Web3: A Decentralized Societal Infrastructure for Identity, Trust, Money, and Data

J.W. Bambacht and J.A. Pouwelse
J.W.Bambacht@student.tudelft.nl, J.A.Pouwelse@tudelft.nl
Distributed Systems, Delft University of Technology
January 21, 2022

***Abstract*—A movement for a more transparent and decentralized Internet is gaining popularity globally. Users raise more awareness to the privacy of their online identities and data. The problem with Web2, the current version of the Internet, is its focus on companies that provide services in exchange for user data. The ownership of this data remains in the hands of the platforms. Web3 aims to solve this by making infrastructures decentralized and bring back power to the users. Decentralization is characterized by a zero-server architecture. This problem is not only limited to big-tech companies, but also for governments. Governments generally own and manage the identities of their citizens. The identity owner does not even have control over what personal information is exchanged to affiliated organizations. Financial privacy for individuals is in-existent as governments have insight in every transaction. Cash payments and blockchain transactions is the only form of money exchange that is quasi-private. Governments and banks have also started using cloud services to enable a more efficient and innovative manner of analysis. This exposes even more security and privacy risks for the users. The use of such services is against the nature of decentralization. Big-tech centralized communication platforms like WhatsApp fail to deliver privacy to their users. They also fail to provide trust in identity authenticity between participants of online conversations. Trust enforcement is a difficult topic as it requires more personal information and behaviour to be successful.**

The overall challenge is to move back the power to users and citizens, something that has been violated for far too long. The initial step is to introduce decentralization as it gives the user full control over its own data. This thesis is the first exploratory study into a decentralized social infrastructure for identity, trust, money, and data. A working infrastructure has been developed for Android that makes use of the P2P network overlay IPv8 [1], and TrustChain [2], a personalized blockchain. It makes a contribution to a reformed financial and tech sector that is more efficient and effective in serving the wider economy, and more resistant to the bad behavior of all kinds. Creating such an infrastructure that is decentralized and anti-fragile is seen as essential for the future.

I. Introduction

The current digital infrastructure and financial system are unfit and structurally unfair to citizens. Citizens have no alternative to big tech monopolies, banking services, and their anti-competitive practices. Governments have failed to protect *consumer welfare* while keeping control over their citizens' personal identities. The WhatsApp[1] messaging platform is a motivating example of market failure. WhatsApp violates terms of service over a long period [3]. A migration of WhatsApp users was initiated after an update of their terms of service [4]. The sudden migration was a consequence of WhatsApp aiming to give more user data to the mother company Facebook[2]. Signal[3], a competitor focused on privacy and openness has barriers to market entry, no network effect, and compete against a long existent closed protocol. These platforms have stickiness: easy to use, but it's often considered too complex to migrate. Citizens, and small(er) competitors [5], are powerless in this uncompetitive market.

The digitization of citizens' personal information by governments poses privacy concerns. Institutions and organizations require citizens' government-issued documents, both online and offline. Revocation mechanisms for identities are not in place. Your document has been copied and stored numerous times by banks, insurance companies, hotels, employers, and so on. The user has to rely upon that it is handled and stored with care. Authentication mechanisms for digital identities are widely deployed and exploited by authorized institutions. More than a required amount of personally identifiable information is requested and stored on the organization's central server. In these situations, the users are not in control of their data and are not able to exert any influence. A direct side-effect of a more digitally oriented society is the concern around people's privacy. The European Union started an ongoing effort into the General Data Protection Regulation (GDPR) [6] in 2016. Its main purpose is the misuse of privacy-sensitive data by companies. Big companies and platforms have failed to offer compliance to personal data protection. Over 900 filed cases of GDPR complaints, good for over 1.3 billion euros [7], most were dedicated to big-tech companies. The GDPR in combination with an intensive effort of the EU and marketing campaigns, people finally became more aware of their privacy of online data and identities [8]. Another implicit consequence of digitization is that an increasing amount of personal data may be exposed. Storage of personal data and weak security mechanisms of platforms both at the expanse of the user. Centralized applications offer good performance in terms of efficiency, consistency, and synchronization. The users have to trust that the owner of the platform has the best intentions with their personal data. This is difficult, especially when our data is used as their product, sold for personalized advertisement, for company revenue and value.

[1] https://www.whatsapp.com

[2] https://www.facebook.com

[3] https://signal.org

Governments, banks, and tax offices have almost complete insight into (digital) money flows of their citizens, often using big-tech cloud services, as a violation of the privacy of the user. Additionally, transaction costs are disproportional as debit card transactions range from about €0,05 to €0,20 per transaction [9], and for online payment services like iDEAL even more [10]. Blockchain-based wallets, in combination with Central Bank Digital Currencies (CBDC), can have a positive impact on both privacy and costs. Those transactions are not (directly) traceable by governments and banks, and no additional costs are charged for transactions and possession of (bank) accounts. The use of cash transactions in the form of banknotes and coins still offers decent privacy. The use of cash is (slowly) decreasing but still accounts for about one-fifth of all transactions in the Netherlands [11]. We must not forget that the exchange of cash is not the only purpose of cash as it also provides a store of value.

This research makes the following contributions: (1) design of a decentralized infrastructure in which the owner of a self-sovereign identity has control and power, (2) trust enforcement in authenticity between communicating participants in the network, (3) generic transfer of value (money and data) between identities. The final product is an infrastructure that is near market-ready and handles all the before-mentioned aspects.

