Safepaths: Vaccine Diary Protocol and Decentralized Vaccine Coordination System using a Privacy Preserving User Centric Experience

Abhishek Singh[1] and Ramesh Raskar[1]

[1]MIT Media Lab

December 16, 2020

Abstract

In this draft, we present an end to end decentralized protocol for the secure and privacy preserving workflow of vaccination, vaccination status verification, and adverse reactions or symptoms reporting. The proposed system improves the efficiency, privacy, equity and effectiveness of the existing manual system while remaining inter-operable with its capability. We also discuss various security concerns and alternate methodologies based on the proposed protocols.

1 Introduction

Motivation: Recent announcements of vaccines have created a sense of hope for the near future of the society currently burdened with lock-downs and quarantines. However, a lot of effort and coordination is required to go from successful vaccines to successful vaccination programs in order to curb the disease spread. We believe a user-centric design using vaccination cards and/or mobile phones can play a critical role in the micro-planning and last mile issues. In this work we integrate work in user privacy, cryptography and user interaction to design secure and private protocols which span from the starting step of vaccination program enrolment to all the way to symptoms reporting from potential side effects of vaccination.
We consider first four of the following parts to the system:

- Indicating eligibility based on priority tiers (anonymity via vaccine coupons),
- Second dose coordination (privacy preserving record linkage),
- vaccine verification/passports (interoperability and privacy),
- Safety.efficacy monitoring (crowdsourced monitoring of safety and efficacy using private aggregation)
- Trust and communication (social media analytics, contextual messaging)

Description: First we describe different participants and their corresponding roles.

- *Issuer* is a trusted entity that initiates the enrolment process by distributing the coupons which would be eventually used for getting vaccinated as described below. In our use-case the *issuer* could be CDC or any similar authoritative body that currently monitors the distribution of the vaccines.
- *Distributor* is the entity that receives the vaccination coupons from the *issuer* and distributes it locally to individuals. In our use-case the *distributor* could be the city government office.

- *Holder* is the user who wants to get vaccinated and use their vaccination status later on to obtain a vaccination passport that will be used as a proof of vaccination status.
- *Coordinator* is the entity that performs the vaccination. This could be nearby health/pharmacy store like CVS.
- *Verifier* refers to the set of authorized users that can verify the vaccination status of any *holder* after their consent. This can be a venue owner that is managing access to their facilities.

2 Methodology

We use existing and well studied cryptographic building blocks in the following protocols for performing operations such as sign and verify which can be built using digital signature scheme such as DSA [9], schnorr signature [11] and others. We further use $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$ operations that can be performed using asymmetric key cryptography protocols like el-gamal [6]. We refer the reader to see [8] for more detail on different asymmetric key cryptosystems and different aspects of their security. Now we describe the three key stages of our protocol:

2.1 Eligibility and Vaccine Coupon Certification

Description: The certification protocol includes obtaining vaccination enrolment from the *distributor*, getting vaccinated, and obtaining proof of vaccination status from the *issuer*.

Issuer (e.g. CDC) creates a finite number of vaccine coupons to be rationed, x_i, and sends them to distributor (e.g. a city government). The holder (end user) can receive this either electronically or as a printed QR code on vaccination card.

Protocol:

- *Issuer* takes k for a given region; generates a set of coupons $\mathbb{X}$ by generating a sequence of numbers and signing them as $(x_1, \mathsf{sign}(x_1)), (x2, \mathsf{sign}(x_2)), ..., (x_n, \mathsf{sign}(x_n)) := PRG(k)$. Here PRG is a pseudorandom number generator.
- *Issuer* gives $\mathcal{X}$ to the distributor.
- The *distributor* gives x_i to the *holder* i.
- The *holder* verifies the coupon by performing $\mathsf{verify}(x_i, \mathsf{sign}(x_i))$ locally.
- Upon receiving x_i, the *holder* registers the coupon with their name or any other unique information. We refer this personal information as h_i. *Holder* commits to this information to their coupon by uploading $k_i = x_i^{h_i||s_i}$ to a *registration server*, here s_i is a secret used by the holder to ensure sufficient entropy of the exponent. Note that to make the protocol truly decentralized, the *registration server* should be independent of the *issuer* server.
  - **App based solution:**
  - **Physical vaccine card solution:**
- The *registration server* verifies x_i and $\mathsf{sign}(x_i)$ and marks x_i as registered if it has not been registered yet in the database.

