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 4, 2022

Abstract—**Current messaging platforms not only violate privacy awareness of its owners but also have no proper way to enforce trust between participants. Migration to other platforms is too complex, forcing the user to continue to use these platforms. The legitimacy of the person that added you or chatting with is sometimes hard to determine. These platforms abuse the users' private data by making them the product to earn money and gain company value. The platforms are the owner and in control of its users' data and can even decide whether to 'delete' it at any point in time. While the centralised structure is partly the cause of that, decentralisation gives each user full control over its own data. Privacy is mostly fulfilled by the data only traversing the network without making an intermediate stop at the platforms central server. The lack of trust between users can be enforced by the integration of legitimate self-sovereign identities (SSI). These digital identities are composed from legally valid government documents and can therefore be considered trustworthy. Trust is an integral part when it comes to online communication, especially with involvement of money transfer.**

This thesis is the first exploratory study into a scalable societal infrastructure for identity, trust, money, and data. The implementation 'ConfIDapp' is built on a personalized blockchain called TrustChain [1]. It makes a contribution to a reformed financial/tech sector that is more efficient, more effective in serving the wider economy, and more resistant to bad behaviour of all kinds. Creating a societal infrastructure which is decentralised and anti-fragile is seen as essential, also due to our learnings from the Covid crisis.

I. Introduction

The current digital economy and financial system is unfit and structurally unfair to citizens. Citizens and economic actors have no alternative to banking services, big tech monopolies 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 fails terms-of-service over a long period [2]. In the beginning of 2021, tens of millions of WhatsApp users migrated to other services due to an update of their terms-of-service [3]. The sudden migration was a consequence of WhatsApp aiming to give more user data to mother company Facebook[2]. Signal[3], a competitor focused on privacy and openness has barriers to market entry (although the WhatsApp situation clearly helped), no network effect, and compete against a long existent closed protocol. It is certain that platforms like WhatsApp has stickyness: you simply install and use it, but it's often considered too complex to migrate. Unfortunately citizens are powerless in this uncompetitive market. Governments need to actively support *adversarial interoperability*.

People and businesses are increasingly becoming digitally oriented. Since 2016, the European Union has put an ongoing effort into the General Data Protection Regulation (GDPR) [4]. The GDPR targets the misuse of privacy-sensitive data by companies. Since then, big companies and platforms has failed to offer compliance to personal data protection. Over 900 cases of GDPR complaints were filed until the moment of writing, accounting for about 1.3 billion Euro's in total [5]. It is no surprise that the largest fines belong to big tech companies like Amazon, WhatsApp, and Google. With the help of the introduction of the GDPR, intensive effort of the EU, and marketing campaigns, people finally became more aware and more in control of their own online identities [6]. Becoming increasingly digitally active naturally has the deficit of exposing an increasing amount of personal data online. Many companies has been targeted by hackers stealing their users' personal data. Unfortunately, these companies often lack proper security mechanisms. Too much personal data is stored on their central servers. While centralized applications offer good performance in terms of efficiency, consistency, and synchronization, it is a gold mine for hackers when it comes to privacy and confidentiality. In a centralized system the users has to rely on trust that the owner of the system has the best intentions with their personal data. This is sometimes difficult when the service did not fulfil this in the past.

Some well known applications like Facebook tend to use the user as their product. Data derived from the users' interactions, preferences, and locations is more effectively applied for personalized advertisement, generating more company revenue. Every minor detail is tracked and stored on their central server. While these companies contributed by connecting people online in the last decade(s), their actions and visions nowadays has rightfully been criticised by many.

Apart from profitable businesses, governments also started digitizing citizens information. Organizations and other institutions require user information to effectively be able to execute their business. Many organizations require the user to submit it's government-issued document, both online and offline. Think of banks, insurance companies, hotels, and even employers. The amount of times your document is copied

[1] https://www.whatsapp.com
[2] https://www.facebook.com
[3] https://signal.org

and stored somewhere is tremendous. The user has to rely on the fact that it is handled and stored with care. Authentication mechanisms for digital identities, e.g. DigiD in the Netherlands, are widely deployed and exploited by authorized institutions. However, during authentication, privacy standards are often not respected. After authentication, a lot of personal identifiable information is sent to the organization and stored on their central server. Unfortunately, data leaks regularly occur in organizations, even government supported agencies. In the end, users should be in control of their own personal information, not the government or organizations.

