Identity System Essentials

Samuel M. Smith Ph.D.

and Dmitry Khovratovich Ph.D.

Evernym

29 March 2016

Contents

Overview

The purpose of this white paper is to describe the essential characteristics of an identity system that provides sovereignty, security and privacy. Here the meaning of identity is derived from the characteristics of the identity system, that is, what the identity system provides. Instead of defining identity a priori, this white paper describes an identity system and then defines identity within the context of that identity system. Many of the features of the identity system has been influenced and inspired other proposed systems such as Open Reputation.[1] [2] [3] This paper argues that an identity system that simultaneously provides a high degrees of sovereignty, security and privacy is best obtained via an open platform that employs distributed consensus protocols and modern cryptographic techniques.

Interaction Between Entities

One essential purpose of the identity system is to enable entities to interact with each other. An entity may be a person, a group of people, an organization, or a non-human computational agent. Each entity that uses the identity system is uniquely identifiable with respect to all the other entities within the identity system but may be only pseudonymously or even anonymously identifiable by other entities even those involved in an interaction with a given entity. More formally the interactions supported by the identity system are limited to a well defined set of transactions. Indeed, the identity system is transactional in nature. The benefits of the identity system are exemplified by the associated characteristics of the supported transactions.

The identity system provides a necessary but not always sufficient enabling framework for transactions between entities. By analogy, the identity system is the hat rack for transactions that are the supported hats. Transactions might of different types and for different purposes. For example, transactions might be social such as viewing or editing the content of a website or blog or exchanging text messages or email. Transactions might be financial such as purchasing items or applying for a loan. Finally, transactions might be legal such as proving one is of drinking age or is the owner of a parcel of land.
These transactions are primarily but not exclusively executed with the aid of networked computing devices. This is both a bane and a boon. A bane because digital networked transactions are potentially vulnerable to exploit both directly through theft, forgery or other fraud or indirectly through unintended disclosure, data aggregation or surveillance. A boon because digital networks can not only enhance transactions by reducing costs and broadening the reach of entities that might beneficially engage in transactions but also employ modern cryptographic techniques that can be used to protect transactions from exploits. Indeed it is the latter that is the motivation for the identity system, that is, enable entities to participate in more beneficial transactions that are also secure and private.

Attribute Exchange

In general, each party (entity) to a transaction may require information about the other party before proceeding with the transaction. The primary role of the identity system is to facilitate the required information exchange between the parties about each other so that the transaction may proceed. The type of information exchanged might include identifiers, credentials, tokens, capabilities, behavior, attributes,

reputation, biometrics, or other personally identifiable information. At this level of the description all the the information above can be generically grouped into either identifiers or attributes. In this context identifiers have a special role. A given set of attributes provided in an information exchange about an entity is labeled with an identifier specific to that set of attributes. The attributes labeled with a given identifier are all the other information needed to facilitate that entity's side of the transaction and may include what might otherwise be called identifiers such as an entity's name, address, or age. Each party in the transaction may, and typically will, request that some attributes (such as driver licence) are certified by a certain institution. In this case the certification is presented to the other party alongside the transaction details.

Least Disclosure

The guiding principle behind the selection of attributes to enable a transaction is called least disclosure, that is, the identity system should disclose the minimum amount of information about a given entity (party) needed to facilitate the transaction and no more. The reason for following this principle is to both maintain privacy and mitigate the potential side effects of disclosure. Privacy will be discussed in more detail later.

Identifiers

As mentioned previously, an identifier is typically encoded as a string of characters, that is uniquely associated with a subset of attributes about an entity. The associated subset of attributes may be provided to facilitate a transaction or transactions.
A public identifier, which corresponds to a (potentially confidential) set of attributes, is called a cryptonym. It can be implemented as a keyed hash of the attribute set, or as a (hash of) the public key associated with the set. Cryptonyms are primary identifiers.