2.2 First and Second dose linkage

Description: After receiving the vaccine card from the *distributor* and registering it with the server, the *holder* proceeds for the vaccination stage. In our proposed

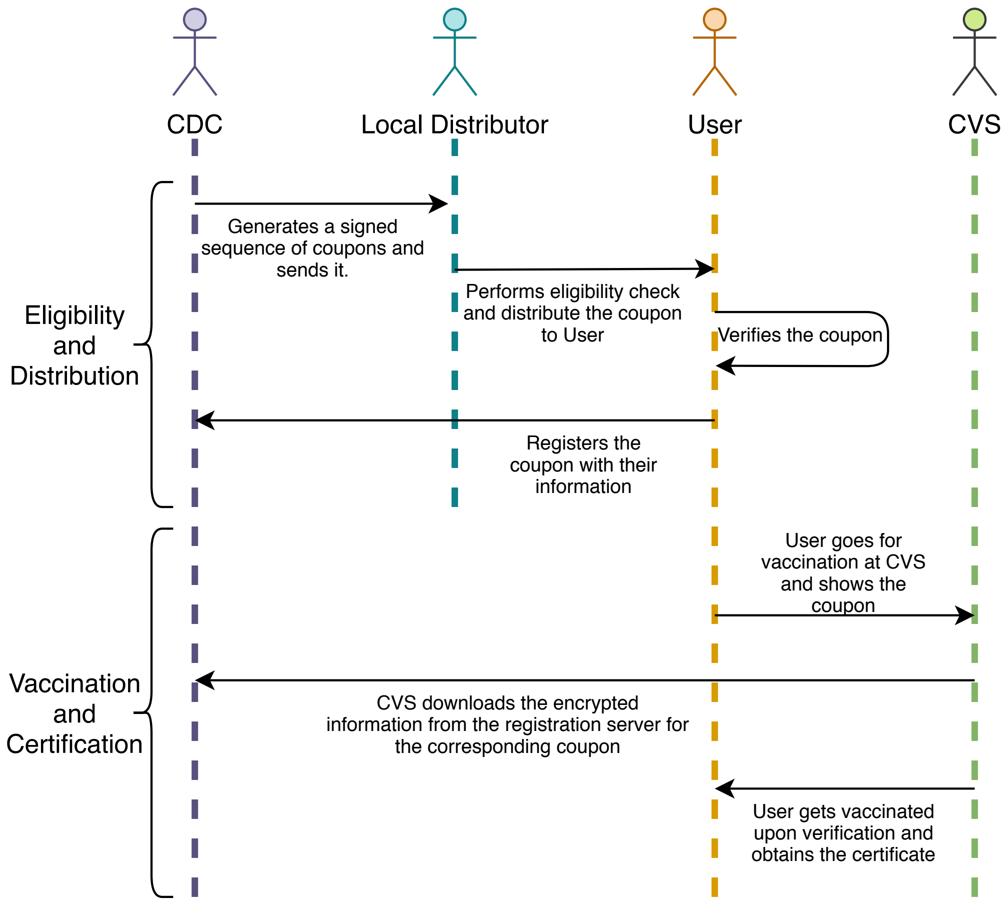

Figure 1: Sequence diagram depicting two stages before a user gets vaccinated. The first stage involves eligibility evaluation and distribution of the vaccination coupon. In the next stage, the user gets vaccinated and obtains the certificate which would be used later for second dosage reminder, status verification, and health outcome monitoring.

Protocol:

- Now that the *holder* is registered, they go to the *Coordinator* for getting vaccinated and show x_i.
  - **App based solution -** show x_i as a QR code to the *verifier*.
  - **Physical Vaccine Card solution -** give the coupon code x_i
- The *coordinator* uploads x_i to the *registration server* and obtains their commitment k_i.
- The *coordinator* then verifies the identity of the *holder* through driving license or some other means of verifying the identity.
- The *issuer* computes $cert = \mathsf{sign}(x_i || \text{``}dose\_num\text{''})$ and returns it to the *verifier*. The value "dose_num" is based on whether it is the first dose or the second dose.
- The *holder* gives certificate *cert* to the *holder*.
  - **App based solution -** scan the QR code shown by the verifier
  - **Physical vaccine card solution -** Print the digital signature as a QR code and give it to the user

