is indistinguishable from at least $k-1$ other records in terms of identifiability. Such an indistinguishable group of records is called a bucket and corresponds to Pfitzmann and Hansen's anonymity set. Sweeney uses the term quasi-identifiers to denote a subset of record fields or attributes that are sensitive to linking attacks, i.e. attributes that are likely to appear in other data collections. Therefore, such quasi-identifiers can be used to correlate data collections and possibly disclose the identity of subjects. Given a data collection, Sweeney assumes, it is possible to find attributes that qualify as quasi-identifiers. Based on that assumption a data holder wishing to release an anonymised version of its data collection could redact values of quasi-identifiers until an acceptably large bucket size is reached. Under Sweeney's quasi-identifier assumption, variations within non-quasi-identifier attributes of the same bucket are not considered problematic.

ℓ-Diversity Machanavajjhala et al. [6] proposed ℓ-diversity as an enhancement of k-anonymity. Two attacks against k-anonymity are presented which utilise potentially low diversity within non-quasi-identifier attributes, denoted as sensitive attributes. The *homogeneity attack* shows that the size of a bucket is insignificant if a sufficiently high proportion of records within a bucket share the same sensitive attribute. A subject known to be part of that bucket can be assumed to share that value with reasonable likelihood, too. The *background knowledge attack* demonstrates that background knowledge about a subject can be used to single out the corresponding bucket and to reduce the anonymity set of that bucket by eliminating records that are incompatible with the attacker's background knowledge. As a countermeasure, ℓ-diversity requires ℓ different values for each sensitive attribute within a bucket.

t-Closeness Li et al. [7] demonstrate that the sensitive values within each bucket need to be distributed closely to the distribution of the overall data collection to avoid two kinds of attacks that are still possible with ℓ-diverse data. The *skewness attack* utilises a potential mismatch between the relative frequency of a stigmatising sensitive value within a bucket and that within the overall data collection. Based on that mismatch, an attacker can infer that subjects within that bucket are more likely to share the sensitive value than subjects in other buckets. The *similarity attack* shows that ℓ-diversity is not sufficient to protect against the homogeneity attack, if the diverse values are semantically similar and thus fall within the same category rendering the bucket homogeneous. Li et

al. introduce the notion of t-closeness which requires the value distributions of each bucket to differ no more than t from the distribution of the overall data collection.

Differential Privacy Dwork [8] introduces Differential Privacy not as a metric for the degree of anonymity of sanitised data collections, but as a property of data processing algorithms. An algorithm is considered ϵ-differentially private if it processes similar inputs into outputs which are only distinguishable with a certainty that is bound by ϵ. As a consequence, the probability is limited that such an algorithm exposes discernible information after adding or removing a subject from the input data. Differential Privacy does neither limit the amount nor the sensitivity of the information which is exposed within the tolerance of the ϵ boundary.

Summary While the definition of Pfitzmann and Hansen mostly conforms with legal notions of anonymity and is less formal, the other metrics aim at providing a provable property that allows an unambiguous reasoning about the sensitivity of data or algorithms. Such formalisation necessarily comes at the cost of simplification. Focusing on quasi-identifiers neglected that practically any information can be used in background knowledge attack to single out individuals or at least reduce the anonymity set. Regardless of the size of an anonymity set, information about subjects is disclosed if the anonymity set as a whole is abnormal and this abnormality reflects on all subjects within this anonymity set. For an analysis of the aforementioned notions and anonymisation techniques in the context of EU regulations we refer to the opinion paper of the Article 29 data protection working party [9].

1.3 Profiling and Unlinkability

Striving for anonymity and for less disclosure of personal data is not sufficient to protect individuals against a non-transparent and potentially malicious data processing. Even if a subject might not be identifiable by a processor, his or her data can be linkable and thus aggregated over time to build a profile of that subject. Consider a system in which each user is only represented and re-identified by a unique token, that is only meaningful in the context of that processor and not linkable to any external data. Profiles of such users would be considered legally anonymous if the processor had no likely means to identify the natural person behind that profile. For example, consider a news aggregation service, which knows neither its users' names nor e-mail addresses, IP addresses, or any other identifying information that has any meaning to third parties. The service recognises its users by a randomly chosen unique token and records their news preferences to provide a targeted selection of news to each user. This service could plausibly argue, that due to the lack of identifiability, no personal data is handled and therefore data protection regulations do not apply. Consequently,

this service could legally sell those profiles or use them for different purposes like targeted advertising without any restrictions.