It's no secret that offline money transfers in the form of banknotes and coins will eventually disappear. Currently, cash is still the second most preferred payment method, with in the Netherlands in 2020 worth for one-fifth of all transactions and two-fifth of all person-to-person transactions [7]. The Netherlands is one of the countries in Europe that is further digitally developed than average. Cash payments tend to be more important in less developed countries. The transfer of money, both online and offline, currently has the deficit that it requires additional costs. The costs of the use of an ATM or in-store debit-card transactions range from about €0,05 to €0,20 [8] per transaction, uncorrelated to the transaction value. Online payment services like iDEAL, the leading online payment method in the Netherlands, has an even more increased cost, depending on the webshops' contract with iDEAL. In the current economy the charged transactions costs are of unnecessary proportions. These costs can be neglected in certain blockchain-based applications. In future solutions the option for online/offline cash-like money transfer should still be available. Retaining peoples' privacy and lurking from government agencies over peoples' transactions should end. The application of Central Bank Digital Currencies (CBDC) enables cash transfers between people without the intervention of banks and authorities.

This research make the following contributions: (1) infrastructure in which a legitimate self-sovereign identity is central, (2) generating trust between participants in the network, (3) decentralized infrastructure for generic transfer of value (money and data) between identities.

II. Problem Description

The problem includes that the user, or citizen, is not in control over their own identity. The user requires the need for a self-sovereign identity. In short, the definition of a self-sovereign identity is that its owner is in full control over its own identity. A definition is characterised by the ten principles/properties of Allen [9]. The principles together target the insurance of the users' control within its own SSI, with a balance between transparency, fairness, and protection of the individual. A more extended view on the principles and their application with blockchain-based SSI's is given in Stokkink and Pouwelse [10]. The question really is how to effectively compose a legitimate digital identity and how to propagate it to gain trust without unnecessarily neglecting the privacy of its owner. The crucial part is to find the sweet spot between the amount of exposed privacy-sensitive information and the amount of trust that can be deduced from that information. In general, the more information that is exchanged, the higher the trust will be with the downside of unnecessarily violating the users' privacy.

The transfer of money, data, or any other form of privacy-sensitive information desperately requires secrecy and privacy. WhatsApp, the most widely used messaging app [11], promises its users end-to-end encryption. Reverse engineering introduces the possibility to manipulate and forge messages in chats [12]. Phishing using WhatsApp is frequently experienced as well. With Facebook, the least trusted big tech company when it comes to user trust in privacy [13], trusting (new) friends is even harder as fake profiles often tend to look legitimate. The underlying problem with existing applications is the lack of identity validation. These identities are manually created and propagated with a decent chance of not being legitimate. Phone messaging platforms like WhatsApp build trust based on a phone number and an optional nickname and profile photo. With the introduction of digital identities gathered from legally valid government-issued documents, trust can more easily be propagated to other identities. To keep the exchanged information secret from any eavesdropper, it must be encrypted. The encryption must be sufficient such that only the receiver can decrypt and read the contents. End-to-end encryption does not offer full disclosure at all times. In a centralized structure, the central server still requires the address, or identity, of the recipient of the message. The central server may process (and store) the messages' metadata, possibly containing IP addresses, sender, and other privacy-sensitive information. Stored data on a central server also opens the possibility to be vulnerable against hackers. Governments could in some cases enforce the service to hand over this data. In a decentralized system the users does not need to trust the system since there is no central authority that makes the decisions.