II. Problem Description

The purpose of this study is to design a societal infrastructure that combines and transfers identity with trust while facilitating a private transfer of money and data in a permission-less fashion without the involvement of any centralized component. By removing these single points of failure, the violation of privacy and security of users is reduced. With centralization, even if the data is exchanged in encrypted form, an intermediary (the platform owner) is still able to see and collect your metadata. The user should in all situations be the owner of his own personal data and make the decision of what information is exchanged to others.

One of the key aspects is the use of citizens' self-sovereignty identities. The definition of SSI is characterized by the ten principles of Allen [12, 13], that target the insurance of the users' control within its own SSI, with a balance between transparency, fairness, and protection. The control of these identities is currently with governments. Moving the control to the user results in various advantages. Firstly, the user is the owner of their own identity and can view and decide what information to share. Secondly, as governments don't have control anymore, less personal data management is required, less bureaucracy, and a cost reduction for facilitating the heavily secured infrastructure and successful authentication. And thirdly, a minimum amount of personal data is stored on central servers or in the cloud, reducing the possibility of data breaches and theft.

The self-sovereignty of data, in any form, is also a fundamental issue of centralized platforms. Despite the use of servers is profitable in terms of availability and synchronization, it violates the privacy of users. The data itself is often encrypted, but the metadata that contains various attributes (sender, recipient, time, location, .etc) cannot. WhatsApp, the most widely used messaging app [14], promises its users end-to-end encryption. Despite their efforts, manipulation of messages [15] has been possible.

Communication channels lack trust in the authenticity of other participants' identities. No platform currently integrates government-issued identity information, let alone its use for enforcement of trust to other participants. The information that is generally applied to enforce trust (name, picture, phone number, or email) are components that require manual provision. Malicious actors try to apply them as genuinely as possible to mimic someone's identity. A desirable change would be to remove most of these editable components.

The exchange of money has some deficits with respect to privacy, mainly caused by governments, banks, and tax offices. Financial accounts are heavily supervised with no possibility of a private (digital) exchange. The transfer of money comes with disproportional costs, adverse cross-border payments in terms of speed and additional costs, and unwanted transparency. Many of these issues can be solved by the use of blockchain technology. With (almost) zero costs, transactions are executed between wallets anywhere in the world in a matter of seconds. Even internal use of blockchain for banks themselves will save about 10 billion dollars globally [16].

A decentralized societal infrastructure has much to offer in terms of privacy of the user, trust enforcement, self-sovereignty, cost reduction. In the following sections, the design and implementation of the first user- and identity-centric infrastructure is presented. Also, an experimental analysis of a self-developed P2P data transfer protocol within the designed infrastructure is executed and evaluated.

III. Related Work

The application of SSI's enables citizens to be in control of their own identity. Governments enable citizens to authenticate organizations and institutions to their identities, stored on their central server. DigiD[4], the primary authenticator in the Netherlands, enables citizens to authenticate only by use of their mobile phone. Personal data is transferred from the government's server to the organization's server. Not only does this require a perfectly secure connection and infrastructure on both sides, but the citizen also has no control over what information is actually shared. Also, the government spends an enormous amount of money as every successful authentication costs roughly $€0,13$. SSI's can be successfully applied to mobile applications that replace the necessity of authentication services like DigiD. A first example is IRMA[5], a mobile platform that authenticates itself once and stores the SSI and other personal information locally on the phone. This can then effectively be applied to authenticate organizations without using government servers. The user also controls what information is required and what is actually shared. Sovrin Network[6], a blockchain-based ecosystem that enables other developers to build their own SSI application on top with the

[4]https://www.digid.nl
[5]https://irma.app
[6]https://sovrin.org

TABLE I: Characteristics of competing platforms

	decentralized	P2P	open source	E2E Encryption	metadata	requirements	attributes used for trust enforcement	wallet	maturity [a]	note
WhatsApp [17]	✗	✗	✗	curve25519	✓	phone number	phone number, name, profile picture and status	✗	high	
FaceBook Messenger [18]	✗	✗	✗	curve25519	✓	FaceBook profile	FaceBook profile, name, profile picture	✗	high	[b]
WeChat (QQ) [19]	✗	✗	✗	✗	✓	phone number	phone number, name, profile picture and ID	money	high	
Telegram [20]	✗	✗	✓	MTProto	✓	phone number	phone number, name, username, profile picture and status	✗	high	[b]
iMessage [21]	✗	✗	✗	NIST P-256 curve	✓	Apple profile	phone number, name, email, profile picture	✗	high	
Signal Messenger [22]	✗	✗	✓	curve25519, curve448	minimum [c]	phone number	phone number, name, profile picture and status	crypto	high	
Session Messenger [23]	✓	✗	✓	curve25519, curve448	minimum [c]	✗	name, profile picture	✗	medium	[d]
Status.im [24]	✓	✓	✓	curve25519	minimum [c]	✗	username, profile picture	crypto	high	
Sylo [25]	✓	✓	✗	curve25519	✓	✗	name, profile picture	crypto	high	[e]
Berty [26]	✓	✓	✓	curve25519	minimum [c]	✗	name, profile picture	✗	medium	
Our design (Section IV)	✓	✓	✓	curve25519	minimum [c]	official Identity	identity name and verification status, profile picture	crypto	medium	

[a] The current state of development in terms of completeness and usefulness
[b] E2E encryption not enabled by default
[c] no storage of metadata, only required for routing
[d] fork of Signal Messenger, onion routing for metadata anonymity, undelivered messages stored one of the distributed service nodes
[e] everyone can set up node and will be rewarded in crypto token SYLO

same goals in mind. In some situations, it is required that the identity must be revoked, for example when the identity is lost or stolen. Both IRMA and Sovrin introduce authorities that handle the revocation, which is a violation of the principles of SSI as it should be an authority-free system. Chotkan [27] provides a distributed attestation revocation of SSI's. Offline verification is more privacy-aware and offers a more robust solution for digital attestations.