Unlinkability is defined by Pfitzmann and Hansen [4] as the property of two or more items to be indistinguishable regarding their relation from the perspective of an attacker. Applied to the example of a news aggregation service, unlinkability of users' news consumption demands that two news requests by the same user appear indistinguishable from requests which were made by two different users. With unlinkability, profiling of users is impossible, since each profile comprises one item only. Unlinkability is a strong privacy guarantee which in turn comes at the cost of losing the ability to personalise services.

1.4 Lightweight Anonymity and Privacy

In 2012, Hsiao et al. proposed a setting for anonymous communication networks (ACNs) called Lightweight Anonymity and Privacy [10]. In this setting, the attacker model is relaxed, packets travel near-optimal routes, and only an intermediate level of privacy can be achieved. Lightweight anonymisation techniques can achieve higher efficiency compared to other anonymisation techniques. They can be a tool for so-called zero-effort privacy [11].

2 Anonymity on the Application Level

This section provides an introduction to tracking mechanisms on the application level and discusses their threat to anonymity online.

2.1 Tracking

Useful and legitimate applications for online tracking include the provision of personalised services, the distribution of personalised advertisements, and the measurement of website or application utilisation in order to derive patterns from the collected data. However, the collection and aggregation of this data can provide deep insights into online activities of a single user [12].

Tracking does not only influence the privacy of a user, it additionally introduces a high risk of discrimination based on the collected data. Findings by Hannak et al. [13] show that some e-commerce sites offer their products with different prices based on individual user profiles, which is also known as price discrimination.

In the following, we consider common tracking mechanisms (such as storage-based tracking and fingerprinting) which are capable to uniquely identify a computer system. Furthermore, we present behaviour-based tracking as an approach to identify a specific user instead of a computer system. Tracking mechanisms are a field of active research and development. Thus, countermeasures are required to continuously adapt protection mechanisms to the technical progress.

We consider the attacker to be a remote entity such as an online service with the privilege to store data or execute code on the victims machine. However, we

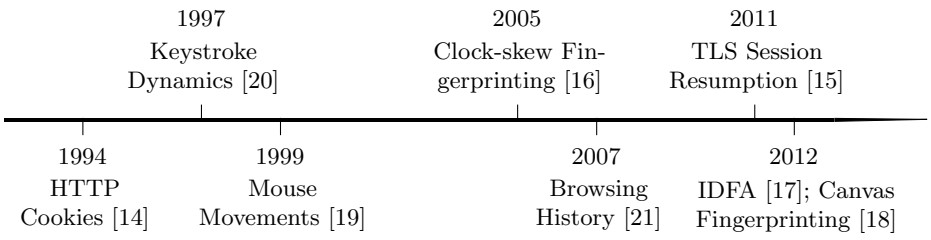


Fig. 1: Timeline of tracking mechanisms based on their first documented occurance.

assume the attacker is not capable of compromising the operating system of the host device. This assumption applies to website operators who run their code in the user's browser or to application providers, whose software is installed on the user's machine.

Storage-based Tracking With storage-based tracking mechanisms, users can be uniquely identified through information stored on their device. Among others, these mechanisms include HTTP cookies [14], session identifiers stored in hidden fields, Apple's advertising identifier (IDFA) [17], and the TLS session resumption cache [15]. For trackers, it might be attractive to access stored information which can persist on the user's device and is available within different sessions or even different applications such as the IDFA. For this group of tracking mechanisms, the user has at least theoretically the option to delete the stored tracking information and thus, thwart the creation of profiles.

Fingerprinting Fingerprinting mechanisms collect information about the user's computer system with the aim to discriminate the system among an anonymity set. Preferably, the collected information is difficult or unlikely to be changed by the user, thus, fingerprints can be used to recognise the user's computer system for long periods of time. However, if the fingerprint can extensively aggregate information about the user's computer system, the user might be recognised even after a partial modification of his or her system. Examples for fingerprinting are clock-skew measurements [16] and canvas fingerprinting [18].

Behaviour-based Tracking Behaviour-based tracking mechanisms aim to identify a user by characteristic traits such as mouse movements [19], keystroke dynamics [20], and browsing history [21]. While storage-based tracking and fingerprinting is capable to identify a specific device or application, the mechanism of behaviour-based tracking targets to identify a specific user. Research indicates, that the precision of identification for behaviour-based tracking decreases for large sample sizes [22]. In comparison, storage-based tracking can be used to

store unique information with a high entropy on a user's device and therefore has a negligible error rate when used for tracking purposes.

