NEXTLEAP: Decentralizing Identity with Privacy for Secure Messaging

Harry Halpin

NEXTLEAP: Decentralizing Identity with Privacy for Secure Messaging

Harry Halpin
Inria
2 rue Simone Iff
Paris, France 75589 Paris Cedex 12
harry.halpin@inria.fr

ABSTRACT

Identity systems today link users to all of their actions and serve as centralized points of control and data collection. NEXTLEAP proposes an alternative decentralized and privacy-enhanced architecture. First, NEXTLEAP is building privacy-enhanced federated identity systems, using blind signatures based on Algebraic MACs to improve OpenID Connect. Second, secure messaging applications ranging from Signal to WhatsApp may deliver the content in an encrypted form, but they do not protect the metadata of the message and they rely on centralized servers. The EC Project NEXTLEAP is focussed on fixing these two problems by decentralizing traditional identities onto a privacy-enhanced based blockchain that can then be used to build access control lists in a decentralized manner, similar to SDSI. Furthermore, we improve on secure messaging by then using this notion of decentralized identity to build in group messaging, allowing messaging between different servers. NEXTLEAP is also working with the PANORAMIX EC project to use a generic mix networking infrastructure to hide the metadata of the messages themselves and plans to add privacy-enhanced data analytics that work in a decentralized manner.

KEYWORDS

decentralization, privacy, secure messaging, identity, anonymity

ACM Reference format:
Harry Halpin. 2017. NEXTLEAP: Decentralizing Identity with Privacy for Secure Messaging. In *Proceedings of ARES '17, Reggio Calabria, Italy, August 29-September 01, 2017,* 10 pages.
DOI: 10.1145/3098954.3104056

1 INTRODUCTION

How can a user control their identity while retaining privacy in the era of the increasingly centralized cloud? Currently, there is no alternative: Yet the NEXTLEAP project hopes to change that by making the "next leap" ahead in designing a suite of privacy-enhanced and decentralized protocols to build applications that let an entity, from an individual user to a large international organization, control their own identity in the cloud while maintaining their privacy, not only from potential adversaries but from the cloud providers themselves. Indeed, we consider the cloud itself to be the adversary. Note that the cloud is a *distributed* system par-excellance, where we define a distributed system as "a system with multiple components that have their behavior co-ordinated via message passing without the use of a central clock [20], where these components are usually spatially separated and communicate using a network, and may be managed by a single root of trust or authority" while in contrast, a *decentralized* system is a system where "multiple authorities control different components and no single authority is fully trusted by all others" such that "decentralized systems are a subset of distributed systems." [32] If the various trusted authorities can be compromised, the cloud itself is adversarial. This is not entirely far-fetched, as the PRISM programme of the NSA showed that cloud providers, even if secure to outside adversaries, could be compromised internally and illegally engage in surveillance over user data.[1] The goal of NEXTLEAP is to create a "NEXT-generation Legal Encryption Access Project" that takes into account the cloud as an adversarial environment, and allows the rights of ordinary people to be exercised securely over their identity without sacrificing the availability and reliability of cloud environments.

To fix these problems, NEXTLEAP aims to create a secure identity for a federated cloud environment, built in a user-centric manner that allows anonymity, that lets users take control of their own data while still working collectively and socially, as motivated in Section 2. Today users are left at the mercy of large cloud providers such as Google, Facebook, and others more due the lack of functional alternatives, and part of it is not only the "network effect" but that fundamental research problems have largely been ignored by researchers. Our position is that a comprehensive, although far from complete, decentralized security solution will require at least the following problems to be solved:

(1) Federated Identity
(2) Decentralized PKI
(3) Encrypted (Group) Messaging
(4) Privacy-Enhanced Data Analytics

The logic is simple: As currently identity is the most valuable part of a user's life and as such is naturally federated between different aspects of life. In order to build federated identity that can work in terms of actual deployment, we should build on top of existing standards but make them privacy-enhanced with the minimal possible changes, as we develop in Section 3. Once we have a privacy-preserving identity system, in Section 4 we show how to connect this identity system to private key material in order to fulfill use-cases such as access control over data, without falling

ARES '17, Reggio Calabria, Italy
 978-1-4503-5257-4/17/08…$15.00
DOI: 10.1145/3098954.3104056

[1] https://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data

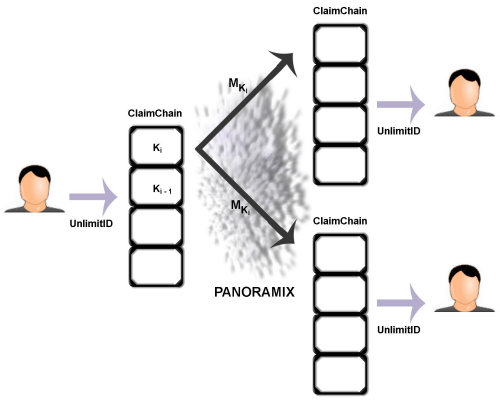

Figure 1: The NEXTLEAP architecture

into the trap of trying to assign "one key per user" via a centralized authority. Lastly, once a decentralized identity has been established with associated key material, Section 5 shows how messages of various kinds need to be sent between identities, ranging from simple e-mail and chat applications to possibly monetary payment or other payloads. In detail, we can use a decentralized PKI to build a decentralized identity layer on top of the well-known Signal protocol. Lastly, one advantage of the cloud is massive machine-learning. As this functionality depends on statistics over groups rather than individual users, we propose techniques for allowing privacy-enhanced data aggregation over identities and messages in Section 6. Note that as the NEXTLEAP project is still ongoing at this moment, Sections 3 and 4 are more mature than the rest of the research presented in this paper.