As mentioned before, this paper presents a *novel* decentralized infrastructure as it incorporates a government-issued identity within a messaging platform. Many other platforms exist, both centralized and decentralized, that apply at least some of the key points of this paper. To create some kind of overview about what is in the market already, Table I portrays a (non-exhaustive) list of significant and related competitors. The difference in characteristics between the centralized and decentralized platforms shows a clear clustering. The centralized platforms all have a privacy-sensitive asset as a requirement for its use and many different attributes are shared with contacts for identification and trust enforcement purposes. The decentralized platforms are examples of Privacy by Design [28] implementations as they try to minimize the leakage of privacy-sensitive information. There are no explicit requirements and the trust attributes are limited to manually chosen names and profile pictures.

WhatsApp [17], FaceBook Messenger [18], and specifically WeChat [19], are all fully centralized platforms that all store metadata of their users. All platforms but WeChat have integration for commonly-used E2E encryption curves, due to their performance in terms of speed and secrecy, and only Telegram [20] applies their self-designed protocol. WeChat, which is monitored by the Chinese government, incorporates strong censorship and interception protocols for data exchanged by its citizens. Luckily, this degree of violation is not present in any other (centralized) platform. Also, these centralized platforms are often obliged to, also because of their infrastructure design, provide information (stored metadata) to governmental instances or apply censorship in some situations, all upon request. Signal Messenger [22], that is centralized but specifically designed with privacy in mind, do not store any personal information. Central servers, however, are deemed necessary for routing and account recovery using the same phone number. Characteristically, most of the centralized platforms don't provide full transparency and rather do not share the structure of their platform openly.

Decentralized infrastructures try to enforce anonymity by reducing the metadata in the network as much as possible. Session Messenger [23], a *decentralized* fork of Signal Messenger, attempts to provide anonymity and preservation of privacy using a technique called onion routing. It makes it nearly impossible for any intermediary (node) to derive both the sender and receiver of the message. It is not possible to apply this technique in a (fully) P2P network as peers only know a limited number of other peers and do not (necessarily) communicate with nodes. Status [24], Sylo [25], and Berty [26] are decentralized, P2P, secure, minimize leakage of privacy-sensitive information, and provide the most preferable features, apart from the absence of SSI integration. Status is built on the Ethereum network and incorporates their own utility network token that fuels their network and provides (paid) options to users. In a similar fashion to Session Messenger, undelivered messages are stored on nodes that obtain your IP

address for delivery at a later moment. This is not a preferable characteristic as this contradicts the principles of privacy. Sylo is a complete platform that does not deliver full transparency and cannot withstand the leakage of information in the metadata. Berty has all potential as it is secure and transparent, minimizes leakage of privacy-sensitive information in terms of metadata and requirements, but is currently not yet fully developed.

Many different implementations exist that are somewhat designed on similar characteristics. The idiomatic platform is decentralized and P2P with no temporary storage messages on nodes applies trusted curves for E2E encryption, no use of metadata, and has no useless requirements. Trust enforcement attributes should be limited to not only manually forge-able components as it achieves higher trustworthiness. Integration of a wallet to provide a more privacy-aware exchange of value is desirable.

IV. Design

The design of the platform can roughly be divided into four main elements: identity, trust, money, and data. These elements are combined and integrated within a framework to form a functional platform. The following section describes the design of the elements in detail. The elements are required to satisfy the requirements and functionalities that are deemed necessary for a self-sovereign, secure, and privacy-aware communication ecosystem.

A. Infrastructure

As the platform has several functionalities, the infrastructure must combine these seamlessly. Firstly, the prominent problem of leading societal platforms is their centralized nature. It's obvious that decentralization targets many weak spots of centralization. A decentralized network has the purpose to provide storage of data in a distributed way. The addition of a P2P network within a decentralized network makes it possible to provide direct communication between peers without any intermediary. As no intermediary is able to act as a middle-man or adversary, it provides an extra layer of privacy and security. This communication can only be sufficiently secure when the message, or data, is encrypted. Apart from the networking layer, we require a way to store and exchange data in a distributed manner. In distributed systems, one of the requirements is synchronization across many independent nodes, which is difficult to realize without the need for continuous communication. Persistent storage and exchange of data, in particular transactions, that do not require continuous synchronization can be provided by blockchain technology.

Peers in the network are constantly looking for other peers in the network as no central server or node monitors the online activity of participants. To maintain a certain degree of anonymity, we don't want to spread personal information to other peers during the introduction. Thus, some sort of anonymous form of peer identification is required. To ensure a completely secure communication channel, the principles of the CIA Triad [29] should be applied. The objectives of a secure system include *Confidentiality*, *Integrity*, and *Availability*. Confidentiality is achieved by encryption as it ensures that data is only accessible to authorized parties. Digital signatures ensure the integrity of the data by providing proof that it is originated from the sender and has not been altered by any third party. The availability is slightly more difficult to ensure in decentralized systems, especially in P2P networks, due to its dependence on the connectivity of individual peers (or nodes). A commonly-used mechanism for secure communication is public-key cryptography [30]. Not only does it provide a confidential exchange of messages and data, but it is also a way to identify peers without exposing any private information. Each peer is equipped with a so-called public-private key pair. A private key is generated at once and should be known to its owner only. The private key performs the decryption of encrypted data. A public key is mathematically derived from the private key and may be publicly disclosed. It is computationally infeasible to derive the private key from the public key. A public key serves multiple purposes. Firstly, it provides a way to find, or address, other peers. Secondly, encryption of data is performed using the public key of that receiver. The peer that encrypted the data is not able to read the contents anymore. Thirdly, digital signatures provide proof of authenticity of a piece of data, which can be verified with the public key of the signatory.