Identities

An identity is defined as the combination of at least one cryptonymous identifier and a subset of attributes for an entity. An identity may have multiple cryptonymous identifiers but one of them must be primary, i.e. associated with this identifier and no other one.Thus an identity may be expressed as a data structure or record in a database indexed by its primary identifier(s). Each entity must have at least one identity. Identity can be kept private thanks to cryptonyms. If a cryptonym is used multiple times, it is essentially a pseudonym used to address an unknown identity (but maybe with certain known attributes). A one-time cryptonym provides full anonymity, but there can be other mechanisms to provide anonymity using modern cryptography.

Aliases

In addition to cryptonyms, an identity may have other identifiers called aliases. An alias or aka is a human friendly globally unique string of text that may be associated with an identity in addition to its primary cryptonym. It is a secondary identifier. An alias might be a human readable globally unique identifier that is uniquely associated with the identity. The uniqueness of an alias may be maintained by an external

authority such as the internet domain name system or the phone number system. An identity may have multiple aliases associated with it. The purposes of an alias is to provide a human friendly way of interacting with identities. For example, a URI is a globally unique uniform resource identifier, it may be an alias or used to generate an alias using a naming convention. Other types of aliased or secondary identifiers include GPS coordinates, email addresses, street addresses, domain names, phone numbers, and social media profiles. Several open standard protocols are working to provide globally unique identifiers that may be controlled by individuals such as XDI[4] and WebDHT[5].

Identity Graph

Because identities include subsets of attributes used to facilitate transactions, an entity may have multiple identities arranged in hierarchies of groupings of the associated attributes. The identities together with the links between them form the identity graph.The connections (edges) between identities in an identity graph is not merely limited to groupings of attributes but could be based on associations derived from usage of each identity in transactions with other entities. The identity graph may also have references to the details of that associated transaction as well as the terms of use of the disclosure of the associated attributes.

Hierarchical Deterministic Keychains

If cryptonyms are implemented as public keys, an identity graph could result in a large number of private keys used to produce cryptonym-verifiable signatures. One solution to this problem is to use hierarchical deterministic keychains[8]. In a hierarchical deterministic keychain only one private key per graph needs to be retained. This is the root private key which has an associated root public key. All the other key pairs are generated deterministically from the root key pair. The public key or cryptonym of a derived key can then be used to safely and securely generate the associated private key. The private key is needed for these operations but can be generated on demand by the holder of the root private key and no one else. This enables fine grained granularity of identities without the associated burden of retaining a large number of private keys.

Sovereignty

Another guiding principle of the identity system is that entities should own their identities, that is, have control over their own identity graphs. Another way of saying this is that an entity should have sovereignty over its identities. An entity is a sovereign source of identity when that entity is in control of its own identity graph including the creation, modification, storage, distribution, disclosure, and destruction of the associated identifiers and attributes. A sovereign identity system supports this capability.

This does not preclude entities from creating and maintaining identity graphs of other entities. Indeed because entity interaction is transactional, entities may need to maintain identity graph profiles of the other entities with whom they interact. But entities should be enabled to maintain an identity graph of their own identities and be an authoritative source of identities and the associated identifiers and attributes.

Control over Attributes and Identifiers

A key part of sovereignty is control over the identifiers not just the attributes. Identifiers are the foundation of transactions. In non-anonymous systems an entity may build value via reputable participation in transactions with other entities. The links, references, history, reputation and interconnectedness are all tied to the associated identifiers. Thus if an entity does not control its own identifiers it may no longer have access to the value created over time with the connected entities. It is not enough to merely have control over the management of the attributes but sovereignty also requires control over the identifiers and hence control of its whole identity graph.