The diverse components of the NEXTLEAP architecture fits together in the manner shown in Figure 1. As shown, a user wishes to send the same end-to-end encrypted message M encrypted with her public key k_i to two or more recipients. To send a message, she has to authenticate to her identity provider and authorize the sending of the message, using a privacy preserving technique for federated identity called UnlimitID developed by NEXTLEAP that is detailed in Section 3. Her key is stored in her claimchain, a blockchain-based decentralized audit log of both her present key material and past key material described in Section 4. She can retrieve the most recent keys from the receiver(s) of her message by querying their chain and may validate it against her existing information or information from other chains. Then the message is sent using a secure group messaging protocol detailed in Section 5. These messages can have metadata hidden, such as recipient, timing, and size information by using the mix-networking infrastructure developed by the PANORAMIX EC project.[2]

2 MOTIVATION

Why not just hand data over to the increasingly centralized cloud providers? One motivating factor for a more decentralized internet is *technological sovereignty*. Alternative models to centralized cloud providers exist: For example, emerging community wifi and broadband may even provide infrastructure as a common good in some regions, and we can imagine a counter-vailing tendency to store data autonomously in order to provide security and privacy properties not easily implemented in centralized clouds. Many countries face an emergent situation of cybercolonialism, "the policy or practice of acquiring full or partial political control over another country's cyber-space, occupying it with technologies or components serving foreign interests, and exploiting it economically."[3] This is precisely what is at stake in the current debate about the NSA's and GCHQ's pervasive surveillance infrastructure, including historic attempts to prevent the development and widespread use of security and cryptology technologies, and their current active compromise of international communications and end-users. To summarize the argument, as long as people do not control their own means of communication, they leave themselves open to domination and exploitation by those that do control their communication. This control either be active and with the consent of the cloud providers, such as the active storing of data overseas where it is under usually weaker protections or it can be passive and without consent, such as the massive metadata surveillance operations by the NSA, as no meaningful international treaty can technically defend a poorly-designed protocol like e-mail (SMTP) against surveillance. However, calls to force companies to comply with European Union regulations are often ignored or implemented in a superficial manner, as illustrated by the debacle around the W3C "Do Not Track" standard and recent Google compliance (or lack thereof) with "the right to be forgotten." Indeed, a call for a purely regulatory approach is bound to fail as there is not enough leverage over Silicon Valley, and even less over companies in jurisdictions such as China and Russia. Rather than leave most of the world in the unfortunate situation where it must simply chose what super-power gets to spy on their activities, technical alternatives should be created.

The success of decentralized system is needed to improve a country's innovation capacity, not only its political sovereignty. An example of how this plays out in action in Europe is the "Made in Germany" e-mail effort, where major e-mail providers such as T-Online, GMX, Web.de were brought together with companies such as Deutsche Telekom in order to make sure email never leaves German servers. This effort has led to a giant boom in demand, leading to an increase in uptake after launch that gave impetus to its implementation. The most important opportunity in Europe is the revised General Data Protection Regulation to replace the dated 1995 Data Protection Directive. Already we are seeing the beginning as rulings such as the 2014 "right to be forgotten" ruling have had a large impact on centralized providers. As the General Data Protection Regulation enforces an approach based on privacy by design where data minimization (such as metadata minimization) and end-to-end encryption are required (as well as storage of data in local jurisdictions), it could help the adoption of decentralized and privacy-preserving protocols.

NEXTLEAP is unusual as it is not funded via the Trust and Security Unit of the European Commission, but by Collective Awareness

[2] http://panoramix-project.eu/

[3] https://conspicuouschatter.wordpress.com/2014/06/21/the-dawn-of-cyber-colonialism/

Platforms (CAPs) initiative to use the internet for social good.[4] Under the CAPS programme, there is increasing research on topics such as the common-based peer production and research projects around decentralization, but the CAPs community does not yet know how to socially, legally, and technically build tools that harness these commons while preserving privacy and extending fundamental rights. Despite their desire to help the public interest in some fashion, commons-based projects ranging from Wikipedia to blockchain present serious privacy and security problems that must be solved to harness them for the public good, lest inadvertently users simply hand over even more data to be harvested by centralized services or the decentralized alternatives actually be *worse* than centralized services in terms of privacy and security. Current efforts to create decentralized alternatives to the Cloud typically are not taking these problems into consideration, instead focusing decentralization without privacy: For example, many users believe that the blockchain is privacy-preserving and anonymous, while it is actually a public ledger that is transparent and allows easy data-mining. Alternative efforts such as those based on Linked Data are still dependent on the centralized domain name system. Often projects to use the Semantic Web for decentralized social networking such as SoLiD[5] do not have clearly described or incorrect security properties.[6] Emergent code-bases such as SwellRT[7] from the P2PValue CAPS project[8] and Objective8 from D-CENT CAPS project[9] both have failed to have any definition of privacy-by-design or explicitly stated security policies, and have all failed to reach mass adoption. However, failure may be a blessing in disguise as the lack of a clear success for decentralization allows new protocols and designs to be incorporated into software before fundamentally insecure protocols reach mass adoption and so could suffer from security incidents that would drive users away from decentralized alternatives.

3 FEDERATED IDENTITY

3.1 Problem: Privacy in Federated Identity

Identity is the opposite of anonymity, where both personal data and possibly unintentional "digital traces" can be *unlinked* from the user. In opposition to identity systems that attempt to "link" attributes to a user across systems, anonymity systems aim for a property of "unlinkability," namely that the same user can use multiple services without their identity (and behavior) being linked. Anonymity has classically been defined as "the state of not being identifiable within a set of subjects," where that set of subjects is called the anonymity set [25]. Note that an anonymity set of possible subjects has been refined in terms of information-theoretic measures that look at anonymity on a much-more fine-grained level, such as the probability that a given identity is attached to a given transaction [30]. Anonymity is usually defined not as either "yes" or "no," but in terms of the anonymity set that can be gathered by an adversary attempting to identify a particular user or users. The goal of NEXTLEAP is to design a system that allows a user to maintain their privacy by associating their identity with a larger set of users, and so unlinking it from any particular action, while still allowing the identity framework to prove claims about the user.

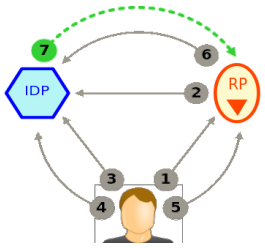

Figure 2: OpenID and UnlimitID Information Flow

In terms of identity, an identity ecosystem is a collection of services that wish to share data about an entity. A user that is sending some kind of information to a *relying party* (RP), a services that wishes to access verified attributes or claims about the entity. The source of the claims is called an *identity provider* (IdP), a service that stores and can possibly verify identity claims on behalf of a user. The common example would be having a user send their username and password combination to Facebook via Facebook Connect, the identity provider, to sign-on to a third party service such as a newspaper like the Guardian, the relying party. The Guardian also may require some information from Facebook, such as the full name of the users and their interests in their Facebook profile, in order to customize their service.