Online transfer of money is naturally, and always has been, a more thoughtful process than sending messages or data. In an online infrastructure it would contribute significantly when the receiver of your money is trusted by you. However, in contrast to a person-to-person bank transfer, the buyer rarely checks the legitimacy of the receiver, (the webshop) before making the payment. The introduction of CBDC's does not contribute to this problem. Cash payments will eventually disappear in the future. CBDC's provides users the function of online and offline money transfer. The government and tax authorities don't have the capability to look into (personalized) blockchains, which makes the use of CBDC's much more interesting as a replacement for cash-like payments. The only part that is visible to these authorities are the deposit and withdrawal of money from users' official bankaccount. Every transaction on the blockchain, which can be seen as transactions with cash in your physical purse or wallet, is private to a certain proportion.

III. Related Work

This is the first work that presents a societal decentralized infrastructure that combines identity with the enforcement of trust and the transfer of money and data. Nowadays, there exists many applications that enables the transfer of messages and data. Most of these applications are centralized in essence. As mentioned before, centralized structures are part of the problem of insecure and privacy violating applications.

Currently no application incorporates a self-sovereign identity within a societal application. All chat applications manually create identities based on personal preferences like phone number, mail-address, nickname, and profile photo. There are however applications available that allow users to authenticate its government-issued identity to verified authorities. DigiD[4], the predominant form of identity authenticator in the Netherlands, enables users to authenticate theirselves with only their mobile phone. The application provides the authentication mechanism and exchange of personal data between the government servers and verified authorities. Every time the identity is fetched from the government servers, and users have no option to decide what information to share. In short, they are not in control of their identity. Furthermore, authentication through DigiD is an unnecessary costly process since it roughly costs €0.13 per successful authentication [14]. IRMA[5], a platform that fetches and creates SSI's and other personal information from the government servers and other associated authorities. Instead of DigiD, IRMA applies the SSI to authenticate the user. The user is more in control of its identity and can therefore make its own decisions. IRMA also enables users to sign documents or personal-information using their SSI. It's an extended and more privacy-aware solution to DigiD. IRMA is not widely accepted and integrated yet, as organizations should allow IRMA's authentication mechanism. Although IRMA has been designed with privacy in mind, it still requires the identity (and other authorities) to be imported through DigiD (at least once). Another solution for SSI's is Sovrin[6], an international non-profit organization, provides an ecosystem that enables the applicance of SSI's online. Third party developers can create their own SSI application using the services of the identity network of Sovrin. Offline verification is an important aspect since it is more privacy-aware and offers a more robust solution for the use of attestations. Both IRMA and Sovrin are not suited for offline verification of credentials for different reasons [15]. The work of Chotkan [15] provides a distributed attestation revocation for self-sovereign identities. It introduces a revocation mechanism for identities (and their credentials) that are lost or replaced.

Many authorities and institutions are diving into the concept and development of Central Bank Digital Currencies. The European Central Bank (ECB) and national banks of distinct European countries invest into the design of a digital Euro [16]. China already progressed to the final stages of the development of a digital Yuan. Given China's history, it is probably no surprise that the digital Yuan is based on a centralized ledger. Instead of a CBDC that is only issued and backed (and not controlled) by a national bank, the Chinese government has the ability to track and control every digital Yuan, imposing limitations or conditions on its use if necessary. The U.S. is considering potential adoption of a digital dollar [17], although thoughts on the matter are divided.

IV. Design

The most dominant problem of current chat applications is its centralized nature. A decentralized application targets the weak aspects of these applications. No central authority decides what happens with your data. Even less metadata is exposed when sending data. The packets from sender to receiver traverses a network of nodes with a low probability of being stored by a malicious node. And even when it is stored, not much information can be deduced. Communication between two parties become secure and anonymous. Governments no longer have the option to compel user information because the application's developer simply doesn't has the ability. The application's availability is more resistant because of the decentralized nature. Many nodes together make sure that the network remains operational, even when some nodes are offline. Decentralized applications are however limited in terms of freshness of sent data. In a centralized infrastructure data would simply be stored on a central server and fetched when the receiver is online, ensuring an optimal freshness of the data. In a decentralized infrastructure, in case both communicating parties are not connected at the same point in time, no data can be exchanged. Decentralized applications therefore cannot deliver real-time guarantees at all times.