As we want to reduce the exposed metadata to an absolute minimum it is important that data packets are not widely spread on the network, hoping that other peers will deliver it to the intended recipient. The metadata of a packet should, ideally, only contain information about delivery, that is, the receiver's public key or IP address. The risk of exposing privacy-sensitive information in a P2P network is minimized as peers directly communicate without any intermediary nodes or peers. As peers come and go, it may happen that peers change connectivity status or change their network address. In these situations the peers announce their new address to all previously connected peers. As this may sound contradicting in terms of privacy, no personal information, including communication histories, can be deduced as peers also connected to random peers to increase their network reach.

Apart from the networking layer, the infrastructure requires a persistent and decentralized store of data, in particular, transactions to enable the transfer of (digital) money. The blockchain is often applied to store transactions between two individuals in a permanent and uneditable manner. Every transaction on the blockchain is entangled to its previous block, making it a reliable 'chain' of tamper-proof assets. This very basic form of storage is a fast, lightweight, and structured alternative to conventional storage. Every transaction can be back-traced to create a well-organized overview, which is well suited to serve as a wallet.

Networking Layer: To be able to communicate with other peers we need a networking layer that handles communication. This can be realized using IPv8[1], a P2P networking layer that provides authenticated and privacy-aware communication between peers. IPv8 is developed as a possible successor of IPv4, in an attempt to overcome IPv4's weak characteristics

and increasing problems. The objective of IPv8 is to provide a zero-server infrastructure with equal status and power within the network for everyone and to provide perfect secrecy with E2E encryption. IPv8 is capable to establish connections to other peers, even for devices that are connected using NAT or behind a strong firewall. A customized NAT traversal technique, UDP hole-punching, is effectively applied to provide increased privacy. The endpoints of the networking layer are independent of any central infrastructure.

IPv8 applies the concept of so-called network overlays (communities). This enables developers to build applications on top of the base networking layer by creating their own community. The base community includes all functionality in regards to peer connectivity, communication, data serialization, and encryption. In every community, the list of peers may differ because peers must join a community to participate, or the peer is not in the list of connected peers (yet). A specific discovery community takes care of discovering and connecting new peers that are present in the same community, on the basis of distinctive discovery strategies. The communication with others in the network and community are handled through endpoints (sockets). IPv8 supports both online and offline communication, by either using a UDP endpoint Bluetooth endpoint, or both.

Distributed Ledger: The ecosystem requires a distributed ledger that provides the transfer of data and (digital) money and its storage. TrustChain [2], a permission-less scalable distributed ledger, is already integrated as a community on top of IPv8 and is therefore a proven candidate. TrustChain has the capability of sending and receiving trusted transactions between peers. The blockchain-based data structure is a tamper-proof immutable chain of transactions. There is no central control over the transactions. Trust between participants is built as they communicate in a Sybil-resistant way. Every peer implements a personalized chain that only contains blocks that are either sent or received by a peer. TrustChain has three basic functionalities: sending and receipt of blocks, broadcasting of blocks and crawling of chains. The send and receipt process includes both parties in one transaction. The initiator (S) signs and sends a *proposal block* (a half block) to the counterparty (R). On receipt of the *proposal block*, (R) creates, signs, and sends the *agreement block* back. During the process, the integrity of the received block is validated and both parties add the half blocks to their chains. The half blocks are linked by the public key of the counterparty. The transaction is considered complete when both parties received and signed both half blocks. Not only is it possible to send a block to a specific peer, but also to broadcast a block to all currently connected peers. The crawl functionality is nothing more than retrieval of a peer's chain using its public key. It regularly happens that any request to another peer does not result in a response. In these cases, the request is automatically repeated until a response is received.

B. Identity

Identity is an integral part of citizens when it comes to ownership over their self-sovereign identity. Integration of their legally valid government document introduces various new purposes. One of these purposes is the authentication of online institutions. The owner controls the exchange of its own information to organizations compared to the conventional governmental authentication that blindly sends every piece of information. Authentication can only serve its purpose if the information within the self-sovereign identity is authentic. IRMA, application mentioned in Section III, achieves authenticity by fetching the attributes from the government's central servers once. This is not a suitable option as we desperately want to eliminate the use of central servers. Every citizen is obliged to possess a physical government-issued travel document in the form of a passport or identity card. These documents contain (visually) the exact same identity information in the machine-readable zone (MRZ) as the government's servers. The documents contain a built-in biometric chip that is able to communicate with the NFC chip of mobile phones. Due to security concerns, the chip can only communicate after knowledge of the MRZ of the document is proven. This required information is placed in the MRZ of the document. The phone camera can effectively be applied to obtain these attributes while ensuring authenticity. The use of AI ensures a correct scan of the attributes on the document. The mandatory attributes are transferred to the chip of the document to request all embedded attributes digitally in an authentic manner. The chip only returns the attributes when the provided attributes are valid. This process is deemed secure and authentic because (I) forged attributes in the MRZ zone are useless and cannot influence the process, and (II) the biometric chip in the documents is considered secure for this purpose as there mainly exists eavesdrop-attacks [31]. However, it must be noted that there does not exist a way to revoke access to a stolen or lost passport until the expiration date of the document. We must also consider the situation that a device has no support for the NFC chip, defectively or physically. No (offline) method exists to obtain the identity while still providing authenticity. This means that all identity-related functionalities cannot be trusted to contain truthful and authentic information. There is no other choice to disable these functionalities for these particular devices and users. All attributes of the self-sovereign identity must be stored in encrypted form on the phone to prevent identity theft when hacked, lost, or stolen. Not only do we need secure storage, but it is also desirable to enable biometric protection for access to the application, and when executing possibly irreversible actions like the transfer of money.

