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

Abhishek Singh[1], Ramesh Raskar[1], and Anna Lysyanskaya[2]

[1]MIT Media Lab
[2]Brown University

February 10, 2021

**Abstract**

In this early 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 capabilities. 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 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 the first four of the following parts of 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 and efficacy monitoring (crowdsourced monitoring of safety and efficacy using private aggregation)
- Trust and communication (social media analytics, contextual messaging)

**Participants:** Let us consider the 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 later. In our use-case the *issuer* could be CDC or any similar authoritative body that currently monitors the distribution of the vaccines.

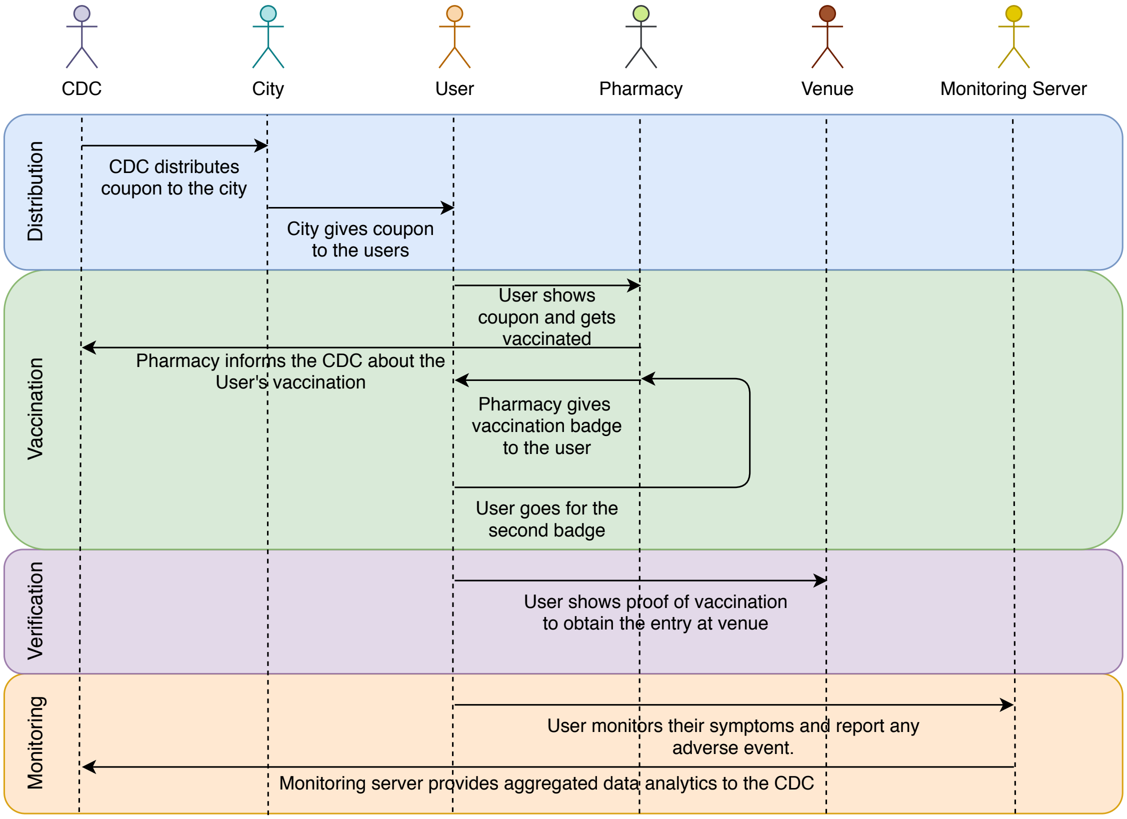
CDC
City
User
Pharmacy
Venue
Monitoring Server
Distribution
Vaccination
Verification
Monitoring
CDC distributes coupon to the city
City gives coupon to the users
User shows coupon and gets vaccinated
Pharmacy informs the CDC about the User's vaccination
Pharmacy gives vaccination badge to the user
User goes for the second badge
User shows proof of vaccination to obtain the entry at venue
User monitors their symptoms and report any adverse event.
Monitoring server provides aggregated data analytics to the CDC

Figure 1: Entity and information flow depicting the system at a high level. We project this diagram as the backbone of the proposed system. We describe the detailed information flow and privacy preserving mechanisms in the next figures.