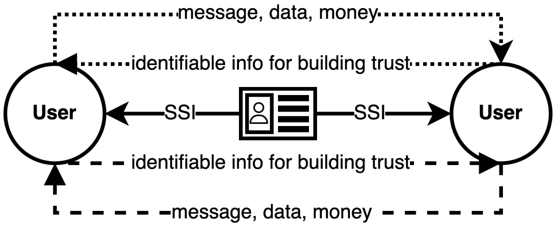
message, data, money
identifiable info for building trust
User
SSI
SSI
User
identifiable info for building trust
message, data, money

Fig. 1. 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 make the decision to trust the contact and send a message, data, or money.

A. EVA Protocol

The designed EVA protocol always tries to move forward, even when mistakes (unacknowledged blocks) are encountered. This mechanism is similar as the principle of piggybacking in TCP, as it does not 'waste' a window and acknowledgement for the unacknowledged blocks. As it increases the performance in general scenarios, it might act in the opposite direction in some cases. Another difference of the EVA protocol is the non-requirement of in-order delivery of packets.

V. Implementation

VI. Useability Study

To confirm how the implementation is experienced by its users an initial usability study is executed. The goal of this study is to improve the unclear parts for the final delivery. The usability study is a task-based approach and the difficulties and time is measured during execution. ...

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

VII. Experimental Analysis and Evaluation

In this section, an experimental analysis is performed to derive the optimal parameters for the designed EVA-protocol of section IV. Additionally, an evaluation using these optimal settings for larger sized transfers to proof 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 acknowledgements (by the receiver) is as low as possible. The latter two constraints contribute to a lower runtime because no extra blocks has to be transmitted and there's no additional idle time waiting for an acknowledgement.

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 [18], 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 than 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 behaviour. 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 informatorial bytes, a safe margin of approximately 200 bytes is chosen.

The window size is defined as the number of bytes (or blocks) the sender can transmit without having to wait for an acknowledgement of receipt, sent by the receiver. Theoretically, a higher window size would directly contribute to higher transfer speeds. A smaller number of acknowledgements has to be sent and received, decreasing the overall idle time. A 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 a high window size and loss due to undelivered blocks. Well-known sliding protocols apply a relative 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 does not have a direct influence on the performance. The retransmit interval may effect 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 acknowledgements to be retransmitted while it is still in transit. For a loosely set interval, the protocol may unnecessarily has to wait for a window or acknowledgement. The overall transfer timeout interval, another parameter, is less critical and will only affect the performance when a window or acknowledgement has been retransmitted and employed all retransmit attempts. The retransmit attempt count likewise has little influence on the performance.

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

Parameter	Values
Block Size (B)	600, 700, 800, 900, 100
Window Size (W)	16, 32, 48, 64, 80, 96, 112, 128
File Size	5MB, 10MB
Executions	5

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. Both phones run the same version of the app and are connected to the same WiFi network. As it is important to catch outliers and produce consistent results, each experiment is executed five times. Also, to verify the independency on 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 such that it can easily 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 I 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 2. We can clearly see the effect of both the block and window size. The runtime remains improving for an 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 acknowledgements. Since the runtime is known, an estimate about the transfer speed can be given. The windows $W = 80$ and $W = 96$ both produce an average transfer speed of approximately 215kB/s.