For example, in the days before portable phone numbers, if a user wished to switch telephone providers, they could only do so by switching to a different phone number. This meant that all of their contacts had to be notified and update their address books to accept the new phone number. If a user did not have a complete record of everyone with whom they had shared their phone number they would not be able to notify them to change their records. Consequently the user did not have sovereignty of the identity associated with the phone number identifier. Portable phone numbers enable sovereignty over the identifier. In many cases, the phone company also provides an address book for the user that contained contacts of the user. Switching phone companies requires that the user export their contacts and then reimport them to their new phone provider. Consequently, the degree of sovereignty over some of the attributes (address book) was incomplete.

Indeed many of the most popular identity providers, such as FaceBook or LinkedIn or Google or Amazon or eBay support very low levels of sovereignty over the associated identities. The user's access to both the identifiers and attributes (content) can be disabled at any time at the discretion of the identity provider. The user then loses all value associated with their identity. The user's identities are locked up in a silo out of their control.

Moreover, as numerous publicized cases have shown, centralized identity providers are both susceptible and vulnerable to exploit for identity theft because of the attractive value of the centralized store of identities and the common mode failure weaknesses of their firewall based security. These identity providers extract value from the content of the data stored with the identities. In order to extract that value, they must have access to unencrypted copies of the data. This makes an attractive target for an attacker because the attacker merely has to breach the firewall to gain access to all the data.

Disadvantages of Personal Data Store

One approach to sovereign identity is to enable the user to be their own identity provider with their own personal data store and identity server. Examples include Open Mustard Seed[9] and OpenPDS/SA[10][11]. The problem with a personal data store/identity server (PDS/IS) is that it places a burden on the entity for hosting, maintaining, and securing their PDS/IS. This can be too much of a burden for private individuals and small organizations, thereby preventing adoption. In addition, although on average a PDS/IS is less attractive as a target for exploit because of the relative low value of the data, the inability of an individual to follow best practices for security of their identity server makes them extremely vulnerable to exploit.

Disadvantages of IAAS

Another approach to sovereign identity is to use an identity as a service (IAAS) provider, wherein each entity contracts with a service that provides identities but under the control of the client. The identity service acts as an agent or proxy for the client entity thereby relieving the client user of the burden of hosting, maintaining, and securing their identities including the attribute data store and identifiers. In order that the served identities to be self sovereign, the identifiers and data must be portable to other IAAS providers, that is, the identity provider must be replaceable. This is an important characteristic and implies at the very least that the identity service have an open standard interface if not an open source platform (cf. BYU Domains[12] [13]built on CPanel[14]).

Many IAAS providers silo their hosted identities behind identifiers that are proprietary to that provider, thus these identities are not portable. This is often the case for user managed access (UMA)[15] identity providers. The user may be able to set rules and policy for how their identities are used by the identity provider, but the user does not own the identifiers used to access or connect with their identities. So even though the user may be able to export their attributes and reimport them into a different identity provider, the user is not sovereign because the user does not control the identifiers. This is analogous to the telephone number portability example described earlier.

A centralized IAAS provider, however, even if it supports sovereign source identities, is still susceptible to attack because of the attractive value of all the identities it hosts and is still vulnerable to exploit because of the common mode failure of firewall based security. Indeed, this is the crucial problem of a sovereign identity system, that is, how to provide both sovereign and secure identities.