- *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.
- *Pharmacy* is the entity that performs the vaccination. This could be nearby health/pharmacy stores like CVS.
- *Verifier* refers to the set of authorized users that can verify the vaccination status of any *user* after their consent. This can be a venue owner that is managing access to their facilities.

**Phases:** We model the vaccination process as consisting of the following phases:

- *Setup and registration phase* in which the entities establish a trust infrastructure. In particular, the trusted Issuer generates its cryptographic keys and publishes the public half of those keys. Other participants that require keys generate their keys as well and obtain certificates on the public parts of their keys. For example, a distributor may want to be able to digitally sign a statement such as “Alice is eligible for vaccination." Additionally, pharmacies need a way to authentically communicate the fact that Alice has been vaccinated.
- *Coupon distribution phase* in which the Issuer and Distributor jointly create, for each eligible user $U_i$ a string $c_i$ that constitutes the user's vaccination coupon, and communicate it to the user.

- *Vaccination phase one* in which a User uses his coupon $c_i$ possibly together with other forms of identification (or biometrics) to convince a Pharmacy of his eligibility to receive the vaccine. If the pharmacy determines that the coupon is valid for this user, the user receives a first dose of the vaccine together with a badge $b_i$ that can be used as proof of vaccination.
- *Vaccination phase two* in which a User uses his badge $b_i$, possibly together with other forms of identification (or biometrics) to convince the Pharmacy that he's eligible for a second shot of the vaccine. If the pharmacy determines that the badge is valid for this user, the user receives the second shot of the vaccine together with an updated badge $b_i$ that can be used as proof of vaccination.
- *Verification phase* in which a User uses his badge $b_i$, possible together with other forms of identification or biometrics to convince a Verifier of his vaccination status.
- *Gathering useful data phase* A user who has been vaccinated can use his badge $b_i$ to contact the issuer in case of adverse effects or to report other helpful information about the efficacy of the vaccine.

### 1.1 Security, privacy, and usability considerations

Various considerations:

**Unforgeability of coupons** No adversary, even if he controls the distributor, can create coupons without help from the issuer.

**(Optional) Non-transferability of coupons** If both the issuer and the distributor are honest, then a coupon issued to user $i$ cannot be used by any other user.

**Unforgeability and non-transferability of badges**

**(Trade-off) Off-line phases** A phase is off-line if it does not involve that any particular trusted participant who isn't explicitly part of the phase be online. The more phases are off-line, the easier it is logistically.

**User privacy from the issuer** The issuer does not learn who received the vaccine; from the issuer's point of view there is no difference whether it was Alice or Bob.

**(Trade-off) User privacy from the verifiers** The verifier does not learn anything about a user other than her vaccination status; even if the verifiers all band together they can't track the user.

Trade-offs:

The cost of having the vaccination phase offline is that each Pharmacy must be able to issue a badge by itself, meaning that it will need to have a secret key, use digital signatures, etc; also either verification of a badge will require that the verifier know the pharmacy's public key, or that some more sophisticated cryptographic algorithm (such as delegatable anonymous credentials) be used.

Unless the badge is physically embedded into a user's body, it is hard to achieve non-transferability of badges and user privacy from verifiers at the same time. The best we can do is to allow that a verifier learn some attribute (possibly of the verifier's choice) of a user's identity, and then the rest of the proof that a user with this attribute has been vaccinated can be communicated via a zero-knowledge proof.

## 2 Paper card based solution

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 a commonly used digital signature scheme, one such example is ECDSA [3].