Another purpose of self-sovereign identities is the opportunity to use verifiable claims. In some situations, it is required to show your physical identity document to verify some details about your identity. The authority is not only capable of unnecessarily viewing the requested attribute, but also other attributes. This not only violates your privacy but can also damage a person's authenticity by misuse. Verifiable claims are claims about pieces of information, or data, that are verified using attestations. Chotkan [27] designed a system that incorporates verifiable claims that don't reveal the actual requested piece of information by the use of zero-knowledge proofs. To apply verifiable claims in a trustworthy manner the

information from the self-sovereign identity must, again, be authentic.

C. Trust

Centralized platforms have to deal with multiple types of trust. The first logical form is trust in a system or platform. As a user, you want to have faith that your personal data is handled and stored with care. This is often one of the primary problems with centralization. As all user data is stored on the platform's servers, you must have confidence that the data is protected with the highest security standards, exchanged in encrypted form, and not sold to any third parties. If no good alternative exists, also because no known person uses it, the user has to decide whether to use the platform and neglect the privacy-related issues, or not use the platform anymore. Often the first choice is chosen as people value the use of the service more than their privacy. Decentralization completely eliminates this trust, or distrust, as there is no central component or authority that decides over you and your data.

Within messaging and societal applications another form of trust arises. Users have to make a well-educated guess whether they are communicating with the person they are expecting them to be. This guess is mostly based on the provided information, the (dis)similarity in the way they communicate, and the discussed topics. The difference in punctuation, the use of capital letters, and the style of writing can in some cases also be recognized. Unfortunately in most applications, personal information can easily be forged or stolen from people's real online identities. If we look in Table I again, most attributes for trust enforcement of centralized platforms are easily forgeable. Hacked accounts often try to mimic truthful information in combination with simple conversations **REF???**.

The challenge is to exchange just enough trust to the recipient of your message, without exposing an unnecessary amount of private information. At the beginning of the conversation, especially if the users are connected through some online manner, trust (or mistrust) plays a major role. As valid SSI's are incorporated in our design, we can access authentic information. In a normal, physical first meeting, you would introduce yourself by your (first) name, and indirectly with your face, sound of your voice, and the overall atmosphere. Unfortunately, most of these are useless in a digital world. We can, however, exchange the name and photo, as embedded in the SSI, accompanied by the verification status. The verification status denotes the authenticity of the information, indirectly concluding the use of the NFC chip. Various combinations of the first name and surname exist that provide trust, see Table II. Option I may be too general, option II is already more specific but is still too vague. Option III and IV are already more personal and substantial, while option V is revealing the complete name that may be too privacy-violating. **TODO ARGUMENT**.

These attributes of the identity information are sent along with every message, in an encrypted manner. Upon receipt of the information, the system compares it with the currently stored state and looks for differences. Initially, together with the first message, the state is empty. The receiving user will be notified that the identity information has been determined, see Figure 1a. The information is notified in a recognizable manner, containing the sender's identity name, photo, and verification status. If during the conversation at some point the information changes state, noting the identity has been changed in the application, the user will receive a similar notification, stating that the information has been updated, see Figure 1b. This mechanism makes sure that the user always knows who he is communicating with, based on the imported identity of the other party. In case a phone is hacked or stolen, and the thief gained unauthorized access to the application, it is impossible to notice. As long as the biometric protection is in place, in the form of a passcode, fingerprint, or face recognition, it should be impossible to impersonate.

TABLE II: Trust enforcement options using identity name

	Combination	Example
I	{First Name}	Timothy John
II	{Last Name}	Berners-Lee
III	{First Name} {Surname[0]}	Timothy John B.L.
IV	{First Name[0]} {Surname}	T.J. Berners-Lee
V	{First Name} {Surname}	Timothy John Berners-Lee

To preserve the privacy of the receiving side, the identity information will never be sent without a message or transaction. This reduces the risk of malicious actors purposely fetching the name attached to public keys. Currently, the design does not feature the possibility to migrate from one phone to the other. As future designs may include this feature, this method is deemed safe against hackers taking over the account, like phishing attempts in WhatsApp **REF**.

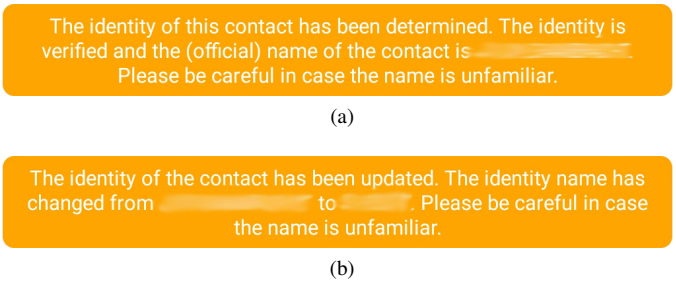


Fig. 1: Transfer of trust attributes

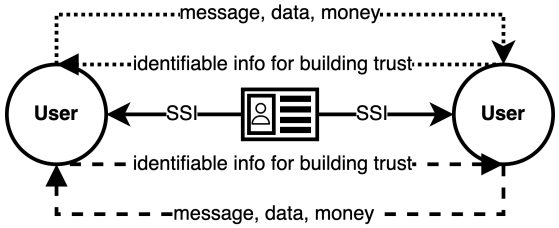


Fig. 2: Process of trying to enforce trust, starting from a government-issued ID to an imported SSI, to other the other users' SSI. The user then makes the decision to trust the contact and send a message, data, or money.

