Towards a decentralized social trust solution to proof-of-address

Mesquita Borba Maranhao M, Suzana; Seigneur, Jean-Marc

Towards a decentralized social trust solution to proof-of-address

Abstract:

The traditional way of proofing someone's address typically relies on using formal documents like a utility bill of a trustworthy local or national institution [1]. Although largely used, this approach may not be viable in some scenarios like people in poverty without official documents. In this poster, we propose an alternative way to proof someone's address using decentralized social trust. We have done an initial validation of this proposal by building a mobile application and by using it with a few users and devices.

CCS Concepts:

- Information Systems -> Information Systems Applications -> Collaborative and social computing systems and tools

Keywords:

Decentralization, Social Trust, Proof-of-address, Certification.

1. Introduction

There are many examples of circumstances when it is necessary to proof the address of someone, e.g., when opening a bank account, registering children in a public school or applying for a job. The traditional way of proofing someone's address often relies on using formal documents like a utility bill or another document issued by an official entity linked to the government sector or a trustful institution in a jurisdiction [1]. This is such a hard prerequisite in some scenarios, like people in poverty without official documents.

There are at least four additional concerns of this approach. First, sometimes only a subset of residents receives all official correspondences making harder for others to proof their address. Second, many paper-based proofs also require a recent document to guarantee that the person still lives in the place. For example, in Brazil it is common to ask for a proof-of-address issued up to 3 months before. Third, when dealing with digital processes, it may be necessary digitalize paper-based documents and an additional image processing is necessary to automatically extract the address. Finally, these documents are also sometimes easily faked, e.g., a user can change an image to forge a proof that he/she resides in a different place.

The requirement of proof-of-address is correlated to financial exclusion. In fact, according to the World Bank from those without formal financial accounts, 25% attribute their exclusion to lacking the necessary documentation such as proof-of-address [2].

This poster discusses alternative ways to proof someone's address and proposes a solution by using decentralized social trust. The proposal is special useful to be applied in some scenarios previously discussed, like people in poverty without formal documents.

The remainder of this poster is divided as follows. Section 2 discusses related work. Section 3 details the technical proposal of this poster while Section 4 explains its initial validation. Section 5 examines conclusion derived from the work and next steps. Finally, the last section presents the references linked throughout the work.

2. Related Work

An alternative way to proof where someone lives without using official documents is a declaration of address. There is a law in Brazil stating that a declaration intended to proof residence, when signed by the interested party or by a sufficient attorney, and under the penalties of the law, is presumed to be true [3]. This approach is linked to a jurisdiction where the law is valid and many times it still requires notarized signatures. This idea of a signed declaration is also used in other countries and programs, like Aadhaar [4].

A second approach is to use location technologies. These technologies can help to determine the location of a device, and authentication technologies may be used in the device to check the person holding it. Considering that the location is not faked, by tracking a device position and doing authentication for some time, it may be possible to infer that the holder of the device usually is in a specific place [5]. One could assume for example that the holder of the device lives where the device usually stays at night.

A third approach is to use blockchain identity software wallets to create an address certificate. A wallet is a software and/or hardware used to generate, manage and store both private and public keys and addresses, which enables DLT (Distributed Ledger Technologies) users to transact [6]. An identity wallet is used to store information linked to a key pair, like proof-of-address or other types of certificates. BlockCerts [7], Kaytrust [8] and RemId [9] are apps that allow the user to request a certificate to a permissioned list of issuers like for example, a university or a government entity. It was not found in these apps a function to request to become a certifier, only a predefined list. Kaytrust and RemId also allow someone to issue a certificate signed with the user's own key, i.e., a user can generate a certificate to himself/herself about some topics, including proof-of-address. These apps save some data on blockchain networks assuring immutability and non-repudiation of the certificates.

The proposed solution by this poster is more similar to the last approach. The main difference is that the list of certifiers is not restricted to a permissioned list managed by the app governance. On the contrary, the proposal of this poster is to create an open ecosystem of certifiers and enable decentralized social trust. This proposal decided not to use blockchain to store the certificates nor their hash as a way to simplify the exchange of certificates avoiding the issue of paying blockchain transactions and enabling offline transactions. Nevertheless, as explained in the future work, the certificates created by this proposal may be useful in different blockchain applications.

3. Implementation

The proposal of this poster was implemented as an application that enables people to certify where other people live by using decentralized trust. The trust is anchored at people with high reputation inside a community, named as Certifier Manager. A Certifier Manager can be for example a priest, a police officer, a bank, a representative of the government or an NGO. This high-trust users delegate trust by creating new certifiers and by issuing certificate for other users.

The main functions of this solution are: (1) to assign a Certifier Manager, (2) to make other people Certifiers, (3) to issue a proof-of-address and (4) to manage their own certificates. These functions are explained below.

The first function is to assign a new Certifier Manager. The proposed solution relies on specific role called Admin to select who should be a Certifier Manager, give him/her a secret code and later approve his/her request to receive a certificate signed with an Ethereum asymmetric cryptographic key pair [10]. Future versions will change the Admin role by fully automating who can be entitled as Certifier Managers using a computational trust engine [11].