2.2 Current Challenges for Anonymity on the Application Level

Tracking does not only occur in the context of web browsing. Empirical studies show that many popular mobile applications connect to services on the Internet [23]. Half of the Top 100 mobile applications under iOS, Android and Windows Phone transfer personal information to online services [24]. Therefore, every application on a computer system needs to be regarded as a potential leak of personally identifying information. In the following we describe challenges and solutions for the prevention of tracking on the application level.

Access Control for Runtime Environments To restrict applications in their ability to collect, process, store and transfer personally identifying information access controls for runtime environments can be used. Therefore, this approach can enforce restrictions onto applications that reduce their functionality and thus limit their capabilities to track the user. For example, an application without access to network interfaces has no means to directly transfer collected data to a tracking service.

Unification of System Configurations The unification of system configurations aims to make tracking by fingerprinting more difficult by reducing the number of possible system configurations which are visible to an application, or by reducing the probability that a system deviates from the standard configuration. A naive utilisation of this approach is to reset the system configuration to the standard configuration after a period of time. However, a malicious application might still deduce differences in systems hardware or user specifics characteristics usage. It remains a challenge to address this issue and to reduce the capabilities of trackers.

Disguise User Behaviour This approach addresses the problem of behaviour-based tracking and aims to reduce the exposure of characteristic user behaviour towards malicious applications. For example, a system might use keystroke dynamics or mouse movements by randomly modifying the latency between two mouse or keyboard events. Characteristic user behaviour can also be deduced from contextual information such as browsing histories. Research by Herrmann et al. [25] indicates that datasets containing browsing sessions of 24 hours have 85.4% accuracy of finding and matching the sessions of the same users, while shorter sessions of 5 minutes yield only an accuracy of 30.4%. Besides such temporal schemes to disguise characteristic user behaviour, also contextual schemes could be applied, where sessions are separated when a new website is visited.

2.3 Current Solutions for Anonymity on the Application Level

The presented challenges point out, that current solutions, and popular operating systems in particular, do not sufficiently protect against tracking. We now investigate operating systems such as Tails [26], Qubes OS [27] with Whonix [28], and Subgraph OS [29] in terms of their ability to prevent tracking. These operating systems assume a stronger attacker model compared to popular operating systems and aim to solve the current challenges for anonymity online. A brief comparison of these operating systems is given in Table 1 and afterwards.

Tails Tails is designed as a live operating system, which is directly bootable from an external medium and aims to leave no traces of its usage on the computer used. As a security feature, the Tails OS provides high barriers for subsequently installed potentially malicious applications to persist after a system reboot. The drawback of this design decision is the limited usability of the system for users who want to install additional applications or to personalise their operating systems with an individual configuration. Tails uses the AppArmor [30] Linux kernel security module for access control policies of installed applications.

Qubes OS Qubes OS with a Whonix virtual machine aims at providing anonymous Internet access as well as strict security by confinements. In this context confinements describe security mechanisms to separate running programs. All applications are installed in virtual machines, and the host utilises the Xen hypervisor [31]. This architecture reduces the trusted code base of the host in comparison to typical monolithic operating systems such as the Linux kernel and therefore the attack surface is diminished. Single virtual machines can be configured with specific security configurations such as a restricted filesystem access, usage of ACNs, or firewall rules. However, the design of Qubes OS comes along with high hardware requirements for the execution of multiple parallel virtual machines.

Subgraph OS Subgraph OS provides anonymous Internet access and a design for strict application confinement. This operating system includes a hardened Linux kernel which is patched with Grsecurity/Pax [32] to provide additional memory protection and enhanced local access control. Subgraph allows a fine-grained confinement of individual applications by, among other mechanisms, application specific firewall rules, control of filesystem access, seccomp filter to restrict permitted system calls, isolation of some drivers such as audio, control of desktop access or process visibility.

2.4 Usability and Security

We tested whether a Tor Browser is able to collect hardware information on Qubes OS with Whonix, Subgraph OS, or Tails. We noticed that the default

System Properties	Tails	Qubes OS + Whonix	Subgraph OS
Live USB OS	Yes	No	No
Firewall Rules	Entire OS	Per VM	Per App
Filesystem Isolation	Per App	Per VM	Per App
Hardware Compatibility	High	Limited	High
ACN Usage	Default	Per VM	Per App
Host Architecture	Debian-based	Xen Hypervisor	Debian-based
Seccomp Filter	No	No	Yes, per App
Isolate Devices and Drivers	No	Yes	Yes
GUI Isolation	Yes	Limited, per VM	Limited, per App
Process Visibility	Yes	Limited, per VM	Limited, per App

Table 1: Comparison of operating systems with support for online anonymity.

privileges of the Tor Browser on these systems are sufficient to collect detailed hardware information. These information could be used for device fingerprinting.