In the paper-based solution we focus on keeping the proposed system as inter-operable as possible with the existing system. The user receives a coupon from the *distributor* (e.g. city) and goes to the *pharmacy* for

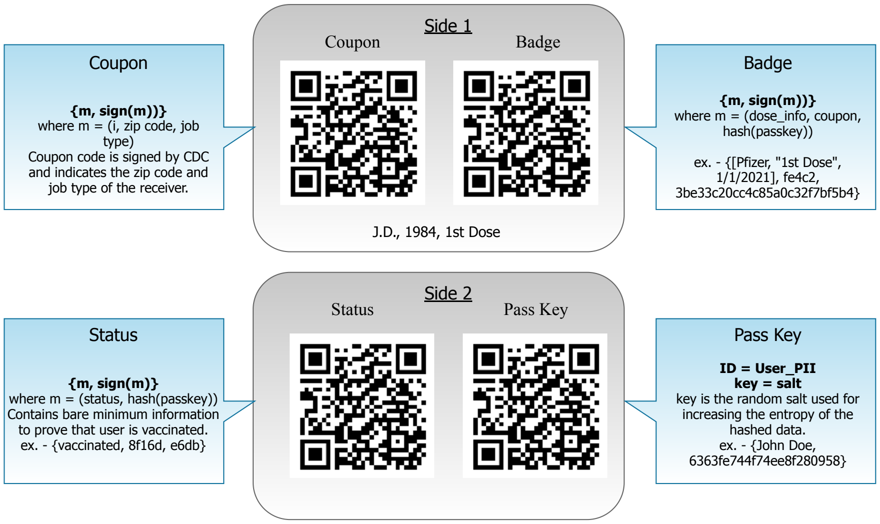
Coupon
{m, sign(m))}
where m = (i, zip code, job type)
Coupon code is signed by CDC and indicates the zip code and job type of the receiver.
Side 1
Coupon
Badge
J.D., 1984, 1st Dose
Badge
{m, sign(m))}
where m = (dose_info, coupon, hash(passkey))
ex. - {[Pfizer, "1st Dose", 1/1/2021], fe4c2, 3be33c20cc4c85a0c32f7bf5b4}
Status
{m, sign(m)}
where m = (status, hash(passkey))
Contains bare minimum information to prove that user is vaccinated.
ex. - {vaccinated, 8f16d, e6db}
Side 2
Status
Pass Key
Pass Key
ID = User_PII
key = salt
key is the random salt used for increasing the entropy of the hashed data.
ex. - {John Doe, 6363fe744f74ee8f280958}

Figure 2: The representation of the paper based solution and how the QR code representation of different stickers could look like. QR code 1 carries coupon information that is used by the user for getting vaccination, QR code 2 carries badge information that consists of information related to vaccination and tied to user identity using passkey. QR code 3 has proof of vaccination and the QR code 4 has passkey of the user that contains PII.

getting vaccinated. Upon vaccination, the pharmacy provides the user a QR code that contains information related to the vaccination status, we refer this piece of information as *badge*. The *badge* can be shown by the user for proving their vaccination status. Now we describe the protocol in more detail -

## 2.1 Protocol

**Coupon distribution** The coupon code is generated by the *issuer* and delivered to the *user* through the *distributor*. The process involves the *issuer* setting up a secret and verifying key for generating the coupons such that the validity of the coupon can be examined by only knowing the verifying key.

- The *Issuer* $I$ (e.g. CDC) first generates its key pair $(sk_I, vk_I)$ for the signature scheme. By $\mathsf{sign}_I(m)$ we will denote the output of the digital signature signing algorithm (such as the ECDSA algorithm [3]) with input as the Issuer's secret signing key $sk$ and the message $m$. By $\mathsf{verify}_I(m, \sigma)$ we will denote the output of the corresponding (e.g., ECDSA) signature verification algorithm under the verification key $vk_I$ on input the message $m$ and a purported signature $\sigma$.

- The *issuer* then generates $n$ coupons for a given zip code and job type of the users by signing the tuple $m_i = (i, \text{zip code}, \text{job type})$ for every i'th coupon. A given coupon $c$ can be represented as

$$c_i = \{m_i, \mathsf{sign}_I(m_i)\}$$

  Finally, the set of coupons $\mathcal{C} = \{c_i | \forall i \leq n\}$ is given to a local *distributor* based on zip code and job type.

- The *distributor* gives the coupon to the users after performing an eligibility check.

**Vaccination** In this phase, the user gets vaccinated and receives three QR-codes in addition to the previously received coupon code. As shown in Figure 2, all four QR-codes are printed on the two sides of the paper with the first side consisting of coupon and badge, and the other side has the status and pass-key. The high level idea is to keep detailed information about the user's vaccination on the first side and keep the minimal information required for proving the vaccination status on the other side.