2.3 Verification and Vaccine Passport

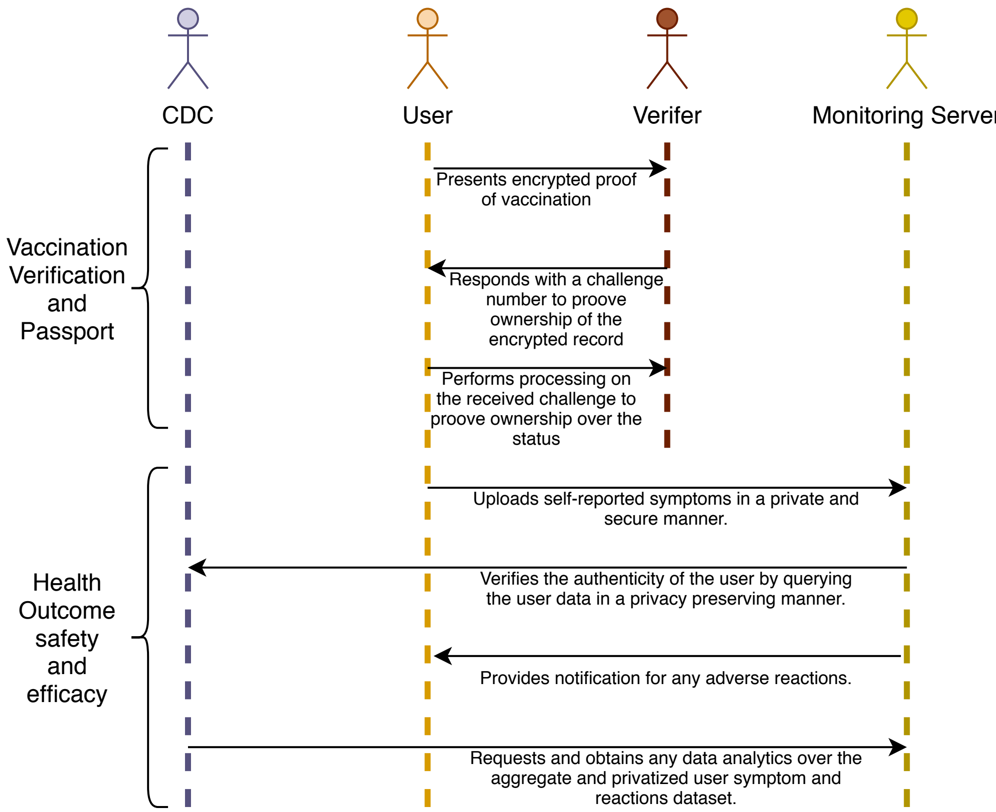

Figure 2: Depiction of the two stages post-vaccination: Vaccination verification and Monitoring of the data. In the first stage the user can use their vaccination status for the places where vaccination status might be mandatory like air travel and etc. In the next stage, the user can report any specific condition or symptoms without leakage of any private information. Furthermore, the user and the CDC can use the system to monitor and analyze the safety and efficacy concerns associated with a given vaccine.

Description: The verification of health status would be based on two verification phases - verification of the integrity of certificate and verification of the identity of the user. The integrity verification happens through digital signature scheme described below and the identity verification happens through personal information verification which can be embedded in the certificate and hence checked manually or the identity verification can be performed by app level biometric security. We propose two protocols for the verification method - one is based on the challenge-response and the other is based on single communication verification. The challenge-response protocol described here is based on Schnorr signature [11]. For the single communication server based verification, more details of the exact cryptographic computation, we refer the user to Singh et al. [12]

Single Communication Verification

- The *holder* presents their x_i and *cert* to the verifier.
- The *verifier* verifies the health status integrity by performing $\mathsf{verify}(cert)$ on the user i's record. The verifier then verifies the identity of the user by obtaining k_i from the registration server.

Contact-less Verification In the following we describe an interactive protocol that takes place wirelessly using channels like bluetooth, NFC and other commonly available sensors on smartphones. Every user performs one message exchange with the *verifier* system and after this message exchange, everyone should have a value k on their device that they can show to obtain access.

- The *holder* send their certificate *cert* signed by their private key. i.e. *holder* sends $c = (\mathsf{Encrypt}(cert, \mathsf{sk_i}), \mathsf{pk_i})$
- The *verifier* decrypts c by $cert = \mathsf{Decrypt}(c, \mathsf{pk_i})$ and verifies the integrity.
- *verifier* sends back the challenge by broadcasting $c = \mathsf{Encrypt}(k, \mathsf{pk_i}), \mathsf{pk_i}$
- The *holder* identifies packet destined for them by matching $\mathsf{pk_i}$ and performs $\mathsf{Decrypt}(c, \mathsf{sk_i})$ to obtain k.

2.4 Assessing Health Outcomes of Safety and Efficacy

The health outcome assessment requires monitoring and reporting from the user standpoint as well as the vaccine provider standpoint. In the following section we describe how we merge the bottom-up and top-down approaches of health outcome assessment is performed.

Upload: *User* can upload symptoms data at any point using their usercoupon ID (x_i). The *issuer* can validate the upload using the usecoupon number and associate it with a verified vaccination by the *coordinator*. *User* can also ask their doctor to submit adverse reaction report using the same usecoupon ID. The upload of symptoms can be made privacy preserving by aggregating using multi-party computation [3, 2] based aggregation method on top of which differential privacy based mechanisms [5, 4] can be used to ensure privacy of the individuals over the aggregate statistics.