Tails, Subgraph OS, and Qubes OS with Whonix provide the Tor Browser within their standard configuration, which provides the features of tab isolation and stream isolation [33] in order to protect against identification based on contextual information such as browsing habits of a user.

Furthermore, Qubes OS and Subgraph OS implement GUI isolation towards the restriction of applications in observing user behaviour in the context of other applications. However, the investigated operating systems do not provide functionalities to protect user tracking based on mouse movements or keystroke dynamics.

As a live operating system, Tails implements the unification of system configurations by resetting the system after each reboot. Consequently, storage-based tracking methods do not persist a restart of the system. Hence, this approach makes it more difficult for fingerprinting mechanisms to collect personally identifying information. However, malicious applications can still retrieve information about the hardware of the system.

Within Qubes OS the feature of disposable VMs supports the approach of a unification of system configurations. In this way, the user installs a potentially malicious application in a separate VM, which can be disposed after its usage. Thus, storage-based tracking over multiple sessions of application usage becomes more difficult, since the VM can be easily disposed in the meantime. Disposable VMs also improve the defence against fingerprinting mechanisms, since modifications of the VM by the user are removed with every disposal of the VM.

Tails and Subgraph OS are based on the Debian Linux distribution and provide a similar usability as Debian. However, as a live operating system, which returns to its initial state after each reboot, Tails has limited use cases. In Qubes OS, it is the responsibility of the user to isolate applications from each other by installing them in different virtual machines. Thus, Qubes OS requires a higher security awareness of the user which limits its usability. As a negative side effect

on usability, Qubes OS has higher hardware requirements in comparison to the other operating systems.

3 Anonymity on the Network Level

This section provides a chronology of techniques which have been used to prevent tracking on the network level. Furthermore, we will present selected applications that are in use today. Finally, we discuss current challenges.

3.1 Chronology

In 1978, Rivest, Shamir and Adleman presented a method for obtaining digital signatures and public key cryptosystems that became known as RSA cryptosystem [34]. The cryptosystem had the novel property that a public encryption key does not reveal the corresponding decryption key. A sender encrypts the message to be sent with the receiver's public key and transmits it via a potentially insecure channel. Only the receiver could decrypt the message with his or her secret private key. Similarly, messages can be signed [34]. The RSA cryptosystem serves as building block for various privacy enhancing techniques until today.

A technique based on public key cryptography that allows unlinkability of sender and recipient was presented by Chaum in 1981. The basic idea makes use of so-called mixes, which sample messages of same length in a batch, change their appearance and forward all of them at the same point of time but in a different order [35]. Unlinkability can be achieved if more than one mix is used, if the mixes are operated by different operators and if at least one mix operates trustworthy and honestly. In the same paper, Chaum introduced digital pseudonyms. A digital pseudonym is a public key with which digital signatures of an anonymous holder of the corresponding private key can be verified. A combination of mixes and digital pseudonyms enables electronic elections in which any party can verify that the votes have been properly counted [35]. In 1991, Pfitzmann et al. presented ISDN-MIXes, a combination of Chaum's mixes, dummy traffic and broadcasts. ISDN-MIXes allow untraceable communication with low communication overhead [36]. Federrath et al. presented mixes in mobile communication systems in 1991. Their mix-based system utilises untraceable return-addresses to hide routing information and achieves location privacy [37]. In 1998, Kesdogan et al. introduced Stop-and-Go-MIXes that provide probabilistic anonymity. Unlike other mixes, Stop-and-Go-MIXes do not collect a fixed number of messages in a batch [38].

Blind signatures were introduced by Chaum in 1983. As traditional digital signatures, blind signatures can guarantee authenticity of a message. However, blind signatures allow signing of a message without revealing the message itself to the signer [39]. Blind signature schemes are utilised in electronic cash and electronic voting systems.

Tracking organisations use personally identifiable information, such as name, date and place of birth, or address, to match or link records with those provided

by other organisations. With Chaum's credential system, presented in 1985, an individual could use unlinkable pseudonyms to interact with different organisations. For instance, a one-time-use pseudonym may be used to purchase goods from a shop and a persistent pseudonym may be used to open an account with a bank. The credential system ensures that individuals are held accountable for abuses created under their pseudonyms. Also, organisations could limit the number of pseudonyms per individual and individuals are able to authenticate ownership of their pseudonyms [40].