- The *user* gets his coupon $c$ and shows it to the *pharmacy* to get vaccinated.
- The *pharmacy* runs $\mathsf{verify}_I(c)$ to validate the coupon $c$ and then checks whether the coupon has been used up already using VAMS-like system[1]. Upon successful validation, *pharmacy* vaccinates the user.
- The pharmacy collects the user's PII and verifies that it matches the user's ID (e.g. - by looking at the user's driving license). User_PII refers to the information tied to the identity of the *user* (e.g. - user's full name, date of birth and etc.), which would be used for vaccination status verification in the future.
- The *pharmacy* generates a header $h$ such that $h = \{\text{dose\_info}, c\}$ and generates

  $$\text{badge\_info} = \{\text{dose\_info}, c, \mathsf{hash}(\text{User\_PII, salt})\}$$

  and submits it to the *issuer* for signing it. Salt refers to a high entropy number to ensure sufficient randomness of the hashed value; in particular, this ensures that the hashed value itself does not reveal a user's PII.
- The *badge issuer* (that could be the same issuer as the entity that issues the coupons or a different entity with a different signing key) receives the *pharmacy* request for signing the badge_info. The *issuer* updates its publicly visible registry by marking the coupon $c$ as used and signs the digest of the user information by performing two signatures $\mathsf{sign}_I(\text{badge\_info})$ and $\mathsf{sign}_I(\{\text{vaccinated, }\mathsf{hash}(\text{passkey})\})$ and returns it back to *pharmacy*. Here vaccinated refers to the minimal information required to indicate a user is vaccinated. This could be as minimal as 0 (not vaccinated) and 1 (vaccinated). However, the standards might require indicating relatively more information such as second dose, date of vaccination and etc.
- The *pharmacy* issues two QR-codes for the user -

  $$\textit{badge} = \{\text{badge\_info}, \mathsf{sign}_I(\text{badge\_info})\}$$

  $$\textit{status} = \{\{\text{vaccinated, }\mathsf{hash}(\textit{passkey})\}, \mathsf{sign}_I(\{\text{vaccinated, }\mathsf{hash}(\textit{passkey})\})\}$$

  $$\textit{passkey} = \{\text{User\_PII, salt}\}$$
- The *user* has finally four QR-codes with them *coupon*, *badge*, *status*, and *passkey*.
- **Second dose** The *user* can receive the second dose in the future from any of the *pharmacy* by presenting their *badge* which would be verified by the *pharmacy* first and the vaccination procedure would the proceed as above. The granular information about the vaccination enables the *pharmacy* to administer the right second dose of vaccine for the *user*.

**Verification** The verification process happens in two phases - first, the validity of the badge is examined to make sure the *user* is carrying a valid vaccination proof and then the identity of the *user* is examined to make sure that the QR-codes belong to the person carying it.

- The *user* goes to the *venue* and present their badge.
- The *venue* verifies the integrity and validity of the badge by performing $\mathsf{verify}(\text{badge})$
- The *venue* request the pass-key if they want to verify the identity of the *user*. The identity is verified using a government issued ID. *User* may or may not give the pass-key based on their preference.

[1] `https://www.cdc.gov/vaccines/covid-19/reporting/vams/index.html`

## 2.2 Security

There are security benefits of fragmenting the user information into three distinct QR-codes. First, User_PII does not get centralized at any point under the assumption that *pharmacy* and *venues* do not collude with CDC or with some global data warehouse. The protocol prevents *users* from forging the vaccination proof by using digital signatures. However, a *user* can sell their coupon to other users in the same zip code. While using zip_code and job_type impedes the potential buyers, more personalization could be done for the coupons.

The proposed solution poses some privacy risks such as *venues* keeping track of every user's passkey and creating a PII database of vaccinated users by coordinating with other *venues*. This can be prevented by mandating the *venue* owners to only perform the scanning using a trusted app that deletes the user record after the verification process is over. However, such a regulation enforced privacy would not protect against worst case attackers. Another potential privacy leakage could happen if the *pharmacy* does not delete their record locally after generating the passkey. This would enable the semi-honest *pharmacy* to keep a local database of private information of every *user*. Pharmacies maybe required to keep the PII records in case the user loses the vaccination card and the salt values.