3.2 Background: OpenID-based Federated Identity

OpenID Connect is a popular federated identity system meant to provide much of the same functionality as Facebook Connect, and is deployed by large identity providers (email providers) such as Google and Microsoft.[10] OpenID Connect builds upon the well deployed base of OAuth 2.0 standard for server-side claim exchange,[11] but optimizes certain elements of OAuth for server-side exchange of identity claims and requires no changes to current browsers. OpenID Connect uses OAuth 2.0 for the authorization flow for Single-Sign On (SSO) while adding a small number of non-opaque identifiers in the response between an identity provider and relying party as well as adding more detailed hooks for using cryptographic signing. Once the user authenticates to an identity provider (usually via a HTTP redirection and a username-password) and so provides the relying party an access token, the identity claims are passed directly from the identity provider to the relying party, and the user is out of the loop. The flow of OpenID Connect is illustrated in Figure 2 (the transfer of identity claims is given in *green* in this diagram) and outlined below:

(1) A user visits a relying party that needs identity claims.
(2) The relying party makes a request for identity claims to the identity provider.

[4] https://ec.europa.eu/digital-single-market/en/caps-projects
[5] https://github.com/solid/solid-spec
[6] For example, WebID depends on MD5 and the deprecated *keygen* tag. This tag, and so WebID with TLS, is being deprecated as the use of private keys cross browsers breaks the Same Origin Policy: https://groups.google.com/a/chromium.org/forum/#!msg/blink-dev/pX5NbX0Xack/kmHsyMGJZAMJ
[7] http://swellrt.org/
[8] http://p2pvalue.eu/
[9] https://objective8.dcentproject.eu/
[10] https://openid.net/connect/
[11] https://oauth.net/2/

(3) The user is redirected to the identity provider from the relying party.
(4) The user authenticates to the identity provider (typically using a username-password combination) and is granted a bearer token.
(5) User is redirected back to relying party and grants authorization token to relying party.
(6) The relying party sends the authorization token to the identity provider and receives an access token (a bearer token with a scope and limited lifespan).
(7) While the access token is valid, the identity provider sends identity claims to the relying party.

OpenID Connect gives the identity provider ability to observe all requests for identity claims by relying parties, which is the primary flaw from a privacy standpoint as identities cannot be delinked from the identity provider. As the traffic of identity claims flows directly between identity provider and relying party, the interaction between the user and a relying party can be logged by the identity provider, as well as traced by third-parties via traffic analysis between the relying party and identity provider. This particular flow has the advantage of possibly authorizing requests for personal data when the user is not online and thus unable to directly intervene at the time of the request, a distinct advantage for some use-cases (such as when the user authorizes the requests ahead of time or on a regularly occurring basis), it is also a danger, as the identity provider may exchange user data with relying parties without the consent of the user, leading to the possibility of identity interactions being unknown to the user. As regarding anonymity, although the architecture of OpenID Connect does not require that identifiers be persistent when sent to the relying parties (and thus allows anonymity to relying parties), the authentication mechanism to the identity provider does not authenticate particular capabilities but instead identifies the entire user or personae on a coarse-grained manner to the identity provider, and so the identity provider is aware of all relying party requests even if the user is anonymous to the relying party. Thus, OpenID Connect can be thought of as absolutely trusting the identity provider, which may be a reasonable assumption in some circumstances but this seems to be a poor choice for use-cases that require a higher degree of privacy.

3.3 Solution: Privacy-Enhanced Federated Identity via UnlimitID

NEXTLEAP has developed UnlimitID, a profile of OAuth 2.0 that can be used in place of OpenID Connect, that is aimed to providing unlinkability such that IdPs and RPs cannot link the different pseudonyms of a user with the same or across RPs [19]. Our design also allows selective attributes disclosure where users can choose which subset of their claims to reveal to the RPs/IdP each time, and claims are locally blinded by the user and can be used an arbitrary number of times before their expiration date without revealing the identity of the owner, and sybil resistance where IdPs can enforce that users may create up to a certain number of pseudo-identities with each RP, with the possibility to limit users to a single pseudonymous account per RP. To achieve these properties, UnlimitID uses selective disclosure credentials, based on algebraic messages authentication (Algebraic MACs) [7], to achieve its undetectability and unlinkability properties.

UnlimitID achieves unlinkability by changing two only steps of the standard OpenID protocol. First, prior to commencing the standard OpenID flow, a user authenticates to the IdP and gets issued a credential encoding a long-term secret key and a key/value pair denoting an attribute. The credential also includes an expiry date after which it will need to be re-issued. All credentials issues contain four fields: a secret key that is unknown to the IdP, a number of pairs of attributes, and an expiration time that defines a temporal epoch where the credentials may be used. The user blinds the credential and deposits it back to the IdP to generate a pseudo-identity specific to a RP service in Step 4 of the OpenID protocol flow. At this point the IdP acts in its normal role in the OpenID Connect protocol, using the pseudo-identity to validate the claims and send blinded claims to RPs using the OpenID Connect flow. At the end of every epoch, a user must re-authenticate to the identity provider in order to get more blinded claims.

The use of blinding via Algebraic MACs preserves the property of unlinkability and the claims the user receives in the first phase can be limited to claims the user wants to reveal to a particular RP, and so the user has the ability to enable selective disclosure. Since authentication is done as normal to IdP the user can authenticate to the RP via the IdP (using the standard redirect-based OpenID Connect flow) and so the user's actual identity is not revealed to the RP, while simultaneously the IdP has to be authenticated to, preventing sybil attacks. As epochs are used, rate-limiting can be enforced to prevent user abuse.