A communication protocol that achieves unconditional sender and recipient untraceability was published by Chaum in 1988. In contrast to mix networks, the Dining Cryptographers Network (DC-Net) relies on secure multi-party computation [41].

In a privacy preserving value exchange (e.g. unobservable and anonymous exchange of digital money) over a network, the main problem is the lack of simultaneity. It gives a temporary advantage to one party who can stop the communication midway through the value exchange process. In 1990, Bürk and Pfitzmann compared two approaches that utilise third parties to overcome this problem [42].

In 1995, Cooper and Birman introduced a service that allows reading from a shared memory without revealing which piece of information is being read. Those so-called blind message services can be used as an alternative to mixes to build a message service that achieves location privacy [43].

Goldschlag, Reed and Syversen introduced onion routing in 1996 [44], a lightweight approach to the dissemination of mixes.

Fig. 2 shows a timeline of development of the aforementioned privacy enhancing technologies.

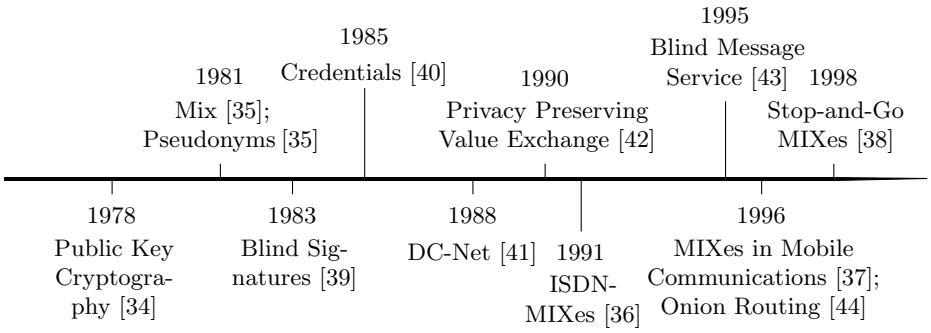


Fig. 2: Timeline of development of privacy enhancing technologies.

3.2 Current Solutions

JAP, formerly known as Java Anon Proxy, is a mix-based solution for anonymous Internet access. A first test version was launched in October 2000 and was developed by the research project *AN.ON – Anonymity.Online* [45]. The full service is running since February 2001 and has been outsourced to JonDos GmbH [46]. JAP is currently used by 5000 paying and several thousand non-paying users [47]. The research project named *AN.ON-Next – Anonymity Online for the next Generation* [48] aims to further develop JAP's mix-based anonymisation techniques.

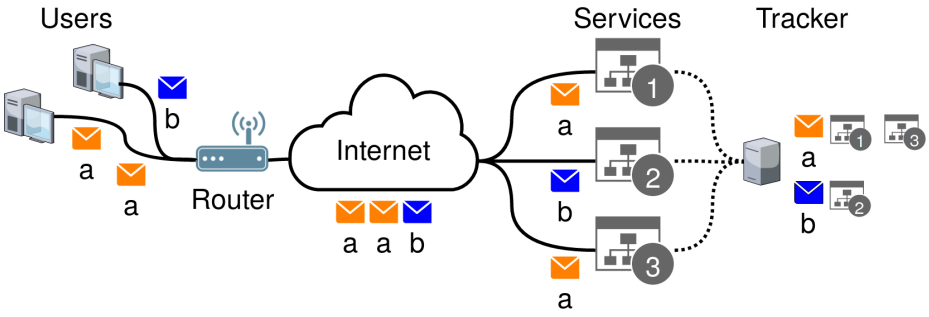
Tor is the most widely used anonymisation network. It was presented by Dingledine et al. in 2004 and is based on onion routing [49]. Today, Tor has millions of users [50]. The servers that are used to relay traffic are mainly run by volunteers. One noteworthy product of the Tor project is the Tor Browser. It combines Tor's protection on the network level with a modified Mozilla Firefox Browser in order to thwart tracking based on fingerprinting on the application level [51]. JAP and Tor protect against relatively strong adversaries. Other solutions focus on weaker adversaries.

IPv6 Pseudonymisation Long-lived IP addresses are one of the easiest ways for tracking. Daily changing IP addresses (on reconnect or through IPv6 Privacy Extensions) are not sufficient [25]. As long as unlinkability of actions against ad networks and websites based on an IP address is intended and sufficient, ISPs can offer anonymity with a new approach to IP address assignment. Multiple users could share the same IP address (*Address Sharing*) or a single user could frequently change his or her IP address (*Address Hopping*). The anonymisation functionality can be implemented on the router or in the datacenter of the ISP. An advantage of this solution is that users are not required to make any changes to their operating system or client software and hardware. As a positive side effect, also devices, whose network configuration can not be changed, can be protected.