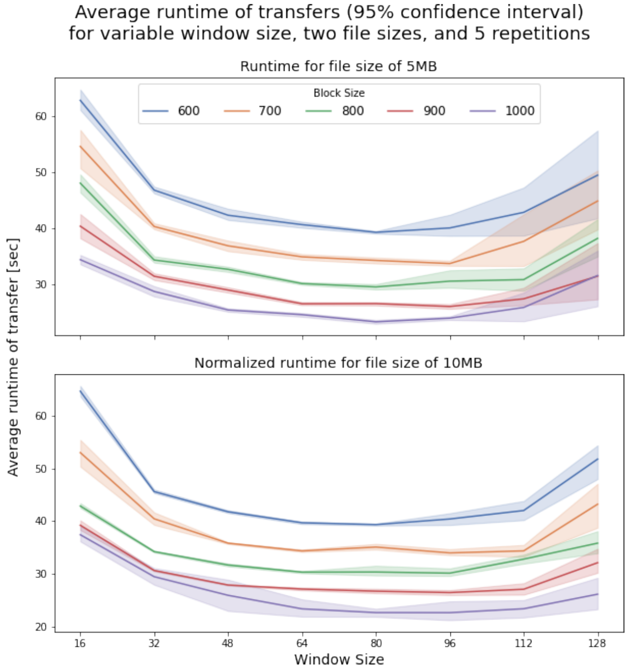

Fig. 2. The runtime of the tests for each window size and block size, and for file sizes of 5MB and 10MB, respectively. The runtime of the latter is normalized to account for the difference in file size. The spread around the lines indicate the 95% confidence interval of the repetitions.

The second requirement, the ratio of unacknowledged blocks during a transfer, is described in the plots of Figure 3. The graphs follows 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 acknowledgements, 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 acknowledgement. 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 4 and 5 the plots of retransmitted windows and acknowledgements 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 are 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.

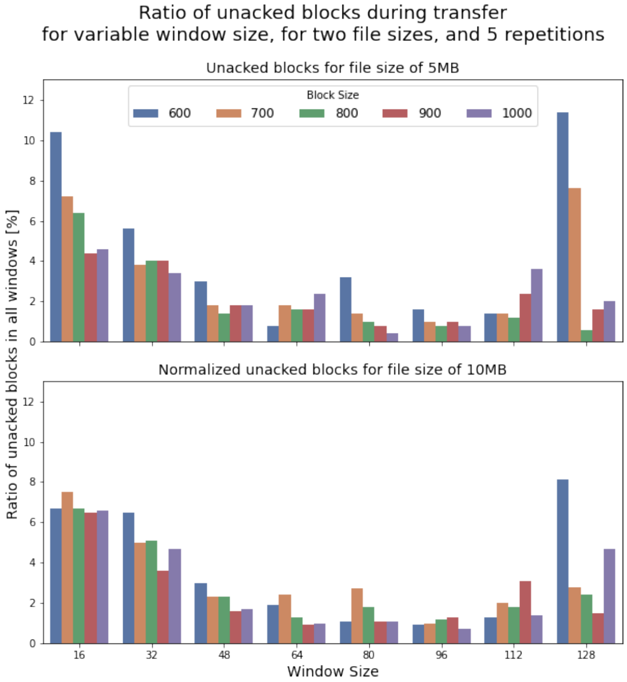

Fig. 3. The ratio of unacknowledged blocks compared to the total number of sent blocks in a transfer for each window size and block size, and for file sizes of 5MB and 10MB, respectively. The ratio of unacknowledged blocks of the latter is normalized to account for the difference in file size.

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 acknowledgement 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.

