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 21, 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.

- *User* is the person 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 [11], schnorr signature [13] and others. We further use Encrypt and Decrypt operations that can be performed using asymmetric key cryptography protocols like el-gamal [8]. We refer the reader to see [10] for more detail on different asymmetric key cryptosystems and different aspects of their security.

Key components of security: At every stage of the proposed protocol we validate two key components from the security standpoint - message integrity and identity. Message integrity validation involves checking a message's authenticity, forgeability and other integrity related aspects. The message itself is usually a certificate provided by a certified issuer. The next aforementioned key component is the identity. Validating this component means validating whether the is indeed a person who they are claiming to be. Validating this claim ensures that a does not obtain vaccine on someone else's behalf and is not able to verify their vaccination status using someone else's certification. This would require either embedding the 's personal information in the token or using a secure enclave [1] where the private key is only accessible to a trusted app for encryption and not even the *user* themselves, since they can potentially use distribute it to someone else.

Vaccine stickers: The *user* carries three pieces of information with them which we represent as vaccine stickers. Since these three stickers can be serve different purposes at different venues. For a smartphone app *user* it appears as a qr-code in the app while for a vaccine card based solution, it will be three different papers with a qr-code in them. The three stickers are as follows:

- vaccine coupon sticker: The coupon sticker is generated by the *issuer* and shared by the *distributor* to the *user*.
- vaccination record sticker: The record sticker is obtained post-vaccination and used as a proof of vaccination .
- vaccination certificate: The vaccination certificate contains a secret signature of the *user* information that can be only linked to the *user* if the *user* provides their secret.
  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 user (end) can receive this either electronically or as a printed QR code on vaccination card.

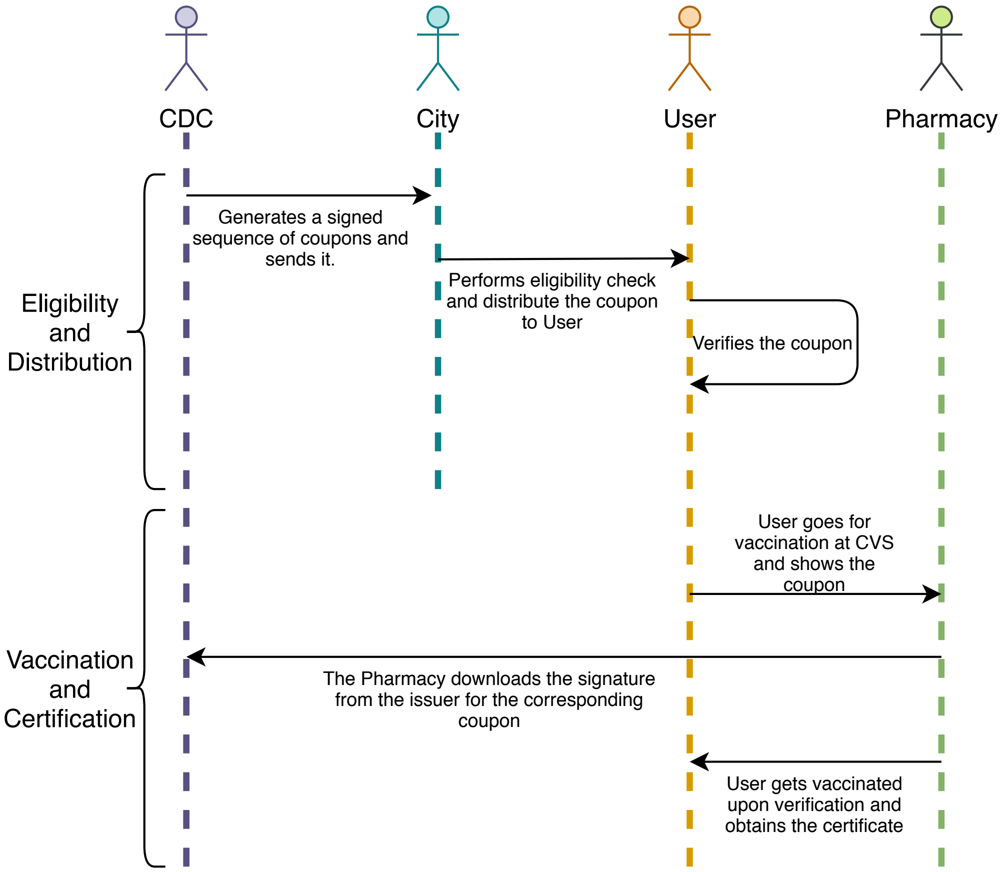
CDC
City
User
Pharmacy
Eligibility and Distribution
Generates a signed sequence of coupons and sends it.
Performs eligibility check and distribute the coupon to User
Verifies the coupon
Vaccination and Certification
User goes for vaccination at CVS and shows the coupon
The Pharmacy downloads the signature from the issuer for the corresponding coupon
User gets vaccinated upon verification and obtains the certificate