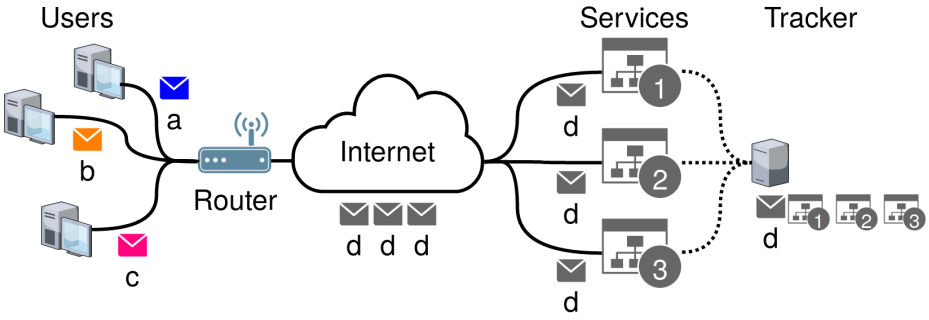
In our attacker model, we assume the user's ISP to be trustworthy. The attacker can be a web service operator who controls one or multiple web services, or a third party tracker that links activities across multiple web services. Furthermore, the attacker could be a man-in-the-middle between the ISP's border routers and the connection's endpoint.

Fig. 3a shows two users with different IP addresses `a` and `b` who communicate over the Internet with three different services `1`, `2` and `3`. Without IPv6 pseudonymisation, a third party tracker that is embedded in all three services can conclude that the user with IP address `a` is using the services `1` and `3` and that the user with IP address `b` is using the service `2`.

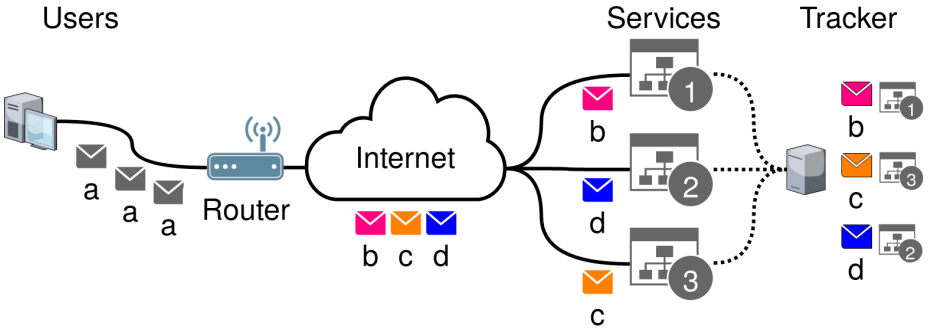
With *Address Sharing*, one IP address is shared among multiple users at a given point in time. All users are using the same IP address and thus form an anonymity group, implying that trackers cannot distinguish users based on their IP address anymore. *Address Sharing* can be implemented with state-of-the-art techniques such as Network Address Translation (NAT) has been deployed on



(a) Without IPv6 pseudonymisation, a third party tracker can conclude that the user with IP address `a` is using the services `1` and `3` and that the user with IP address `b` is using the service `2`.



(b) With *Address Sharing*, one IP address is shared among multiple users. The third party tracker cannot discriminate between the users based on their IP addresses.



(c) With *Address Hopping*, each user distributes his or her traffic over multiple IP addresses. It seems that the different services are being used by different users.

Fig. 3: IPv6 pseudonymisation.

Internet gateways for decades. Fig. 3b shows three users who communicate via a router through the Internet with different services. Each user has his or her own IP address `a`, `b` or `c`. The router replaces the IP addresses on its public network interface, with the result that all users share the same public IP address `d`. The third party tracker cannot discriminate between the users based on their IP addresses. From the viewpoint of the tracker, it appears to be a single user with IP address `d` who is using services `1`, `2` and `3`.

Address Hopping means that each user distributes his or her traffic over multiple IP addresses within a short period of time. Web services or third party trackers can link activities for which the same IP address is used. However trackers cannot link activities based on IP addresses, when the IP address is changed frequently. Fig. 3c shows a single user with IP address `a` who communicates via a router through the Internet with different services. The router replaces the IP address, with the result that the different services `1`, `2` and `3` see different IP addresses `b`, `c` and `d`. A third party tracker cannot link the user's IP addresses easily. It seems (for the attacker) that the different services are being used by different users. Given the large number of IP addresses that are assigned to each user with IPv6 [52], *Address Hopping* becomes possible.