Security

As interactions become increasingly digital, the risk of loss due to identity fraud becomes greater. Therefore, highly secure identity related transactions becomes more important. One useful way of characterizing the security of an identity system is its survivability, that is, how likely is it that the system will survive a given security attack or exploit. Survivability can be broken up into three aspects, these are, susceptibility, vulnerability, and recoverability. Susceptibility is the likelihood that a given system will be the target of an attack. Factors that affect susceptibility include the reward for a successful attack versus the cost of mounting the attack and the degree of visibility of the system. Vulnerability is the likelihood of an attack succeeding. Factors that affect vulnerability include the degree of difficulty of an exploit given the strength of a particular cryptographic system and the failure modes. For example, a system might have multiple layers of security, requiring a separate costly exploit to breach each one. But if there is a back door or other common mode failure that allows all the layers to be breached with a single exploit then the degree of difficulty might be very low given the common mode failure. Recoverability is the likelihood of repair to mitigate any damage or loss in the event that an attack succeeds. Recoverability might include the ability to limit liability to fraudulent credit charges or the ability to quickly disable bank accounts. In many cases where the harm is the disclosure of sensitive information about an entity, recoverability is minimal, making it all the more important that the system have low susceptibility and low vulnerability.
An alternative way to characterize the security the identity system is to view the critical assets of the system from the perspective of the core principles: confidentiality, integrity, or availability. Then we describe the security mechanisms that should be implemented to provide these protection types and measure threats, risks, and vulnerabilities that can compromise the core goals.
The primary assets in the identity system are user attributes, identifiers/cryptonyms, transactions, and metadata. Some of them are confidential (attributes), for some the integrity is the primary goal

(transactions), etc. Threats, for example, are disclosure of user data, integrity violation, denial-of-service attacks on users and services. Controls and mechanisms may include public- and symmetric-key encryption for confidentiality, distributed database such as blockchain for integrity and availability.

Modern Cryptography

Modern cryptography provides a number of fast, secure, and scalable protocols to satisfy the confidentiality and integrity goals. Basic confidentiality is provided by symmetric encryption, whereas more esoteric scenarios where the encryption, storage, search, and processing of the data is delegated or shared between parties are supplied by more advanced concepts such as attribute-based encryption. The concrete tools depend on the functionality required from the identity system.
Integrity and authentication mechanisms are provided by hash functions coupled with digital signatures. Basic signatures can certify that the (private key) owner signed the document, whereas more advanced protocols allow proving more sophisticated statements about the underlying document, such as "age>=21" using tools such as zero-knowledge proofs, pairing-based cryptography, and several others (a detailed list of features provided by modern cryptographic protocols is beyond the scope of this document). Current implementations can be both reasonably fast and secure[16] [17][18]. An important feature of the identity system is that it have convenient and effective private key management which includes key generation, sharing, storage, hiding, recovery and revocation. Hierarchical deterministic keychains described previously provide some convenience. Other techniques include multi-signature protocols for signing approval and for key recovery as well as key hiding protocols for storage.
In a system with centralized identity providers the latter are the most natural attack targets as multiple user accounts can be compromised in a single exposure. Even if the best practices are used for secure transmission and storage protocols, it is the management of both transactions and the associated keys that may become the source of common mode failures and thereby the weakest part of the system. The users must also trust the competence and good will of the managers of a centralized identity system.

Diffuse Trust with BFT-based blockchains

One new approach to achieve transaction integrity and availability is a diffuse trust distributed consensus system. This approach has been popularized with crypto-currencies such as BitCoin and it often referred to as blockchain technology[19]. The core idea is that a ledger of transactions is managed by a disparate set of hosts that use distributed consensus algorithms to approve and validate entries in the ledger. These distributed hosts must come to a majority consensus about any entries in the ledger. The consensus yields the transaction integrity, whereas a large number of hosts provide availability.

The distributed consensus also solves the problem of trust, as the peers typically do not trust each other. In this concept he trust is effectively diffused across all the members of the consensus pool so that the actions of some do not invalidate the group consensus. As a simplistic generalization, the ledger is trustworthy if the majority of the members in the consensus pool are not colluding to defraud.

The entries in the ledger are linked together with a chain of hashes which makes any fraudulent changes to the ledger detectable. The ledger can be independently verified by following the chain. Because changes to the ledger are governed by a majority or super majority consensus of the hosts, an attack on a few of them does not result in a change to the ledger. Indeed, as long as only a minority of the hosts are ever exploited, the ledger is not compromised.