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

Protocol:

- *Issuer* takes a unique g for a given region with location information ℓ of the distributor and generates a set of n coupons $\mathbb{X}$ by generating a sequence of numbers $\{g_1, g_2, ..., g_n\} := PRNG(g)$, Here $PRNG$ is a pseudorandom number generator [2]. Then the issuer concatenates them with ℓ as $\{g_1||\ell, g_2||\ell, ..., g_n||\ell\}$ and signing them as $\{\mathsf{sign}(g_1||\ell), \mathsf{sign}(g_2||\ell), ..., \mathsf{sign}(g_n||\ell)\}$. This is finally represented as the set $\mathbb{X} = \{x_1, x_2, ..., x_n\}$ where $x_i = (g_i||\ell, \mathsf{sign}(g_i||\ell)$.
- The *issuer* gives $\mathcal{X}$ to the distributor.
- The *distributor* gives x_i to the *user* i based on the eligibility requirement in their jurisdiction.
- The *user* validates the coupon by performing a check $\mathsf{verify}(g_i||\ell, \mathsf{sign}(g_i||\ell))$ locally.

2.2 First and Second dose linkage

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

Protocol:

- The *user* goes to the *pharmacy* 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 *pharmacy* uploads x_i to the *issuer* server with the *pharmacy* specific information which would be useful for the *issuer* for the analytics purpose.
- The *pharmacy* then validates the identity of the *user* through driving license or some other means of validating the identity. Let us call this uniquely identifying and private information as "user_info".
- The *pharmacy* uploads the non-private user vaccination record = $x_i||$"dose_info" to the *issuer.* The value "dose_info" is dosage information associated with the vaccine.
- The *issuer* signs the request and returns signed_record = $\mathsf{sign}(x_i||$"dose_info"$)$ and returns it to the *pharmacy.* The *pharmacy* combines the record and the signed record to form "vaccination record sticker" for the *user.*
- Next, the *pharmacy* computes $c = x_i||\mathsf{hash}($"user_info"$||$"secret_key"$)$ and sends it to the *issuer.* The *issuer* signs the request and returns the signature. The *pharmacy* combines c and $\mathsf{sign}(c)$ to form "vaccination certificate" and give it to the user. The "secret_key" is a secret number with sufficient high entropy so as to prevent brute-force attack by the *issuer* on the c.
  - **App based solution -** scan the "vaccination record sticker", "vaccination certificate", and "secret_key" as a QR code shown by the *pharmacy*
  - **Physical vaccine card solution -** Same information is provided to the user as above but through a printed QR code.
- The *user* can present the vaccination record sticker and the vaccination certificate to obtain the second dose.

2.3 Verification and Vaccine Passport

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 . 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 [13]. For the single communication server based verification, more details of the exact cryptographic computation, we refer the reader to Singh et al. [14]

Single Communication Verification

- The *user* presents their x_i and "vaccination certificate" to the verifier either using the app or the vaccine card.
- The *verifier* verifies the health status integrity by performing $\mathsf{verify}($"vaccination"$)$ on the i's record. The verifier then verifies the identity of the *user* by obtaining "secret_key" and verifying the $\mathsf{hash}($"user_info"$||$"secret_key"$)$ locally.