4 Conclusion

We introduced multiple perspectives on online anonymity, including recent developments in EU data protection regulations regarding anonymity. We argued that certain forms of aggregated user profiles might be considered legally anonymous. To extend the scope of data protection regulations to such profiling, the concept of linkability needs to be incorporated into legal interpretations of identifiability and personal data.

We discussed fingerprinting as well as storage-based and behaviour-based tracking and the challenges to achieve online anonymity on the application level. Security hardened operating systems Tails, Subgraph OS, and Qubes OS with Whonix may support the users efforts to online anonymity. However, they do not completely protect against tracking by installed applications.

Building anonymous communication networks on the network level is a challenging effort. As examples, we introduced Tor and JAP, both of which protect against strong adversaries. Lightweight solutions such as IPv6 pseudonymisation which protects against a weaker adversary but aims for a broader public have the potential to gain broad acceptance as the protection comes usually without any significant performance limitations.

5 Acknowledgement

Part of this work has been developed in the projects AN.ON-next (reference number: 16KIS0368) and AppPETs (reference number: 16KIS0381K). Both projects are partly funded by the German ministry of education and research (BMBF).

We thank Ephraim Zimmer and Tobias Mueller for their helpful comments and discussion.

References

1. European Commission: Proposal for a regulation on privacy and electronic communications. https://ec.europa.eu/digital-single-market/en/proposal-eprivacy-regulation Accessed: 2017-11-27.
2. European Court of Justice: Judgement on Case C-582/14. http://eur-lex.europa.eu/legal-content/en/TXT/PDF/?uri=uriserv%3AOJ.C_.2016.475.01.0003.01.ENG Accessed: 2017-11-27.
3. Reid, A.S.: The European Court of Justice case of Breyer. Journal of Information Rights, Policy and Practice **2**(1) (2017)
4. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. (2010)
5. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05) (2002) 557–570
6. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), IEEE (2006) 24–24
7. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: Proceedings of the 23rd International Conference on Data Engineering (ICDE'07), IEEE (2007) 106–115
8. Dwork, C.: Differential privacy. In: Automata, Languages and Programming. Volume 4052 of Lecture Notes in Computer Science., Springer, Berlin, Heidelberg
9. Article 29 Data Protection Working Party: Opinion 05/2014 on Anonymisation Techniques. (2014)
10. Hsiao, H.C., Kim, T.H.J., Perrig, A., Yamada, A., Nelson, S.C., Gruteser, M., Meng, W.: LAP: Lightweight anonymity and privacy. In: Security and Privacy, IEEE (2012) 506–520
11. Herrmann, D., Lindemann, J., Zimmer, E., Federrath, H.: Anonymity Online for Everyone: What is missing for zero-effort privacy on the Internet? In: International Workshop on Open Problems in Network Security, Springer (2015) 82–94
12. Bujlow, T., Carela-Español, V., Solé-Pareta, J., Barlet-Ros, P.: A Survey on Web Tracking: Mechanisms, Implications, and Defenses. Proceedings of the IEEE (2017)
13. Hannak, A., Soeller, G., Lazer, D., Mislove, A., Wilson, C.: Measuring price discrimination and steering on e-commerce web sites. In: Proceedings of the 2014 conference on internet measurement conference, ACM (2014) 305–318
14. Schwartz, J.: Giving Web a Memory Cost Its Users Privacy. http://www.nytimes.com/2001/09/04/business/giving-web-a-memory-cost-its-users-privacy.html Accessed: 2017-11-27.
15. Perry, M.: Disable TLS Session resumption and Session IDs. https://trac.torproject.org/projects/tor/ticket/4099 Accessed: 2017-11-27.
16. Kohno, T., Broido, A., Claffy, K.C.: Remote physical device fingerprinting. IEEE Transactions on Dependable and Secure Computing **2**(2) (2005) 93–108
17. Edwards, J.: Apple Has Quietly Started Tracking iPhone Users Again, And It's Tricky To Opt Out. http://www.businessinsider.com/ifa-apples-iphone-tracking-in-ios-6-2012-10 Accessed: 2017-11-27.