Various distributed consensus algorithms have been developed with varying performance and governance characteristics. These include Proof of Work (POW), Proof of Stake (POS), Byzantine Agreement (BA), and hybrid protocols[20] [21] [22] [23] [24] [25] [26] [27] [28] [29]. The BA protocols have low latency and are able to support up to thousands of transactions per second while being Byzantine fault tolerant (BFT). Byzantine faults include not only failures but malicious behavior of hosts that may attempt to interfere with the formation of a consensus. Byzantine agreement protocols require that members of the consensus pool be selected through a qualification or permissioning process. Thus blockchain ledgers managed by BA protocols are sometimes called permissioned blockchains[30]. Security of a given ledger can be furthered enhanced by periodically anchoring one ledger to one or more other ledgers[31].

Hybrid Ledgers

A distributed consensus pool using a BA protocol may not only trustworthily manage a blockchain ledger but may be used to trustworthily manage entries in a database or a keychain or other distributed consensus pools in a nested fashion or even hybrid ledgers and databases. For example, a blockchain ledger can be used to store an audit trail of transactions for a database where the audit trail includes metadata about the transactions such as identifiers and cryptographic hashes but does not include the actual data. The actual data is stored in a distributed hash table such as Kademlia[32] or distributed datastore such as StorJ[33] or MaidSafe[34] or IPFS[35]. A ledger is only strictly necessary when the transaction data might be inconsistent with previously approved transactions such as payment transaction (where double-spending is a concern) or voting (where a vote must be one-time)Using diffuse trust distributed consensus enables the decentralized and highly secure management of many parts of the identity system thereby providing both sovereign and secure identities.

BlockChain Based Identity

Several ongoing efforts are leveraging distributed consensus or blockchain technology to provide components of identity systems that reside outside of centralized fire walled silos. These include identifier registries such as NameCoin[36], DNSChain[37], and BlockChainID[38]. OneName uses BlockChainID which is more than just an identifier registry; it also includes a distributed hash table database (Kademlia) for storing attributes. These are all based on the BitCoin blockchain. While the BitCoin blockchain is well known and very secure with thousands of host nodes, it suffers from very high latency (1 hour) and very low transaction rates (less than 7 per second). An identity system using a Byzantine agreement protocol could achieve much higher performance. The distributed consensus pool manages a ledger of meta-data about transactions to a distributed identity attribute database. It also manages meta-data about transactions between entities. With a distributed consensus pool, the pool becomes a trusted third party to every transaction, effectively serving as a notary. This allows for transactions with varying degrees of privacy and security depending on how the notary is used including triple signed transaction receipts[39].

Privacy

Privacy is a complex concept with no simple definition, indeed there are many types of privacy and types of privacy violations[40] [41] [42] [43] [44]. What all these types of privacy have in common is that disclosure of information that may be linked to an entity may lead to less privacy. So to preserve privacy requires the

ability to control the degree, timing, and audience of disclosure. Thus we define privacy in terms of the degree of associable information disclosure about an entity. The degree of disclosure or provable association of an identity with a real entity may vary from complete anonymity with no provable association and no public disclosure to no anonymity with complete provable association and full public disclosure.

Many of the negative consequences of privacy violation come not from the use of the information by those parties directly involved in a transaction but from the use of the associated information by uninvolved third parties. Moreover, a third party could collect metadata by aggregating, correlating, and tracking the information disclosed in multiple interactions with multiple second parties over time to infer additional private information not directly disclosed in any of the transactions[45].

We suggest identities being private by default. The associated parties in a transaction may choose to make their identities public but they must opt into that publicity. Identity transactions can be used to create reflexive profiles of identities, that is, the act of originating a transaction with another entity to gather attributes also associates those attributes with the originator as a recipient. Each side of a transaction may maintain an identity graph of the other side not just their own side. This allows an entity to track its own disclosures and also track information disclosed to it.

Three Degrees of Privacy for Transactions

We define three basic degrees of privacy for transactions, they are, full public, semi-private, and full private. Each transaction has metadata that is always public such as a transaction ID, hash, cryptonymous identifiers, timestamp, and classification information. The cryptonyms may be associated with identities that have public identifiers or alias or may be associated with anonymous identities of varying degree. In addition the attributes exchanged in the transaction may be encrypted or unencrypted.