D. Money

As the need for financial privacy grows, many Web3 applications integrate the transfer of some sort of value. Many cryptocurrencies are used to transfer value from one person to the other. Governments of different countries discuss the introduction of a so-called Central Bank Digital Currency (CBDC). These currencies are digital reflections of their native currency, often called stable coins, that provide a fast and private transfer between participants. As governments, banks, and tax offices do not have insights into these transactions, the privacy of the users is preserved at a certain level. Blockchain solutions, in principle, are transparent, meaning that transactions and data on the blockchain is visible to a certain degree. As long as people know your public key, they often can view other people's activity on the blockchain.

TODO

E. Data

One of the key aspects of secure and private communication is the transfer of data. The current implementation of IPv8 also contains a data transfer protocol. As the protocol is very slow, basic, unreliable, it was deemed necessary to design a custom protocol.

The designed data transfer protocol (EVA) tries to exchange data, in binary form, in a progressive manner. This means that even in case some of the blocks are not received, it always wants to continue moving forward. The protocol applies similar tactics as in the Trivial File Transfer Protocol (TFTP).

TODO

V. Implementation

TODO

VI. Experimental Analysis and Evaluation

TODO UPDATE AND INCLUDE NEW RESULTS

In this section, an experimental analysis is performed to derive the optimal parameters for the designed EVA-protocol of section IV-E. Additionally, an evaluation using these optimal settings for larger-sized transfers to prove its applicability.

A. Experimental Analysis

To exploit the best possible performance, the designed binary data transfer protocol requires its settings to be optimal. We define the EVA protocol to be optimal if (1) the runtime, the time between the start of the transfer (by the sender) and receipt of the transfer (by the receiver), is as high as possible, (2) the ratio of unacknowledged blocks (as explained in Section IV) is as low as possible, and (3) the number of retransmits of windows (by the sender) and acknowledgments (by the receiver) is as low as possible. The latter two constraints contribute to a lower runtime because no extra blocks have to be transmitted and there's no additional idle time waiting for an acknowledgment.

The protocol's performance primarily depends on the variable parameters block size and window size. The block size is limited to 1500 bytes due to the MTU (Maximum Transmission Unit) of the ethernet [32], and thus generally the upper bound on the maximum block size. The IPv8 protocol adds a header of approximately 177 bytes to each block for routing, identification, and security purposes. As two blocks with a payload of 500 bytes carry twice as much 'useless' information as one block of 1000 bytes, a transfer using a greater block size is preferred and should theoretically finish earlier. However, other influences during transfer may result in unexpected behavior. An in-depth analysis will provide the optimal value for the block size, in which the payload size is varied from 600 bytes to 1000 bytes. As the exact size of the header is variable and unknown, and the serialized payload itself also contains additional informational bytes, a safe margin of approximately 200 bytes is chosen.

TABLE III: Parameters that are being tested for optimal execution. The file size and number of executions have only been used for consistency. In total 80 combinations of parameters have been executed 5 times.

Parameter	Values	
Block size (B)	$600, 700, 800, 900, 1000, 1100, 1200$	[bytes]
Window size (W)	$16, 32, 48, 64, 80, 96, 112, 128$	[blocks]
File size (F)	$5, 10$	[MB]
Executions	5	[-]

The window size is defined as the number of bytes (or blocks) the sender can transmit without having to wait for an acknowledgment of receipt, sent by the receiver. Theoretically, a higher window size would directly contribute to higher transfer speeds. A smaller number of acknowledgments has to be sent and received, decreasing the overall idle time. Higher window size also increases the possibility of late or lost blocks, specifically in an imperfect or congested network, therefore increasing the number of window and block retransmits. It is thus important to find the trade-off between high window size and loss due to undelivered blocks. Well-known sliding protocols apply a relatively small window size. The stop-and-wait, or alternating bit, protocol, only applies a window size of one, while TCP often uses a window size of 16. As these protocols sent much useless information in the form of headers as well, we'll try to find a suitable window size that is (much) higher. During the analysis, a window size of 16 to 128 blocks is tested.

Apart from the block and window size, other parameters do not have a direct influence on the performance. The retransmit interval may affect the performance when it is either too tight or loose, but it will only play a role in a small part of the cases. A tight interval can force windows of blocks or acknowledgments to be retransmitted while it is still in transit. For a loosely set interval, the protocol may unnecessarily have to wait for a window or acknowledgment. The overall transfer timeout interval, another parameter, is less critical and will only affect the performance when a window or acknowledgment has been retransmitted and employed all retransmit attempts. The retransmit attempt count likewise has little influence on the performance.

Experimental Setup: The experimental setup consists of two phones, a Xiaomi Redmi 9T using Android 10 and a

Huawei P20 Lite using Android 9, both 4GB RAM. The phones have the same version of the app installed and are connected to the same WiFi-6 mesh network (NETGEAR Orbi RBK753). To obtain a better estimation, the experiments are repeated a total of five times. Also, to verify the independence of the file size of the transfer, the experiment is executed for a file size of 5MB and 10MB. Each important step of the EVA protocol is captured in a log to be processed in Python. An automatic Kotlin script makes sure that every combination of parameters, the file sizes, and the five iterations are executed consecutively. Table III gives an overview of the tested parameter values.

Experimental Results: The optimality of the protocol can be determined in combination with the before mentioned requirements.