3.4 Next Steps: Privacy-preserving authentication

The primary advantage of the NEXTLEAP UnlimitID approach over other more complex approaches is that it requires only minor changes to the IdPs from OpenID Connect, and none from the RPs. UnlimitID allows technologically guaranteed privacy-preserving identity in a larger identity eco-system, but it does not work unless the user has a sufficiently large anonymity set, so future research is needed. In particular, there is nothing an IdP can do to prevent a user transfers claims that reveal their identity, such as identity number or uniquely identifying combination of name and birthdate, so that anonymity from the RP is not guaranteed. It would be useful to build an extension of UnlimitID that records the claims sent to a RP over time, and then locally determine if they have violated their anonymity (and a solution for an audit log for claims is presented in Section 4). Also, for many services, real information is not legally required, so another addition would be the automatic creation of "pseudonymous" data, such as random addresses and names for third-party RPs that demand user data.

Another problem is that the authentication to the IdP may not be privacy-preserving either, and different IdPs and RPs may link user data. Currently, users reuse names across sites, allowing names to be linked. Worse, passwords are often re-used to authenticate across sites. The re-use of password allows adversaries to compromise and exfiltrate data from IdPs. A single compromised IdP can

then be used to gather data and abuse multiple RPs, making federated identity very risky. Industry has gathered around the W3C Web Authentication API (previously FIDO), which allows standardized two-factor authentication using a standard challenge-response protocol.[12] The problem with W3C Web Authentication is that while privacy is preserved by having a single master-key derive per-origin keys for the signed proof-of-possession of key material for authentication, the Web Authentication protocol also can violate privacy via device attestation. This can be done as the protocol allows the IdP to query the device identity, which can then in turn be tracked across different origins and be used by both the IdP and RPs to de-anonymize users. Future research is needed to prevent these kinds of attacks on the attestation of the device identity. Comparison is needed to more advanced cryptographic privacy-friendly authentication mechanism such as UProve [22] and Idemix [5] involve selective disclosure credentials, although how to deploy such technologies within decentralized settings without fixed identity providers, seems not to have been fully considered.

4 DECENTRALIZED PKI

Federated identity systems such as OpenID Connect and our privacy-preserving OpenID Connect profile, UnlimitID, untie identity management from key management by simply providing a service that authenticates users. Yet federated identity systems with centralized authentication and storage of claims tend to lock-in users to proprietary eco-systems even if privacy is enforced via technologies such as UnlimitID. Open standards such as SAML and OpenID replicate in the authentication space the assumption of a strong relationship between a user and a hierarchical service provider, school or employer, and as a result do not provide strong decentralization, user mobility, flexibility or privacy to support user-centric authentication and key distribution. So, the next challenge for NEXTLEAP is to allow key material to be bound to identities in a decentralized, privacy-preserving, and secure fashion.

4.1 Problem: Centralized PKI

In order to build privacy and security, usually digital identity is based on binding cryptographic key material to users. Key distribution and management have been the subject of considerable study: the seminal work of Diffie and Hellman introduced public key cryptography as a means to do away with secret key distribution [14]. However, the need to securely associate public keys with individual identities and then allow the correct communication partners to discover this key material was not solved. Approaches involving Public Key Infrastructures (PKI) were deployed to distribute keys of services, and more recently the shortcomings of PKI are addressed by Certificate Transparency solution.[13] These are difficult to extend and scale to associating keys with people rather than services (e.g. there are 6 billion people, but modern PKI systems like "Let's Encrypt" handle around 20 million public key certificates).[14] This "PKI problem" has been one of the most long-standing and hardest problems facing the adoption of public key cryptography. The X.509 certificate infrastructure operates in theory as a centralized key directory, and in practice the X.509 standards were added post-hoc to the centralized domain name system (DNS) and TLS via Certificate Authorities. The X.509 PKI system has been thought of as a failure, as compromises in the certificate authority system have led to fake certificates being issued without being detected. For emerging technologies based on public cryptography, including blockchain technologies and secure messaging, the problem of binding a key to a user has been solved in an ad-hoc manner.

Decentralized alternatives have so far all not reached mass success. The Pretty Good Privacy (PGP) email encryption software relies on a "web-of-trust" instead of a centralized PKI, where users cross sign and validate each others' keys. This approach has well known problems, starting with the fact that trust is not transitive, limiting the reach of cross validation [6], and a large number of usability problems [33] resulting from exposing users to the intimate details key management. Bootstrapping strong identity in pure peer-to-peer settings is an open research problem, with a key challenge is thwarting "sybil attacks" [16] which see a malicious entity creating a number of "fake" users to foil abuse detection and prevention mechanisms, manipulate reputation mechanisms, or win elections. One way is a new field of defenses based on using social relations such as SybilGuard [36], SybilLimit [35] and SybilInfer' [12], or even proof-of-work that most notably forms the sybil protection mechanism behind the Bitcoin digital currency [1]. Yet none of these have seen wide-scale deployment in the case of secure messaging, and problems with spam and abuse will be exacerbated by end-to-end encryption.

4.2 Background: Decentralizing Trust via SDSI

One promising approach is to leverage peer relationships to establish identity and allow users to privately measure the degree of confidence or reputation in other users. Traditional identity mechanisms see authority to designate users flow hierarchically from a higher identity provider. A decentralized approach would allow both use the social graph of peer relationships to combat large-scale sybil attacks, but will also apply ideas from digital currencies to identity where possible to make it very expensive or at least detectable to create multiple fake accounts. However, it is absolutely necessary to provide privacy to the social graph of users while leveraging it to strength security against sybil attacks. Therefore, operations on the social graph, such as adding or removing friends, presence, or establishing reputation metrics will be implemented using privacy-preserving protocols. For example, establishing a number of common friends may be supported through private set cardinality, and establishing presence through private information retrieval as done by the DP5 protocol [3].