The second main function is to make other people Certifiers. There is a set of steps to make this happen in online or offline modes. If both parties (user that wants to be a Certifier and a Certifier Manager) are online, the flow can go in the online mode. The online mode make use of a centralized database with automatic synchronization with the users devices with the use of Firebase Firestore and online connection [12]. If any part is not online, the exchange needs to happen in offline mode. This mode allows people to interact with each other without the need of a central authority and without connectivity. The filled data is saved on the own local device and, whenever possible (in other words, when connectivity is available), synchronized with the remote central server.

The third main function is to issue proof-of-addresses. The description of this function is similar to the second one and may occur in online or offline mode. The main difference is that the user also needs to type his/her own address as part of his/her personal data. A proof-of-address certificate may be issued by any user that has a Certifier or Certifier Manager certificate to a regular user of the mobile app.

This implementation opted for a unique certificate containing name and address. An alternative approach would be to issue two different certificates, one for name and another one for address. The advantage of the first approach is that it is more similar to the traditional utility bills (that contain both name and address) and simpler to deal with. In fact, the Certifier needs to sign only one certificate and the candidate can receive it at once. Having name and address in a unique certificate also minimizes the risk that a certificate issued to one person may be used by someone else. On the other hand, in some scenarios, it may be better to have two different certificates so a user can easily disclose one of them without revealing the other. The mobile app may be easily expanded in the future to include both models.

The mobile app contains the business logic to create the Certifier and Proof-of-Address certificates following a specific format. Both the user who is requesting the certificate and the Certifier have the app installed and they share the same rules (implemented by the mobile app) to create and read certificates in the predefined formats.

The last main function is the user's support to manage their own certificates. The mobile app offers the user the possibility of viewing his/her own certificates as well as the certificates of the Certifiers involved in the signature of his/her certificates. At first, a user does not have in his/her own local storage the information of his/her parent certificates, but the mobile app can fetch the necessary data from the remote server to present it when necessary. After the first fetch, these certificates will also be saved on the local storage of the user device.

The mobile app was coded using React Native, so it can be used in Android and IOS, both cellphones and tablets. Admin functions were implemented using NodeJS. From Firebase, it was used the database Firestore and Statistics modules. An analytic database was created using Google Cloud Big Query.

QRCode [13] was used as a tool to transfer information between two devices, both in online and offline modes. For example, QRCode was used to transfer the content data fields in offline mode when asking a Proof-of-Address. Many fields are grouped together in a string that can be parsed to a JSON format. QRCodes were used the default size of 256 x 256, the same used as default configuration in easyqrcode [14] and in the paper written by Jean-Marc et al. [15]. Bigger size would be problematic for visualization on small-screen devices.

The following Figure shows in the left side the app of a user that has a certificate of Certifier Manager and of his/her Proof-of-Address. In the right side, the user is asking another Proof-of-address certificate using the offline mode.

4. Initial Validation

The first initial tests were executed using the React Native simulator Expo Go[1] on an iPhone 11, an iPhone X_R and an Android LG K22 cellphone.

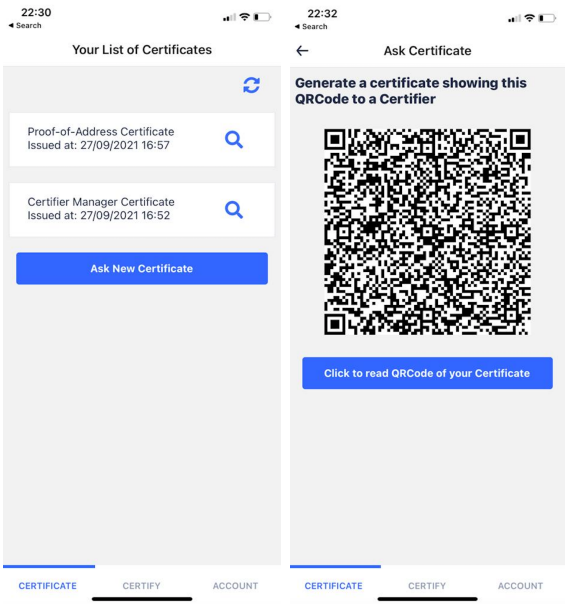

Figure 1:List of certificates (left) and QRCode representing a request of a new certificate (right).

The main issue found was the long time to create the cryptographic key pair when first loading the application. This issue was minimized by presenting a specific screen with a spinner together with an explanatory message. Once created, the private key of the user is saved locally is loaded every time the user reloads the application.

Initial tests confirmed that the more information is coded in the QRCode format, the more difficult it is to work in practice. For example, the longer the time to generated it. Some issues were found when using more than 500 characters on the Android device with the proposed size and referenced lib. Tests also showed that more than 300 characters introduce a significative delay mainly when generating a QRCode on the Android device. This result was consistent with some QRCode generators that advise using QRCodes with less than 300 characters [16] [17]. The QRCodes source data were reviewed in order to make sure that data transferences contain fewer characters than this maximum reference of 300.