The first requirement, the runtime of the protocol, is displayed in Figure 3a. We can clearly see the effect of both the block and window size. The runtime remains improving for increasing block size but is becoming less significant. That last observation indicates that when we only look at the block size, the optimal block size would probably be not much greater than $B = 1000$, if the MTU would have allowed it. The runtime is optimal (lowest) for a block size of $B = 1000$. The window size follows a parabolic curve with the optimum somewhere in the middle of our chosen parameter value range. The optimal window size is slightly less pronounced than the block size, as window sizes of $W = 80$ and $W = 96$ show very similar results, with absolute values of approximately 23 seconds for a transfer of 5MB with a block size of $B = 1000$. Even when the runtime offers a complete picture of the performance of the transfers, it is difficult to decide on the optimal window size just yet. It is important to include the results of the other two requirements of optimality before deciding on the optimal window size, as optimality also includes the least retransmits of blocks, windows, and acknowledgments. Since the runtime is known, an estimate of the transfer speed can be given. The windows $W = 80$ and $W = 96$ both produce an average transfer speed of approximately 215kB/s.

The second requirement, the ratio of unacknowledged blocks during a transfer, is described in the plots of Figure 3b. The graphs follow the same trend as the previous requirement. Although the ratio of unacknowledged blocks is relatively low, every unacknowledged block will cause the runtime to increase as the unacknowledged blocks are added to the next window of blocks. Also, the more unacknowledged blocks are added to the next window, it becomes more likely that some blocks in that window will not arrive in time. Compared to the first requirement, the effect of the varied block size is less pronounced than the window size for the second requirement. For every window, the ratio of unacknowledged blocks for all blocks sizes behave similarly, with small deviations. Also, if we look at the difference between the file size of 5MB and 10MB there is no notable difference. The optimal block size cannot be determined from these results. The window size on the other hand again has a parabolic curve, highlighting the sizes of $W = 64$, $W = 80$, and $W = 96$. The combined ratio of unacknowledged blocks for these three windows (that include all block sizes) are 1.57%, 1.46%, and 1.03%. Based on the number of unacknowledged blocks, a window size of $W = 96$ should be preferred best.

The third and last requirement, the number of retransmits of windows and the number of retransmits of acknowledgments, combines mistakes on both the sender's and receiver's side. During a retransmit, the sender (or receiver) already had to wait (at least) one retransmit interval for the blocks or acknowledgment. This is something that would have serious effects on the run time if it occurs more frequently. It is thus crucial to have it reduced as much as possible. In Figure 3c and 3d the plots of retransmitted windows and acknowledgments are displayed, respectively. For both types of retransmits, the number of retransmits for window sizes of $W = 80$ and lower is neglectable. The number of retransmits for the three largest window sizes is increasing. We could argue that any window size smaller than $W = 112$, or even $W = 96$ if we would be really strict, is a good choice based on the results. The block size, again, does not appear to have a notable influence on the number of retransmits. An attempt was made to check whether the choice of a too tightly set retransmit interval caused the number of retransmits. For the largest three window sizes, this interval was increased from 3 to 5 seconds. As the results showed that the number of retransmits slightly, but not significantly, decreased and the runtime was increased unwillingly, further investigation was deemed unnecessary.

We must take into account that the tests have been performed under somewhat optimal circumstances: phones only running system services and the test application, and both connected to the same local WiFi network. From this, we could argue that if the conditions worsen, the number of retransmits of windows and acknowledgment would logically increase. The tests with a window size of $W = 96$ already exposed more retransmits than for a window size of $W = 80$. Both $W = 80$ and $W = 96$ make good candidates for optimal execution of the protocol. But if a choice had to be made, we would choose the former, primarily due to the slightly better performance in terms of retransmits.

TODO: verify the independence of file size.

B. Performance Evaluation

TODO: 250MB test (or a size that is allowed by the JAVA heap size on the phones...).e

VII. Conclusion

TODO

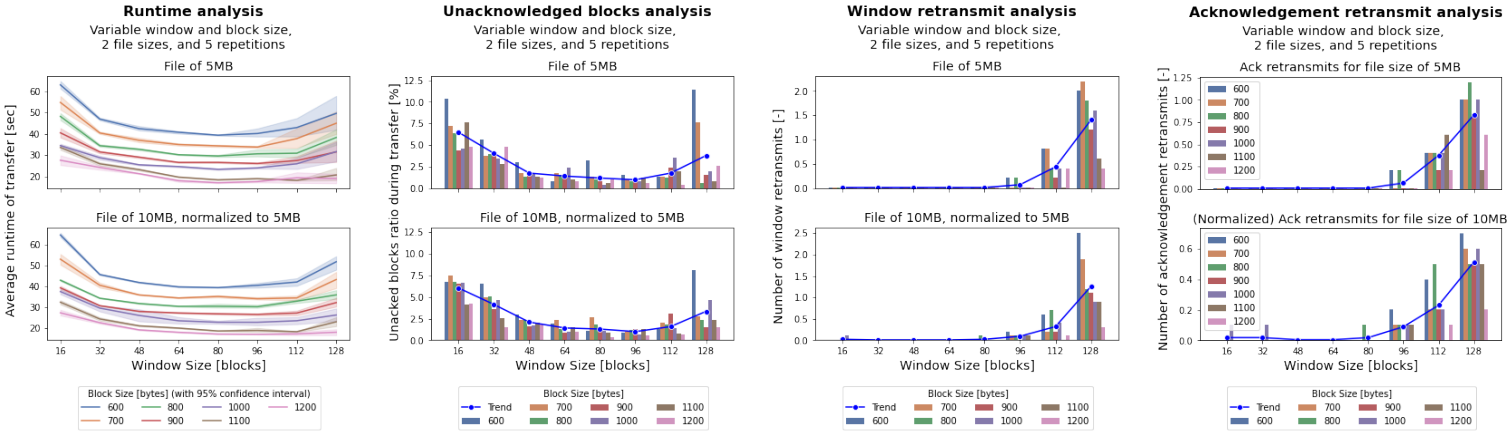
VIII. Future Work

TODO

References

[1] Tribler. Ipv8 documentation. 2021. URL https://py-ipv8.readthedocs.io/_/downloads/en/latest/pdf/.