In a full public transaction, the attributes section is unencrypted and the identifiers are public. In a semi-private transaction either the attributes are encrypted but at least one of the identifiers is public or the attributes are unencrypted but at least one of the identifiers is anonymous. In a full private transaction the attributes are encrypted and all the identifiers are anonymous.

Semi-private transaction allow interactions where a public identities can be used to initiate an otherwise private conversation. This is like a telephone conversation where the parties involved in the call, given by their phone numbers, are known but the actual spoken conversation over the phone is not. Whereas fully private transactions use anonymous identifiers so the conversation cannot be traced back to a given entity.

Group Privacy

Another approach to privacy is called group or class based privacy. In group privacy entities are members of a group or class of entities that share common attributes. Interactions with a member of the group as identified by a group identifier provide a form of group anonymity. This approach diffuses the identity across the members of the group thereby reducing the information content usable by a third party and therefore effectively limiting the degree of privacy disclosure.

Group may have one or more signature or encryption keys (cf. the approach taken by the EPID standard (Enhanced Privacy ID) for internet of things (IoT) identity[46]). The group policy should determine how

many group members (one, a few, or all) have to collude to use a certain key. The service may have policies for inclusion and revocation of membership in the group as well as permission for disclosure to external parties. Groups may be nested. The keys themselves may be blinded such that the service does not have direct access to the bare keys. The keys may reside only in the private keychains of the group members.

Forgetting

One use of identity is to build a reputation for a given identity, based on the behavior of the associated entity when using that identity (identifier). This reputation may be good or bad. Over time the behavior of the entity may change and the reputation may no longer be accurate. Consequently there is a need for some form of forgiveness over time or the ability to "forget" transactions that are no longer relevant to the reputation. In some countries there are laws that specify a right to be forgotten in the sense that entities may request that data about them be deleted over time. Related to the forgettability is the ability of redress to repair errant information[47]. There are three ways that forgettability can be provided by the identity system. The proposed identity system could eventually support all three methods of forgetting.

The first way is to just to delete the encryption keys. Once all copies of the encryption key is deleted there is no way to decrypt the data. Another approach is to make the data non-retrievable by deleting certain number of shares in threshold-storage protocols. The users might need to explicitly waive the right to have the encrypted data deleted and accept key deletion as sufficient for forgetting.

Another way is to use time limited ledgers. The transaction ledgers can be checkpointed every certain period of time so that the ledger is essentially closed out and started over. Current transactions are brought forward and entered anew in the ledger but stale transactions are deleted. As mentioned previously, a distributed consensus pool can act as a trusted third party enabling triple signed transaction receipts. In a triple signed transaction, both sides of the exchange and the notary all sign the transaction. A receipt can be generated that is proof of the transaction. The transaction can be deleted from the ledger, but each party can keep a copy of the receipt that proves all three parties agreed to the transaction. This allows transactions to be "forgotten" from the public ledger.

The final way is to use a mutable database in concert with the metadata transaction ledger. The consensus pool manages the transactions in the database including change or delete transactions. The metadata includes a full audit trail including the delete transactions. This is analogous to accounting systems where transactions can't be deleted but a subsequent transaction can undo the previous transaction. The ledger of transactions is immutable but the data in the database is mutable.

Advanced Cryptographic Tools for Privacy

The distributed consensus pool acting as a trusted third party may enable one more layer of privacy, that is, as the arbiter of zero knowledge proofs between entities. Zero knowledge proofs are a family of cryptographic techniques that allow an entity (prover) to prove something to another entity (verifier) without conveying any information other than the proof[48]. This prevents the verifier from being able to disclose information to a third party. Using zero knowledge protocols, the consensus pool or other trusted parties can aid the proof of attributes of an identity without discovering additional information about the identity. Zero knowledge protocols can be computationally demanding, however, simple forms of zero knowledge such as blinded signatures are not.