It may very well be time for new technologies to revisit the road not taken by the Web: the Simple Distributed Security Infrastructure (SDSI) of Rivest and Lampson. Similar to Bitcoin, principals are identified with keys, in particular with digital signature keys. Also similar to blockchain technologies but in contrast to X.509, each principal is a "certification authority" where "certificates can be created and signed by anyone" [28]. Unlike X.509 PKI, local name-spaces can be created where a principal may associate any arbitrary naming convention with a key. These local names can

[12] https://www.w3.org/TR/webauthn/
[13] http://www.certificate-transparency.org/
[14] https://letsencrypt.org/2017/01/06/le-2016-in-review.html

be explicitly linked across namespaces without centralized permissions. Although the failure of SDSI has been attributed simply to various accidental factors (i.e. it being produced by academics after X.509), there is a outstanding problem with SDSI: Due to its decentralized nature, it was impossible to tell if a key was the latest key, or if a key had been revoked. Key revocation and rotation were considered to be dealt with by "self destruct" or "key update" messages that invalidate keys for a given identifier or bind them to new identifiers, but this work was left undefined [8]. The IETF attempted to standardize SDSI with the parallel work SPKI (Simple Public Key Infrastructure) but it failed in adoption as at the point of its standardization X.509 had already been adopted by the Web.[15]

Traditionally, SDSI can be defined as the following: Given K as the set of public-private key-pairs ($K_i = (\mathsf{pk}_i, \mathsf{sk}_i)$) and a set A of identifiers (usually strings, but any claim in a generic sense is possible), in SDSI a **local name** is a map between an key and one or more identifiers ($K \Rightarrow A$). A certificate is a binding between values and a name, is a signed tuple (K, A, M) where M provides any additional metadata, including but not limited to the validity period of the key. K is the keypair used in signing the certificate. A valuation function for a given identifier A_j is $V(A_j) = K$, a (possibly empty) set of public keys for a given term. Valuation can be defined and implemented as a set of rewrite rules [8]. These rewrite rules allow for a given key the discovery of all associated identifiers for a given key or for a given key to find all associated identifiers and certificates, but these rules simply find all keys. As the set of keys found by $V(A_j)$ could be potentially very large and there is no necessary connection between keys other than possible metadata in M, in practice SDSI did not deal with key handling, discovery, revocation, expiration, and rotation.

SDSI allows the export and linking of local names. For example, (K_1 `Nakamoto`) binds public key K_1 to the identifier `Nakamoto` in a name space (such as Alice's namespace), perhaps including additional information in a certificate such as (K_1, `Nakamoto`, `creator of Bitcoin`, `31-8-2008`). Then Bob can make statements about who Alice thinks Nakamoto is such as (`Alice's Nakamoto Szabo`). SDSI can export (`Bob's Nakamoto`) to refer to an entirely different key (`Bob's Nakamoto Adam`) where (K_2 `Adam`). The identifier `Nakamoto` can be used by any other namespace, such as Eve, to identify another key (K_3 `Nakamoto`) or to associate more identifies with the identifier. Access control is the motivating use-case of SDSI. Traditionally, SDSI defines groups as a set of principals. A group as such does not have its own key, and each member of the group may offer their key as a proof of their membership in the group. Classically groups can be defined via use of the reserved term *Group* and logical *AND*, *OR*, *NOT* as well as *ALL*, *MINUS* (for exclusion of principals) and *ANY*. For example, (`DAO's decision-makers ( Group: ( OR: eth-core-dev ( AND: investor boardmember )))`) defines a group for the DAO's decision-makers where one must be either an Ethereum core developer or an investor and board member to make a decision on a smart contract.

[15] https://www.ietf.org/wg/concluded/spki.html

4.3 Solution: Using Blockchains to Support SDSI

SDSI keychains is the use of SDSI where a simplified blockchain of keys rather than keys are the principal. In terms of blockchain technologies, only a simple authenticated append-only list of data that is authenticated via hash pointers is needed, with the key of each block signing the previous block. Although we are well-aware that blockchains may be used in scenarios where they do not make sense [34], we do want public verifiability of claims with a potentially unlimited number of readers, and each principal having their own chain.This design has been called **claimchain** in a general framework [11], where any data whatsoever may be stored in the blockchain. This use of a claimchain for a *keychain* is similar to the more complex work put forward in CONIKS [21], where the data stored in the claimchain are keys, and each keychain incorporates a Merkle Tree so that verifying the presence of a key in a keychain can be done efficiently. The *head imprint* (i.e. the hash of the latest key) can witness the state of an entire keychain. These head imprints may also be stored inside other keychains so that statements may be made about these heads in order to make statements about another keychain at a given moment in time, allowing the *linked local namespaces* of SDSI. This allows users to make claims about other users and their key material via including their head imprint in the blockhain, allowing users to record on their blockchains "gossip" about their list of contacts by storing the head imprint of other claimchains.

As per deliverables from NEXTLEAP [11], given standard cryptographic definitions, a SDSI tuple $S_i = (K, A, M)$ and its hash $H(S_i) = H(K \parallel A \parallel M)$ and M must include t, time of tuple creation. A **keychain** is an ordered sequence of *keyblocks* $B = \{B_0, B_1, ..., B_n\}$. Each keyblock $B_i = (S_i, P_i, \sigma_i)$ in our chain comprises a tuple S_i that contains claims stored in a Merkle Tree, including possible head imprints ($H(B_{i=max})$) of other keychains (and so the linked local namespaces of SDSI). The block also contains a set of hashes of previous blocks P_i and a signature $\sigma_i = \mathrm{SIG}_{\mathsf{sk}_i}(H(H(S_i) \parallel P_i))$ with sk_i of K_i. Keyblocks so have a global strict ordering as defined by the index i, with the latest keyblock $i = max$ so B_{max} has a unique key K_{max}. All statements A are assumed to apply to the principal defined by B unless the statement is explicitly revoked.

Keys can be revoked by including a revocation statement as part of $M = (K_{new}, R, t)$ where R is revocation made at time t with a new signing key for subsequent keyblocks given by K_{new}. Statements can be revoked by signing new statements later in the blockchain using an explicit revocation statement over the previous statement. The latest key can always be found at the head keyblock as each block has a single signing key, as well as proof of any key rotations and revocations via the aforementioned efficient search for older keys [8]. Statements range over entire histories of keys, rather than just keys, and blocks of statements (such as revocation statements or the addition of new keys) can always be authenticated. We still allow, like the original SDSI, $V(A_j)$ to result in multiple keychains, but they can be compared and a latest key always found by looking at the creation time t of the blocks in case multiple keychains with differing values of B_{max} are found. From this simple mechanism, we believe the entire group-based access control design of SDSI using straightforward logical operators over key material can be

rebuilt using keychains rather than keys. Unlike the original SDSI, given a single key, we can find the latest key and any new identifiers given in the authenticated history of this key. The sharing of data can even be cryptographically done by these keychains. Note that *A* may change over a SDSI-enabled keychain, as would make sense if a user changes their username, as is often done in the change of institutional affiliation, so that the valuation can also resolve in the latest identifier *A*.