18. Mowery, K., Shacham, H.: Pixel perfect: Fingerprinting canvas in HTML5. Proceedings of W2SP (2012) 1–12
19. Goecks, J., Shavlik, J.: Automatically labeling web pages based on normal user actions. In: Procedings of the IJCAI Workshop on Machine Learning for Information Filtering. (1999)
20. Monrose, F., Rubin, A.: Authentication via keystroke dynamics. In: Proceedings of the 4th ACM conference on Computer and communications security, ACM (1997) 48–56
21. Padmanabhan, B., Yang, Y.C.: Clickprints on the web: Are there signatures in web browsing data? (2007)
22. Banerjee, S.P., Woodard, D.L.: Biometric authentication and identification using keystroke dynamics: A survey. Journal of Pattern Recognition Research **7**(1) (2012) 116–139
23. Sy, E., Mueller, T., Marx, M., Herrmann, D.: AppPETs: A Framework for Privacy-Preserving Apps. SAC 2018: Symposium on Applied Computing , April 9–13, 2018, Pau, France (2018)
24. Ren, J., Rao, A., Lindorfer, M., Legout, A., Choffnes, D.: Recon: Revealing and controlling pii leaks in mobile network traffic. In: Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, ACM (2016) 361–374
25. Herrmann, D., Banse, C., Federrath, H.: Behavior-based tracking: Exploiting characteristic patterns in DNS traffic. Computers & Security **39** (2013) 17–33
26. The Tails Project: Tails. `https://tails.boum.org` Accessed: 2017-11-27.
27. The Qubes OS Project: Qubes OS. `https://www.qubes-os.org` Accessed: 2017-11-27.
28. Whonix developers: Whonix OS. `https://www.whonix.org` Accessed: 2017-11-27.
29. Ahmad, D.M.: Subgraph OS. `https://subgraph.com` Accessed: 2017-11-27.
30. AppArmor developers: AppArmor. `http://wiki.apparmor.net` Accessed: 2017-11-27.
31. Xen developers: Xen Project. `https://www.xenproject.org` Accessed: 2017-11-27.
32. Open Source Security, Inc: Grsecurity. `https://grsecurity.net` Accessed: 2017-11-27.
33. Appelbaum, J.: Description of Tor Stream Isolation. `http://archives.seul.org/or/dev/Jul-2010/msg00021.html` Accessed: 2017-11-27.
34. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2) (1978) 120–126
35. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM **24**(2) (1981) 84–90
36. Pfitzmann, A., Pfitzmann, B., Waidner, M.: ISDN-mixes: Untraceable communication with very small bandwidth overhead. In: Kommunikation in verteilten Systemen, Springer (1991) 451–463
37. Federrath, H., Jerichow, A., Pfitzmann, A.: MIXes in mobile communication systems: Location management with privacy. In: Information Hiding, Springer (1996) 121–135
38. Kesdogan, D., Egner, J., Büschkes, R.: Stop-and-Go-MIXes Providing Probabilistic Anonymity in an Open System. In: Information Hiding. Volume 98., Springer (1998) 83–98
39. Chaum, D.: Blind signatures for untraceable payments. In: Advances in cryptology, Springer (1983) 199–203
40. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM **28**(10) (1985) 1030–1044

41. Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. Journal of cryptology **1**(1) (1988) 65–75
42. Bürk, H., Pfitzmann, A.: Value exchange systems enabling security and unobservability. Computers & Security **9**(8) (1990) 715–721
43. Cooper, D.A., Birman, K.P.: Preserving privacy in a network of mobile computers. In: Security and Privacy, IEEE (1995) 26–38
44. Goldschlag, D.M., Reed, M.G., Syverson, P.F.: Hiding routing information. In: Information Hiding, Springer (1996) 137–150
45. AN.ON: Anonymität.Online. `https://anon.inf.tu-dresden.de` Accessed: 2017-11-27.
46. JonDos GmbH: JonDonym. `https://www.anonym-surfen.de` Accessed: 2017-11-27.
47. JonDos GmbH: Status der Mixkaskaden. `https://www.anonym-surfen.de/status/` Accessed: 2017-11-27.
48. AN.ON-Next: Anonymität Online der nächsten Generation. `https://www.anon-next.de` Accessed: 2017-11-27.
49. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC (2004)
50. Tor Project: Tor Metrics. `https://metrics.torproject.org` Accessed: 2017-11-27.
51. Perry, M., Clark, E., Murdoch, S., Koppen, G.: The Design and Implementation of the Tor Browser. `https://www.torproject.org/projects/torbrowser/design/` (2017)
52. Narten, T., Huston, G., Roberts, L.: IPv6 Address Assignment to End Sites. Technical Report 157, RFC Editor (2011)