Another useful anonymity tool is blind signatures. By analogy a blind signature can be achieved with an envelope that is coated with ink on the inside. A letter can be placed in the envelope and a signer can observe the letter being placed in the envelope and sign the outside of the envelop without reading the letter inside. The letter picks up the signature from the ink coating so that once opened a reader can see the signature and know that the signer did indeed observe the letter being placed in the envelope and sign that it is valid but the signer does not know what is written on the letter.

A judicious use of these tools can achieve higher levels of privacy at reasonable computational cost. With the privacy approaches described above, the proposed identity system would be able to provide sovereign, secure, and private identities.

Conclusion

As the purpose of this white paper is to describe the essential characteristics of an identity system that provides sovereignty, security and privacy, the focus has been on those three characteristics. Obviously there are many other aspects of an identity system that have been left out[49]. These include but are not limited to, authentication, authorization, validation of identity attributes, trust assertion and transferal, credential management, identity proofing and reputation[50]. All of these other aspects of identity can be layered on top of the essential foundation described above. The crucial point is that the conventional identity system approaches have difficulty simultaneously providing high degrees of sovereignty, security, and privacy. It is the application of modern cryptographic techniques with distributed consensus on an open platform that enables all three characteristics to be provided in a uniquely beneficial combination.

Endnotes

1. Open Reputation Low Level White Paper.
   https://openreputation.net/open-reputation-low-level-whitepaper.pdf ↩

2. Open Reputation High Level White Paper.
   https://openreputation.net/open-reputation-high-level-whitepaper.pdf ↩

3. Open Reputation.
   https://openreputation.net ↩

4. XDI Identifier Registry, Cloud Name, Cloud Number.
   XDI Identifier Registry ↩

5. WebDHT Decentralized Identifier Registration Algorithm.
   WebDHT Decentralized Identifier Registry ↩

6. Self-certifying Identifiers.
   SFS Self-Certifying Pathname
   SFS-HTTP Self-Certifying URLs ↩

7. Identity Graph in Open Reputation.
   https://openreputation.net/open-reputation-low-level-whitepaper.pdf ↩

8. Mastering Bitcoin, Unlocking Digital Cryptocurrencies By Andreas M. Antonopoulos, 2014.
   Mastering Bitcoin, Antonopoulos 2014 ↩

9. Open Mustard Seed (OMS) Framework, idcubed.org. Individual control over data and identity.
   Open Mustard Seed (OMS) Framework, idcubed.org ↩

10. Open personal data store/safe answers framework (OPENPDS/SA).
    openpds/sa ↩

11. OpenPDS source code on GitHub.
    OpenPDS source code github ↩

12. Control your identity, BYU Domains.
    BYU Domains, Control your digital identity, domains.byu.edu ↩

13. Sovereign Source Identity by Philip J. Windley. “A sovereign-source identity, in contrast, is one created and maintained by a person for their own purposes.”
    http://www.windley.com/archives/2016/01/soverign-source_identity_autonomy_and_learning.shtml ↩