4.4 Next Steps: Privacy-Enhanced Blockchains for Identity

While a blockchain may be appealing and this approach allows decentralized cross-referencing blockchains without consensus, the blockchain nonetheless needs to be designed with privacy by default. After all, sharing contacts and gossiping on a blockchain that can be read by anyone would be a privacy disaster. We plan to use encryption of data on the chain and sign using unique deterministic signature in a manner that follows the approach of CONIKS to preserve user privacy. The other large design choice is where to put the keychains. For federated identity, it would make sense that the public keys be stored on the federated cloud identity provider that could also host services such as UnlimitID, and the long-term private key material used by UnlimitID be stored on the client device. Nonetheless, there exists the need to be able to synchronize state between the key-material and blockchains on different devices. For this, NEXTLEAP will lean on the Soledad software developed from U1DB by Ubuntu [31], but with increased security such as key wrapping to secure key material and encrypting the payload so that synchronization between devices does not leak any data.

5 ENCRYPTED MESSAGING

Once an identity system is established, decentralized encrypted messaging is necessary as the next step in order to enable communication between the 'nodes' in the network. All sorts of data may be communicated, from personal data claims to messages such as e-mail.

5.1 Problem: Decentralizing Encrypted Messaging

Perhaps shockingly, the vast majority of data communicated between identity today is not encrypted end-to-end, but only protected by standards such as TLS. Take for example e-mail: The de-facto standards for email confidentiality and integrity is PGP, which has been standardized as OpenPGP [37]. The S/MIME alternative has seen adoption within corporate environments, but little across organizations or between peer users, mostly due to the need for user certificates [27]. Poor client support throughout the 2000s and the subsequent rise in popularity of webmail clients which do not support either PGP or S/MIME, has hampered the deployment of those protocols. Instead, server-to-server encryption has seen some deployment in the form of StartTLS. A key challenge of asynchronous messaging is ensuring forward secrecy, namely preventing subsequent key compromise from affecting past communications. PGP separates encryption and signing keys to facilitate key rotation, but this is neither automated nor a perfect defense. Currently, the Signal protocol seems like the more advanced solution, but identity keys are under the control of a centralized provider.[16]

To confuse matters more, products and services claiming to be privacy-preserving communication already exist as open source, but suffer from not being decentralized. Technically, most of the software codebases are to varying degrees immature and suffer from a number of either critical security and privacy flaws or are limited in their application. A number of private companies offer some kind of privacy-aware hosting solution. For example, the e-mail provider LavaBoom[17] in Germany and the American-founded Protonmail[18] e-mail provider in Switzerland both claim to provide encrypted email for users, guarded by the high legal standards of both countries. However, both products suffer from a number of critical flaws. Without being open source, the community cannot verify the code actually is privacy-preserving and may not have a secret "back-door" (such as an ability to decrypt e-mail by the provider without the proper authorization from the user or the government) placed in the code.

Unlike email messaging, the vast majority of "post-email" secure messaging programs like Signal can not interoperate even if the same fundamental protocol is used across different services such as Google Allo and WhatsApp: WhatsApp users cannot chat with Google users, and so on. This offers a stark contrast to the email model, where any email service can openly communicate with another (Gmail to Outlook, and so on). Although XMPP does attempt to address this issue of centralization and while Off-the-Record messaging has existed as a de-facto standard for encrypted chat in XMPP since 2004 [4], Off-the-Record messaging does not allow asynchronous messaging but only synchronous messaging. Recent work on OMEMO does allow Signal Protocol-style messaging in a decentralized manner via XMPP.[19]. Yet usage of OMEMO has been restricted to low volume clients compared to Signal or WhatsApp, and the number of Jabber servers themselves is quite low. Importantly, neither federated XMPP or centralized Signal-based silos prevent the social graph of the users from being mapped.

While intuitively many people claim that decentralization helps privacy, in the case of a powerful adversary with the capability to observe all network traffic, de-anonymization and surveillance is actually easier on decentralized networks since by nature the traffic between different nodes in a decentralized network reveals the social graph of all the users [23]. For example, if Alice and Bob try to communicate using a decentralized network via their own self-controlled "nodes," any third party Eve can simply watch the traffic in and out of both nodes and decipher by virtue of timing information the strength of their social connection (now often termed "metadata"), even if the content of the message is encrypted. The operations around messaging in a decentralized system, such as presence and status updates can be protected to avoid leaking the social graph of users to untrusted or semi-trusted intermediaries in the context of a decentralized protocol [3]. However, there are many ways messaging metadata may be protected. The operations around messaging, such as presence and simple status can be protected to avoid leaking the social graph of users to untrusted

[16] https://www.signal.org/docs/specifications/x3dh/
[17] http://www.lavaboom.com
[18] https://protonmail.ch
[19] https://xmpp.org/extensions/inbox/omemo.html

or semi-trusted intermediaries in the context of a decentralized protocol. Techniques like mix networking may hold the key to make decentralized systems privacy-preserving and anonymous to outside observers. In mix networking, padding is added to messages and messages can be sent at constant time intervals, such as done in the Drac system [10]. Yet existing mix-networking systems are traditionally static, and it is unclear how to decentralize mix networking-based designs to handle a dynamic and open number of mix nodes while defending against the degradation of anonymity properties.