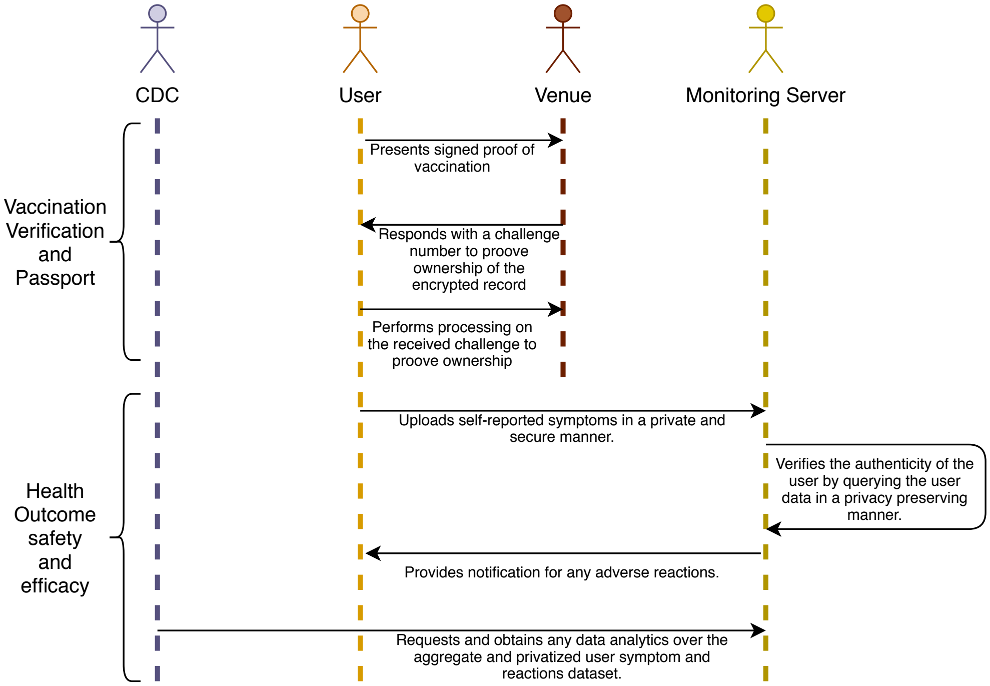
CDC
User
Venue
Monitoring Server
Vaccination Verification and Passport
Presents signed proof of vaccination
Responds with a challenge number to proove ownership of the encrypted record
Performs processing on the received challenge to proove ownership
Health Outcome safety and efficacy
Uploads self-reported symptoms in a private and secure manner.
Verifies the authenticity of the user by querying the user data in a privacy preserving manner.
Provides notification for any adverse reactions.
Requests and obtains any data analytics over the aggregate and privatized user symptom and reactions dataset.

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

Contact-less Group Verification The high-level idea of the contact-less group verification is that the user carries smartphone with them and the *verifier* share a secret number on the wireless channel such that only the users with correct vaccination status can present this secrent number on their screen and walk-in. In the following we describe an interactive protocol that takes place wirelessly using channels like bluetooth, NFC and other commonly available sensors on smartphones. The *verifier* system generates unique and unguessable k repeatedly for small time windows. 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. Here k can be a number, color or an image the user will eventually show to the verifier visually. The challenge k can change continuously. For example, a guard at the venue can change k every minute.

- The *user* send their vaccination certificate *cert* and "user_info"||"secret_key" signed by the *verifier's* public key. i.e. *user* sends $c_1 = \mathsf{Encrypt}(cert, \mathsf{pk_i})$
- The *verifier* decrypts c_1 by $cert = \mathsf{Decrypt}(c_1, \mathsf{pk_i})$ and verifies the integrity of the message.
- *verifier* sends back the challenge k by broadcasting $c_2 = \mathsf{Encrypt}(k, \text{"secret\_key"}), \text{"user\_info"}$
- The *user* identifies packet destined for them by matching "user_info" and obtains $k = \mathsf{Decrypt}(c_2, \text{"secret\_key"})$.
- The *user* then shows the value k on the phone to the verifier.

2.4 Assessing Health Outcomes of Safety and Efficacy

The health outcome assessment requires monitoring and reporting from the 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: can upload symptoms data at any point using their coupon ID (x_i). The *issuer* can validate the upload using the usecoupon number and associate it with a verified vaccination by the *pharmacy.* 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 [5, 4] based aggregation method on top of which differential privacy based mechanisms [7, 6] can be used to ensure privacy of the individuals over the aggregate statistics.

Download: How can the get an alert if their dosage batch is faulty? Or s with specific health conditions maybe at risk? We want to achieve this without the need for to reveal everything about themselves. Similar to GAEN [9] key server, the 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 can be tracked using coupon. If we do not want coupon to be tracked across the journey, the can upload the symptoms without usecoupon ID. (ii) If 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 s without an app, they can log into V-SAFE [3] or VAERS [15] system using their 16 digit coupon ID. Similarly, a doctor updating adverse reaction report can use the coupon ID.

3 Downsides and Attacks

Digital tools for pandemic response can have privacy and ethics issues at various fronts [12], therefore we discuss some of the issues and potential pitfalls for the proposed technology. The protocol provides anonymity but not privacy if the has to interact with the system. The ability to track coupon ID x_i for any 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 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 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 , and hence in the worst case, can collude with the *issuer* or the *reigstration server* to reveal personal information and identify a given .

References

[1] Apple. Storing Keys in the Secure Enclave | Apple Developer Documentation, 2020. `https://developer.apple.com/documentation/security/certificate_key_and_trust_services/keys/storing_keys_in_the_secure_enclave`.

[2] Elaine Barker and John Kelsey. Recommendation for random number generation using deterministic random bit generators. `https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final`.

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

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

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

[11] 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.

[12] 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.

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

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

[15] 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.