After correcting some bugs, the tests could be performed using the three devices in all functions described in the Section 3, including both online and offline modes. A second set of tests was executed after using Google Play Developer[2] to have a production-ready version of the app. There was not significant difference on the app performance after installing the app on the Android device (instead of

[1] https://expo.dev/client

[2] https://console.developers.google.com/

executing it inside Expo Go simulator).

To complete the validation of this work, we are carrying out an experiment involving a Priest and twenty more people in a Brazilian favela called Rocinha, in Rio de Janeiro city. We have already interviewed the chosen Priest, from the Church called *Nossa Senhora Aparecida*. He explained that the Church should help to improve the access to public services and to create a volunteer solidarity network. He is interested to be a Certifier Manager and he also believes that other Priests would be interested to participate as well.

5. Conclusion and Next Steps

In this poster, we proposed a solution to proof someone's address using decentralized social trust. It is an alternative approach of traditional options and useful mainly when applied to poor people without formal documents. We also did an initial validation of this proposal, by building an app and testing it in different devices and with a few users.

Besides improving the validation by testing the app with more users and devices, we intend to interview some people to better understand some aspect of this research. For example, who could perform the Certifier Manager role in real life and how to extend the existing solution to other types of certifications. It is also in the roadmap of our work the decentralization of the Admin's role and the investigation of how computational trust can be used to achieve this goal. Finally, the choice of Ethereum-compatible key pair in the solution of this poster opens new ways to expand the proposal in at least two ways. First, to expand the app to send transactions containing the certificates to blockchain applications. In this direction, a careful software and a legal design are needed to deal with the privacy issues. Second, to create business models to motivate certifiers to participate in the solution or to measure their reputation linked to cryptotokens.

Reference

[1] S. M. B. M. Moreno, J.-M. Seigneur, and G. Gotzev, "A Survey of KYC/AML for Cryptocurrencies Transactions," in Handbook of Research on Cyber Crime and Information Privacy, 2020. Accessed: Oct. 03, 2021. [Online]. Available: https://www.igi-global.com/chapter/a-survey-of-kycaml-for-cryptocurrencies-transactions/261722

[2] B. Cooper, A. Esser, and M. Allen, "The use cases of central bank digital currency for financial inclusion: A case for mobile money," Jun. 2019. Accessed: Oct. 03, 2021. [Online]. Available: https://cenfri.org/wp-content/uploads/2019/06/CBDC-and-financial-inclusion_A-case-for-mobile-money.pdf

[3] J. Figueiredo, I. Arbi-Ackel, and H. Beltrão, "LEI No 7.115," Aug. 29, 1983. http://www.planalto.gov.br/ccivil_03/leis/l7115.htm (accessed Oct. 03, 2021).

[4] "UIDAI," Unique Identification Authority of India | Government of India. https://uidai.gov.in/ (accessed Feb. 26, 2020).

[5] Incognia, "Privacy policy | Incognia," Jan. 11, 2021. https://www.incognia.com/policies/incognia-policy (accessed Oct. 03, 2021).

[6] ITU-T, "X.1400 : Terms and definitions for distributed ledger technology," 2020. Accessed: Oct. 04, 2021. [Online]. Available: https://www.itu.int/rec/T-REC-X.1400

[7] Blockcerts, "Blockchain Credentials," Blockcerts. http://blockcerts.org/ (accessed Oct. 03, 2021).

[8] "KayTrust - Manage digitals identities of your customers." https://www.kaytrust.id/ (accessed Oct. 03, 2021).

[9] "REM ID." https://wdi.net/rem/ (accessed Oct. 03, 2021).

[10] W. Gavin, "Ethereum: a secure decentralised generalised transaction ledger istanbul version." Oct. 04, 2021. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[11] S. Jean-Marc, "Trust, Security and Privacy in Global Computing," University of Dublin, 2005.

[12] "Cloud Firestore | Firebase Documentation." https://firebase.google.com/docs/firestore (accessed Oct. 04, 2021).

[13] W. Denso, "QRCode." https://www.qrcode.com/en/codes/ (accessed Oct. 04, 2021).

[14] EasyQRCode React Native. 2021. Accessed: Oct. 04, 2021. [Online]. Available: https://github.com/ushelp/EasyQRCode-React-Native

[15] S. Jean-Marc, L. Carlos, and M. Alfredo, "Secure User-Friendly Wi-Fi Access Point Joining," presented at the International Wireless Communications and Networking Conference, Shanghai, 2013. Accessed: Oct. 04, 2021. [Online]. Available: https://archive-ouverte.unige.ch/unige:55387

[16] "QR Code Generator." https://goqr.me/ (accessed Oct. 04, 2021).

[17] "QRCode Monkey," QRCode Monkey. https://www.qrcode-monkey.com (accessed Oct. 04, 2021).