5.2 Next Steps: Extending Signal Protocol with Identity and Groups

As discussed earlier, the most relevant project is Signal[20] as the same underlying Signal protocol is now deployed also by WhatsApp to over a billion users. Besides traditional confidentiality properties traditional in encrypted e-mail, there is the need to provide forward secrecy properties to asynchronous messaging, a feature usually reserved for synchronous ones. Currently, this is achieved through a centralized user representative, such as the centralized Signal server in the cloud, acting on behalf of the user to facilitate key management operations relating to key updating. The reason Signal has not decentralized is likely due to the fact that distributing such functionality over diverse services and peers to ensure no central point of failure is difficult at best. Yet in its current cryptographic design the Signal protocol also suffers from a number of fundamental issues, including but not limited to the lack of *transcript consistency* and no notion of group messaging with forward secrecy. Rather than group messages, until recently messages were just resent from end-point to end-point, which leads messages often in a group to arrive out of order. More recently, a single long-term symmetric key is established but the property of forward secrecy is lost and group management is no longer cryptographically enforced. The reason for this is that the Signal Protocol is designed as a simple authenticated key-exchange protocol, but lacks any notion of identity to bind the key derivation ratchets in Signal to a given identity. By adding identity to the Signal Protocol via the keychains in Section 4, NEXTLEAP can use the blockchain of key material to track per-conversation keys in a group messaging scenario, and thus make sure that messages arrive in order. In essence, the group can establish its own collective keychain that references each member of the group's keychain, and possibly using the mechanisms developed in SDSI for access control if needed.

The metadata problem is harder to solve than simply using Tor [15], and the preference of NEXTLEAP is to use mix-networking. Although sending the messages via Tor (The Onion Router) does help against certain classes of adversaries, Tor is well-known to not be resistant to global passive adversaries [13]. A better approach for a decentralized system based on personal data transfer and messages between users is to use a layer of mix-networking that can defend against a global passive adversaries, hiding the social graph of messages including size and time of message. Although mix-networking has historically been too high latency, NEXTLEAP believes that it is likely now possible to create newer, low latency mix networking systems that use a reasonable amount of dummy traffic modelled from the distribution of real traffic from a service provider or user. Currently, NEXTLEAP is working with the European Commission PANORAMIX project to deliver a generic mix networking framework, similar as Tor has done for onion-routing. The plan is to start operating as a stop-and-go mix with six mix-networking nodes in different European jurisdictions that could then in turn support multiple decentralized providers, and use this as a generic infrastructure for messaging and claim transfer in identity systems.

[20] https://whispersystems.org

6 PRIVACY-ENHANCED ANALYTICS

Moving into privacy-enhanced and decentralized environments will economically be unfeasible unless there exist analytics based on machine-learning in order to improve the performance of the system and understand user behavior, but currently this is done in a very privacy-invasive manner. Still, it seems that the benefits of data-mining should be possible in a privacy-respecting framework.

6.1 Problem: Privacy-aware and Decentralized Data-mining

Private Information Retrieval (PIR) is a family of techniques that allow users to access information online without revealing which records or queries they are performing, usually by running "fake" queries so that the real query is statistically unobservable by the server with the data or by decentralizing the servers themselves so that no server has all the data necessary to reveal the full query. PIR has been proposed as a means to achieving private communications, as in the Pynchon Gate system [29]. However, PIR based systems have also been proposed for implementing both privacy preserving relational databases [24] and the privacy-preserving presence systems necessary for secure messaging [3]. Any decentralized reputation mechanism will need mechanisms to access peer information privately, and we propose PIR will be a key infrastructure to facilitate machine-learning without centralization. In cases where users do need to access their own storage privately, the Oblivious RAM primitive may be used instead [26]. While the basic mathematics of both PIR and ORAM are well understood it would be pioneering their use in protecting decentralized system and their widespread deployment.

While both PIR and ORAM may be used to access information, solutions need to be found for generating aggregate statistics, and privately eliciting and aggregating the "wisdom of crowds" in decision making and reputation. These systems are special instances of the family of Secure Multi-party Computation protocols such as those implemented by the state of the art SPDZ [9] protocols or the ShareMind compiler [2]. Aggregating distributed data in a private manner requires users to be certain that their private inputs will not be easily deduced from either the mechanism or the final result of computations. This is not trivial to achieve since the utility of the system relies on learning "something" from individual user inputs. The state of the art privacy definition for reasoning about the degradation of privacy in such contexts is differential privacy [17].

6.2 Next Steps: Generalizing PrivEx

This part of the research is still very much in work, but the simplest approach is simply to do aggregate data collection and simple statistics over multiple nodes in a decentralized system. This is currently being done by the PrivEx system over the network traffic information of Tor [18]. The approach still needs to be generalized for generic anonymized data collection over aggregates of messages and personal data. NEXTLEAP plans techniques from private information retrieval will be adapted to a dynamic decentralized setting to support look ups, and differentially private mechanisms will ensure the result of these look ups leaks little information about individual preferences or opinions. No previous system integrates privacy-enhanced systems private information retrieval, with advanced privacy protocols, and leveraging such an approach over real-world messaging and personal data would be ground-breaking.

7 CONCLUSION

Are decentralized identity solutions for the cloud that help privacy even possible? NEXTLEAP is working on developing common protocols for an infrastructure that encompasses basic services such as authentication, authorization, and messaging, as well as allowing the necessary analytics to be done in a privacy-preserving manner. As a European project, we believe these principles are well-suited for Europe, as Europe is by nature a federation of different nations and principalities, it makes even more sense to have an approach that emphasizes decentralization while letting the "network effect" of Europe still be taken advantage of, and we believe this model could easily be exported to other regions interested in technological sovereignty and data protection such as South America, Asia, and the post-Snowden United States. Furthermore, due to this demand, there is strong reason to believe that a number of new models of social innovation exist that would support the possibility of a decentralized approach in reaching wide adoption. The decentralized infrastructure pioneered by NEXTLEAP could be ran by volunteers, but also by corporations, non-profits, governments, and activist groups. Yet there are a large number of research problems, and we hope that by tackling a few of the more foundational ones, we can make the vision of a private and secure decentralized Internet possible.

8 ACKNOWLEDGMENTS

Much of the text comes from our original proposal to start NEXTLEAP. George Danezis provided much of the text that have been re-used and other partners from NEXTLEAP also provided editorial assistance, although all mistakes are my sole responsibility.

REFERENCES

[1] Adam Back, Ulf Möller, and Anton Stiglic. 2001. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Information Hiding*. Springer, 245–257.

[2] Dan Bogdanov, Sven Laur, and Jan Willemson. 2008. Sharemind: A framework for fast privacy-preserving computations. In *Computer Security-ESORICS 2008*. Springer, 192–206.

[3] Nikita Borisov, George Danezis, and Ian Goldberg. 2015. DP5: A private presence service. *Proceedings on Privacy Enhancing Technologies* 2015, 2 (2015), 4–24.