14. CPanel hosting platform.
    CPanel
    CPanel Documentation
    CPanel Code ↩

15. User Managed Access.
    UMA ↩

16. RAET Reliable Asynchronous Event Transport Protocol. End-to-end encrypted and signed.
    Raet secure protocol ↩

17. Ioflo Asynchronous Flow Based Programming Framework.
    Ioflo
    Ioflo Github ↩

18. NaCl, ECC Networking and Cryptography library.
NACL Library
CurveCP
LibSodium
libnacl ↩

19. Blockchain: Blueprint for a New Economy, Melanie Swan, O'Reilly, February 8, 2015.
Blockchain, Swan 2015 ↩

20. Proof of Work (POW) in Bitcoin.
https://bitcoin.org/bitcoin.pdf ↩

21. Proof of Stake (POS).
http://en.wikipedia.org/wiki/Proof-of-stake ↩

22. Byzantine Fault Tolerant Consensus Protocol or Byzantine Agreement.
Byzantine Fault Tolerance Protocols ↩

23. Bitcoin side chain, Factom White Paper.
http://factom.org ↩

24. Practical Byzantine Fault Tolerance.
PBFT 1999
PBFT Recovery 2002 ↩

25. Tendermint Byzantine Agreement Protocol.
Tendermint BA Protocol
Consensus in the Presence of Partial Synchrony, 1988 ↩

26. Aardvark, Robust Byzantine Fault Tolerant Protocol.
Aardvark Robust BFT 2009 ↩

27. RBFT: Redundant Byzantine Fault Tolerance by Aublin, Mokhtar and Quema, 2013.
RBFT: Redundant Byzantine Fault Tolerance 2013
BFT Protocols, 2014 ↩

28. Ripple Protocol Consensus Algorithm.
Ripple Ripple Protocol Consensus Algorithm
Fast Byzantine Consensus, 2006 ↩

29. Stellar Consensus Protocol.
Stellar Consensus Protocol, 2015 ↩

30. Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems by Tim Swanson, 2105.
Permissioned Distributed Ledgers, Swanson 2105 ↩

31. Multiply Anchored Block Chains.
https://openreputation.net/open-reputation-low-level-whitepaper.pdf ↩

32. Distributed Hash Table.
Distributed Hash Table ↩

33. StorJ Decentralized Cloud Storage.
StorJ
StorJ White Paper ↩

34. MaidSafe Crowdsourced Cloud Storage.
MaidSafe Crowdsourced Cloud Storage ↩

35. Interplanetary File System, IPFS.
IPFS ↩

36. NameCoin ID Registry.
NameCoin ID Registry ↩

37. DNSChain Domain Name Registry.
DNSChain Domain Name Registry ↩

38. BlockChainID Registry.
BlockChainID Registry
BlockStack BlockChainID
OpenName ID Registry
OneName ID Registry ↩

39. Triple-Signed Transactions.
Triple signed receipts
Triple entry bookkeeping
Ricardian contracts ↩

40. Privacy as familial similarity. “I’VE GOT NOTHING TO HIDE” AND OTHER MISUNDERSTANDINGS OF PRIVACY, by Daniel J. Solove. 2007.
“I’VE GOT NOTHING TO HIDE” AND OTHER MISUNDERSTANDINGS OF PRIVACY, Solove 2007 ↩

41. Privacy in the Modern Age: The Search for Solutions, by Marc Rotenberg (Editor), Jeramie Scott (Editor), Julia Horwitz (Editor) May 12, 2015.
Privacy Modern Age ↩

42. Intellectual Privacy: Rethinking Civil Liberties in the Digital Age, by Neil Richards (Author) 2015.
Intellectual Privacy ↩

43. Identity is the New Money by David Birch, 2014.
Identity is the New Money, Birch 2014 ↩

44. The Necessity of Standards for the Open Social Web by Harry Halpin, 2014.
The Necessity of Standards for the Open Social Web, Halpern 2014 ↩

45. From Bitcoin to Burning Man and Beyond, edited by John H. Clippenger and David Bollier, 2014.
From Bitcoin to Burning Man and Beyond 2014 ↩

46. Enhanced Privacy ID.
Enhanced Privacy ID ↩

47. Identity Ecosystem Framework.
IDEF ↩

48. Zero Knowledge Proof.
Zero Knowledge Proof
SCIPR Lab ↩

49. Digital Identity by Phillip J. Windley, O'Reilly Media, August 2005.
Digital Identity, Windley ↩

50. Building Web Reputation Systems, Randy Farmer and Aaron Glass 2010.
Building Web Reputation Systems, Farmer & Glass 2010 ↩