Download: How can the user get an alert if their dosage batch is faulty? Or users with specific health conditions maybe at risk? We want to achieve this without the need for user to reveal everything about themselves. Similar to GAEN [7] key server, the user app downloads the *adverse events data report* that is public for their state every morning. The apps checks if their own dose batch (company, batch or vaccination site) has any public alerts. The app also checks if there is a specific alert for their health condition (e.g. vaccine may have adverse reaction to certain food allergy or immune health conditions)

Aggregate view: How can vaccine maker, a US state or CDC have detailed or aggregate view? (i) With anonymity, a user can be tracked using usercoupon. If we do not want usercoupon to be tracked across the user journey, the user can upload the symptoms without usecoupon ID. (ii) If user does not wish to upload symptoms in the raw, we describe a protocol to use secure multi party computation to provide aggregates statistic without revealing privacy of any individual. (iii) For users without an app, they can log into V-SAFE [1] or VAERS [13] system using their 16 digit usercoupon ID. Similarly, a doctor updating adverse reaction report can use the usercoupon ID.

3 Downsides and Attacks

Digital tools for pandemic response can have privacy and ethics issues at various fronts [10], therefore we discuss some of the issues and potential pitfalls for the proposed technology. The protocol provides anonymity but not privacy if the user has to interact with the system. The ability to track usercoupon ID x_i for any user i provides pseudoanonymity which is not a full proof notion of privacy. Because health services and verified access requires human interaction, we expect systems will require user to furnish a state-ID. So the name or some identifying information needs to be embedded as part of the QR code. This can be solved by letting the user encrypt their name with their own private key before uploading to the *registration server*. However, these systems can be made more secure by storing the information in a secure enclave on the server and making the network logs auditable. However, these system layers security would not prevent the aforementioned worst case attack. In our proposed protocol, the *distributor* knows the information about the user, and hence in the worst case, can collude with the *issuer* or the *reigstration server* to reveal personal information and identify a given user.

References

[1] CDC. Ensuring the safety of covid-19 vaccines in the united states. *Monitoring, Vaccine Safety*, 2020.

[2] Henry Corrigan-Gibbs and Dan Boneh. Prio: Private, robust, and scalable computation of aggregate statistics. In *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*, NSDI'17, page 259–282, USA, 2017. USENIX Association.

[3] George Danezis, Cédric Fournet, Markulf Kohlweiss, and Santiago Zanella-Béguelin. Smart meter aggregation via secret-sharing. In *Proceedings of the First ACM Workshop on Smart Energy Grid Security*, SEGS '13, page 75–80, New York, NY, USA, 2013. Association for Computing Machinery.

[4] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51, 2016.

[6] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, pages 10–18, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[7] Google. Apple and google: Exposure notifications: Using technology to help public health authorities fight covid-19, 2020. `https://www.google.com/covid19/exposurenotifications`.

[8] Neal Koblitz and Alfred J. Menezes. A survey of public-key cryptosystems. *SIAM Rev.*, 46(4):599–634, April 2004.

[9] Thomas Pornin. Deterministic usage of the digital signature algorithm (dsa) and elliptic curve digital signature algorithm (ecdsa). *Internet Engineering Task Force RFC*, 6979:1–79, 2013.

[10] Ramesh Raskar, Isabel Schunemann, Rachel Barbar, Kristen Vilcans, Jim Gray, Praneeth Vepakomma, Suraj Kapa, Andrea Nuzzo, Rajiv Gupta, Alex Berke, Dazza Greenwood, Christian Keegan, Shriank Kanaparti, Robson Beaudry, David Stansbury, Beatriz Botero Arcila, Rishank Kanaparti, Vitor Pamplona, Francesco M Benedetti, Alina Clough, Riddhiman Das, Kaushal Jain, Khahlil Louisy, Greg Nadeau, Vitor Pamplona, Steve Penrod, Yasaman Rajaee, Abhishek Singh, Greg Storm, and John Werner. Apps gone rogue: Maintaining personal privacy in an epidemic, 2020.

[11] C. P. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, 1989.

[12] Abhishek Singh and Ramesh Raskar. Verifiable proof of health using public key cryptography. *arXiv preprint arXiv:2012.02885*, 2020.

[13] Weigong Zhou and Susan S Ellenberg. Surveillance for safety after immunization: Vaccine adverse event reporting system (vaers)—. *Morbidity and Mortality Weekly Report: MMWR. Surveillance Summaries. Surveillance summaries*, 2003.