[4] Nikita Borisov, Ian Goldberg, and Eric A. Brewer. 2004. Off-the-record communication, or, why not to use PGP.. In *WPES*, Vijay Atluri, Paul F. Syverson, and Sabrina De Capitani di Vimercati (Eds.). ACM, 77–84.

[5] Jan Camenisch and Els Van Herreweghen. 2002. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 21–30.

[6] Germano Caronni. 2000. Walking the web of trust. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proeedings. IEEE 9th International Workshops on*. IEEE, 153–158.

[7] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. 2014. Algebraic MACs and keyed-verification anonymous credentials. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 1205–1216.

[8] Dwaine Clarke, Jean-Emile Elien, Carl Ellison, Matt Fredette, Alexander Morcos, and Ronald L Rivest. 2001. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security* 9, 4 (2001), 285–322.

[9] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P Smart. 2013. Practical covertly secure MPC for dishonest majority–or: Breaking the SPDZ limits. In *Computer Security–ESORICS 2013*. Springer, 1–18.

[10] George Danezis, Claudia Diaz, Carmela Troncoso, and Ben Laurie. 2010. Drac: An Architecture for Anonymous Low-Volume Communications. In *Privacy Enhancing Technologies*, Mikhail Atallah and Nicholas Hopper (Eds.). Lecture Notes in Computer Science, Vol. 6205. Springer Berlin / Heidelberg, 202–219.

[11] George Danezis, Bogdan Kulynych, Carmela Troncoso, and Marios Isaakides. 2016. ClaimChains: A Decentralized Identity System based on hash chains. (2016).

[12] George Danezis and Prateek Mittal. 2009. SybilInfer: Detecting Sybil Nodes using Social Networks.. In *NDSS*. San Diego, CA.

[13] George Danezis and Andrei Serjantov. 2004. Statistical Disclosure or Intersection Attacks on Anonymity Systems. In *6th International Workshop on Information Hiding (Lecture Notes in Computer Science)*, Jessica J. Fridrich (Ed.), Vol. 3200. Springer, 293–308.

[14] Whitfield Diffie and Martin E Hellman. 1976. New directions in cryptography. *Information Theory, IEEE Transactions on* 22, 6 (1976), 644–654.

[15] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*, Matt Blaze (Ed.). USENIX, 303–320.

[16] John R. Douceur. 2002. The Sybil Attack.. In *IPTPS (Lecture Notes in Computer Science)*, Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron (Eds.), Vol. 2429. Springer, 251–260.

[17] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Theory and applications of models of computation*. Springer, 1–19.

[18] Tariq Elahi, George Danezis, and Ian Goldberg. 2014. Privex: Private collection of traffic statistics for anonymous communication networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1068–1079.

[19] Marios Isaakidis, Harry Halpin, and George Danezis. 2016. UnlimitID: Privacy-Preserving Federated Identity Management using Algebraic MACs. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*. ACM, 139–142.

[20] Leslie Lamport. 1978. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM* 21, 7 (1978).

[21] Marcela S. Melara, Aaron Blankstein, Joseph Bonneau, Edward W. Felten, and Michael J. Freedman. 2015. CONIKS: Bringing Key Transparency to End Users. In *24th USENIX Security Symposium, USENIX Security 15*, Jaeyeon Jung and Thorsten Holz (Eds.). USENIX Association, 383–398.

[22] Wojciech Mostowski and Pim Vullers. 2011. Efficient U-Prove implementation for anonymous credentials on smart cards. In *Security and Privacy in Communication Networks*. Springer, 243–260.

[23] Arvind Narayanan, Vincent Toubiana, Solon Barocas, Helen Nissenbaum, and Dan Boneh. 2012. A Critical Look at Decentralized Personal Data Architectures. *CoRR* abs/1202.4503 (2012).

[24] Femi Olumofin and Ian Goldberg. 2010. Privacy-preserving queries over relational databases. In *Privacy enhancing technologies*. Springer, 75–92.

[25] A. Pfitzmann and M. Köhntopp. 2001. Anonymity, unobservability, and pseudeonymity - a proposal for terminology. *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability table of contents* (2001), 1–9.

[26] Benny Pinkas and Tzachy Reinman. 2010. Oblivious RAM revisited. In *Advances in Cryptology–CRYPTO 2010*. Springer, 502–519.

[27] Blake Ramsdell. 2004. Secure/multipurpose internet mail extensions (S/MIME) version 3.1 message specification. (2004).

[28] Ronald L Rivest and Butler Lampson. 1996. SDSI-A Simple Distributed Security Infrastructure. CRYPTO. http://people.csail.mit.edu/rivest/sdsi10.html.

[29] Len Sassaman, Bram Cohen, and Nick Mathewson. 2005. The pynchon gate: A secure method of pseudonymous mail retrieval. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*. ACM, 1–9.

[30] Andrei Serjantov and George Danezis. 2002. Towards an Information Theoretic Metric for Anonymity. In *Designing Privacy Enhancing Technologies, Proceedings of PET'02*. Springer-Verlag, LNCS 2482, 41–53.

[31] Elijah Sparrow, Harry Halpin, Kali Kaneko, and Ruben Pollan. 2016. LEAP: A next-generation client VPN and encrypted email provider. In *International Conference on Cryptology and Network Security*. Springer, 176–191.

[32] Carmela Troncoso, George Danezis, Marios Isaakidis, and Harry Halpin. 2017. Systematizing Decentralization and Privacy: Lessons from 15 years of research and deployments. *CoRR* abs/1704.08065 (2017). http://arxiv.org/abs/1704.08065

[33] Alma Whitten and J Doug Tygar. 1999. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0.. In *Usenix Security*, Vol. 1999.

[34] Karl Wst and Arthur Gervais. 2017. Do you need a Blockchain? Cryptology ePrint Archive, Report 2017/375. (2017). http://eprint.iacr.org/2017/375.

[35] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. 2008. Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 3–17.

[36] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. 2006. Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review* 36, 4 (2006), 267–278.

[37] Philip Zimmermann. 1995. Pretty good privacy: public key encryption for the masses. In *Building in big brother*. Springer-Verlag New York, Inc., 93–107.