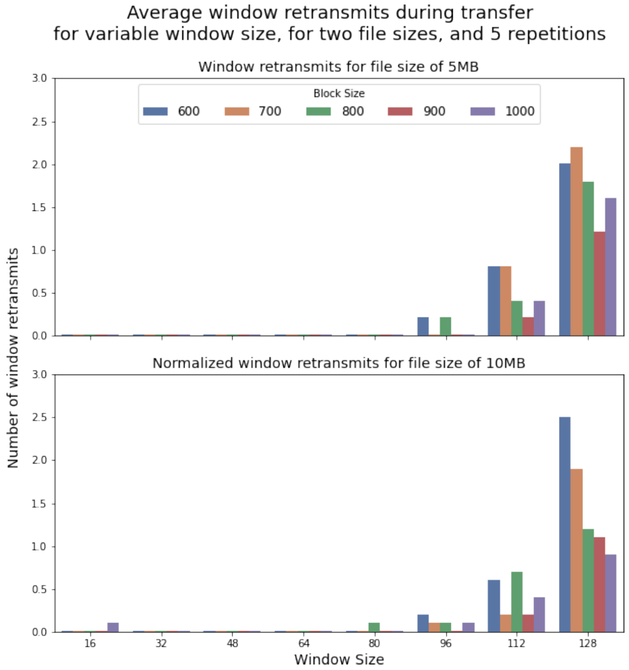

Fig. 4. The number of retransmitted windows for each window size and block size, and for file sizes of 5MB and 10MB, respectively. The number of retransmitted windows of the latter is normalized to account for the difference in file size.

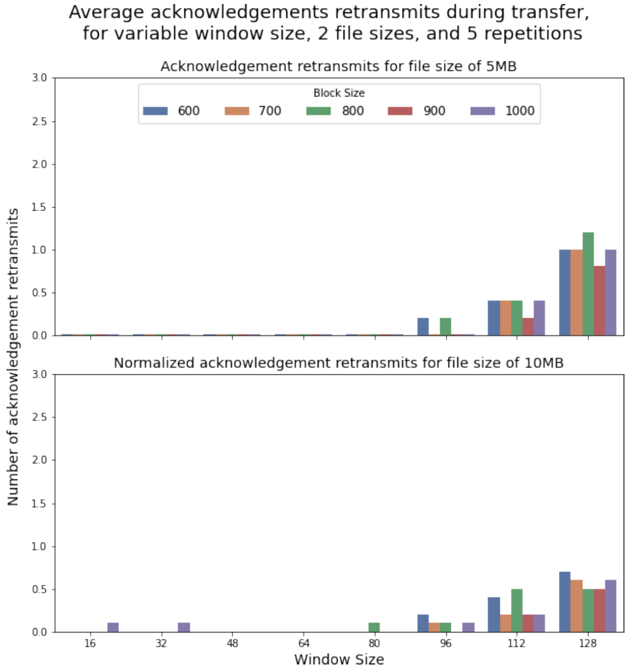

Fig. 5. The number of retransmitted acknowledgements for each window size and block size, and for file sizes of 5MB and 10MB, respectively. The number of retransmitted acknowledgements of the latter is normalized to account for the difference in file size.

TODO: verify independency of file size.

B. *Performance Evaluation*

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

VIII. Conclusion

...

IX. Future Work

- Data vault for more secure storage of private data and identity.
- Updated EuroToken protocol for a more reliable, better scalable, and faster transfer of Euro's.
- Extra features to become a serious contender:
  - group chat
  - phone and video calls
  - live location
  - identity-binded certificates like diplomas and corona certificate
  - biometric security for unlocking sensitive identity information
  - cryptocurrencies support in wallet
- Currently only personal identities, future could include authorities and organizations as well for money transfer

X. Acknowledgements

References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

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

[8] PinDirect. Wat kost een pintransactie? [Online] Available: https://pindirect.nl/kennisbank/

uw-eigen-pinautomaat/wat-kost-een-pintransactie/, 2020.

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

[10] 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.

[11] 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.

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

[13] Rani Molla. Facebook is the least-trusted major tech company. [Online] Available: https://www.vox.com/2018/4/10/17220060/facebook-trust-major-tech-company, 2018.

[14] Logius. Tarieven 2022 voor digid, digid machtigen en mijnoverheid. [Online] Available: https://logius.nl/onze-organisatie/zakendoen-met-logius/doorbelasting, 2021.

[15] 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.

[16] Michiel Bijlsma, Carin van der Cruijsen, Nicole Jonker, and Jelmer Reijerink. What triggers consumer adoption of cbdc? 2021. URL https://www.dnb.nl/media/amwfjgey/working_paper_no-_709.pdf.

[17] Reuters. Analysis: U.s. fed navigates policy minefield with impending digital dollar report. [Online] Available: https://www.reuters.com/business/finance/us-fed-navigates-policy-minefield-with-impending-digital-dollar-report-2021-09-30/, 2021.

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