[2] Pim Otte, Martijn de Vos, and Johan Pouwelse. Trustchain: A sybil-resistant scalable blockchain.



(a) (b) (c) (d)

Fig. 3: The results of all executed tests for each window size and block size. Each experiment is executed for a file size of 5MB and 10MB and is repeated a total of five times.

Future Generation Computer Systems, 107: 770–780, 2020. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2017.08.048. URL https://www.sciencedirect.com/science/article/pii/S0167739X17318988.

[3] Nicolo Zingales. Between a rock and two hard places: Whatsapp at the crossroad of competition, data protection and consumer law. *Computer Law & Security Review*, 33(4):553–558, 2017.

[4] The Guardian. Whatsapp loses millions of users after terms update. [Online] Available: https://www.theguardian.com/technology/2021/jan/24/whatsapp-loses-millions-of-users-after-terms-update, 2021.

[5] "Wall Street Journal". Big tech braces for a wave of regulation. [Online] Available: https://www.wsj.com/articles/big-tech-braces-for-wave-of-regulation-11642131732, 2022.

[6] Official Journal of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). [Online] Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN, 2016.

[7] Gdpr enforcement tracker. [Online] Available: https://www.enforcementtracker.com.

[8] Aikaterini Soumelidou and Aggeliki Tsohou. Towards the creation of a profile of the information privacy aware user through a systematic literature review of information privacy awareness. *Telematics and Informatics*, 61: 101592, 2021.

[9] PinDirect. Wat kost een pintransactie? [Online] Available: https://pindirect.nl/kennisbank/uw-eigen-pinautomaat/wat-kost-een-pintransactie/, 2020.

[10] ZakelijkBankieren.nl. Vergelijk ideal kosten per aanbieder in nl. [Online] Available: https://www.zakelijkbankieren.nl/kosten-ideal/, 2021.

[11] Dutch Payments Association. Facts and figures on the dutch payment system in 2020. [Online] Available: https://factsheet.betaalvereniging.nl/en/, 2020.

[12] Christopher Allen. The path to self-sovereign identity. [Online] Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html, 2016.

[13] Quinten Stokkink and Johan Pouwelse. Deployment of a blockchain-based self-sovereign identity. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1336–1342, 2018.

[14] Statista. Most popular global mobile messenger apps as of october 2021, based on number of monthly active users. [Online] Available: https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, 2021.

[15] Roman Zaikin and Oded Vanunu. Reverse engineering whatsapp encryption for chat manipulation and more. [Online] Available: , August 3-8, 2019.

[16] "Computer Weekly". "blockchain technology will help banks will cut cross-border payment costs by $10bn in 2030". [Online] Available: https://www.computerweekly.com/news/252509262/Blockchain-technology-will-help-banks-will-cut-cross-border-payment-costs-by-10bn-in-2030, 2021.

[17] WhatsApp. Whatsapp encryption overview. technical white paper. Nov 2021. URL http://www.cdn.whatsapp.net/security/WhatsApp-Security-Whitepaper.pdf.

[18] FaceBook Inc. Messenger secret conversations. technical whitepaper. May 2017. URL https://about.fb.com/wp-content/uploads/2016/07/messenger-secret-conversations-technical-whitepaper.pdf.

[19] WeChat. Wechat - free messaging and calling app.

"[Online] Available: https://weixin.qq.com", Jan 2022.

[20] Telegram. Telegram messenger. "[Online] Available: https://telegram.org", Jan 2022.

[21] Apple. imessage security overview. "[Online] Available: https://support.apple.com/en-au/guide/security/secd9764312f/web", May 2021.

[22] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila. A formal security analysis of the signal messaging protocol. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 451–466, 2017. doi: 10.1109/EuroSP.2017.27.

[23] Kee Jefferys, Maxim Shishmarev, and Simon Harman. Session: A model for end-to-end encrypted conversations with minimal metadata leakage. *CoRR*, abs/2002.04609, 2020. URL https://arxiv.org/abs/2002.04609.

[24] Status. The status network. a strategy towards mass adoption of ethereum. June 2017. URL https://status.im/whitepaper.pdf.

[25] Felix Schlitter, John Carlo San Pedro, Paul Freeman, and Callum Lowcay. Sylo protocol: Secure group messaging. 2020. URL https://www.sylo.io/whitepaper/sylo-protocol.pdf.

[26] Berty. Berty protocol. "[Online] Available: https://berty.tech/docs/protocol/", Oct 2020.

[27] R. Chotkan. Industry-grade self-sovereign identity, on the realisation of a fully distributed self-sovereign identity architecture. Master's thesis, Delft University of Technology, 2021.

[28] Ann Cavoukian. Privacy by design: The 7 foundational principles. May 2010.

[29] Wikipedia. Information security. "[Online] Available: https://en.wikipedia.org/wiki/Information_security", 2022.

[30] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. Nov 1976. URL https://ee.stanford.edu/~hellman/publications/24.pdf.

[31] A. Juels, D. Molnar, and D. Wagner. Security and privacy issues in e-passports. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 74–88, 2005. doi: 10.1109/SECURECOMM.2005.59.

[32] Andrew S. Tanenbaum and David Wetherall. *Computer networks, 5th Edition*. 2011.