# 3 App based solution

As discussed in the Section 2.2 the paper based solution introduces some of the security and feasibility constraints which limits the effectiveness of the overall system. We propose the following smartphone based protocol to circumvent some of the issues.

## 3.1 Protocol

**Coupon distribution** The coupon distribution happens in a similar way described in the paper-based solution. The coupon code is generated by the *issuer* and delivered to the *user* through the *distributor*. Unlike the paper-based solution, the *user* also generates a public-private key pair and a private data structure for their personal information that would be useful for the upcoming phases.

- The *issuer* generates key pairs for the signature scheme and issues the coupon in precisely the same way as discussed in the paper-based solution.
- The *distributor* provides coupon code

  $$c = \{(i, \text{zip code, job type}), \mathsf{sign}(i, \text{zip code, job type})\}$$

  to the users which either is scanned in person by the *user* or gets downloaded in the *user* app through a confidential URL provided by the *distributor*.
- The *user* generates a public-private key pair $(pk_U, sk_U)$ such that private key $sk_U$ is stored with the user in a secure enclave [1] or sufficiently secure system where even the *user* can not see it themselves but only decrypt information by using decryption API with the operating system (this property would be useful later on to prevent forgeability of the vaccination status). The *user* also generates

  $$\text{pii\_hashes} = \mathsf{hash\_tree}(\{\text{User\_PII}_i, \text{salt}_i | \forall i \in \{\text{User\_PII}\}\})\}$$

  Here hash_tree takes a set of *user's* PII (e.g. - name, age, residence and etc.) and processes individual elements of the set to construct a merkle tree [3].
- **Second dose** The *user* can receive the second dose reminder through the app in the future and they can go the *pharmacy* and get their *badge* scanned. The *pharmacy* validates the *badge* and the vaccination procedure would the proceed as mentioned above for the first dose.

**Vaccination** The high level idea here is that the *user* presents the coupon code at the *pharmacy* then *user* gets vaccinated and then *user* presents their PII to the *pharmacy* so that it can be signed securely by the *issuer* for later use.

- The *user* shows their coupon code in the app for scanning at the *pharmacy.*
- The *pharmacy* validates the scanned coupon code $c$ and proceeds to vaccination upon successful validation of the coupon.
- After vaccinating the user, the *pharmacy* generates

  $$\text{badge\_info} = \{\text{dose\_info}, c, \mathsf{pii\_hashes}\}$$

  $$\text{v\_status} = \{\text{vaccinated}, pk_U, \mathsf{pii\_hashes}\}$$

  Using the pii_hashes, *user* can perform selective disclosure of the information and still be able to prove the integrity of their badge.
- The *pharmacy* sends $h$ to be signed by the *issuer* and then returns the *badge* and *status* to the user by providing {badge_info,$\mathsf{sign}$(badge_info)} and {v_status, $\mathsf{sign}$(v_status)} respectively.

**Verification** In the verification phase, the goal of the user is to prove that they have been vaccinated by providing a digitally signed information that consists of sufficient and minimal information about the *user's* vaccination status and identity. In comparison to the paper based protocol, the *user* can now selectively disclose only a subset of information by using the pii_hashes.

- The *user* walks in to the public venue and presents their badge.
- The *venue* scans the badge and validates the signature by performing $\mathsf{verify}_I(status)$ and requests particular personal information.
- The *user* gives information by performing consent based *selective disclosure* of their private information.
- The *venue* confirms the identity of the user based on the information received and the government issued ID.

**Contact-less Group Verification** The high-level idea of the contact-less group verification is that the user carries a smartphone with them and the *verifier* shares a secret number on the wireless channel such that only the users with correct vaccination status can present this secret number on their screen and walk-in. Figure 3 depicts the group verification protocol where individuals display the generated code to the guard.

The contact-less group verification uses a secure channel for communication between every *user* and the *verifier* to prevent against any spoofing based threat that is possible in a wireless communication. The secure channel can be established using the TLS protocol, for more details about the design and implementation of the protocol, we refer the reader to the RFC [12]. To establish the TLS connection, the *verifier* can share the certificate in three possible ways -

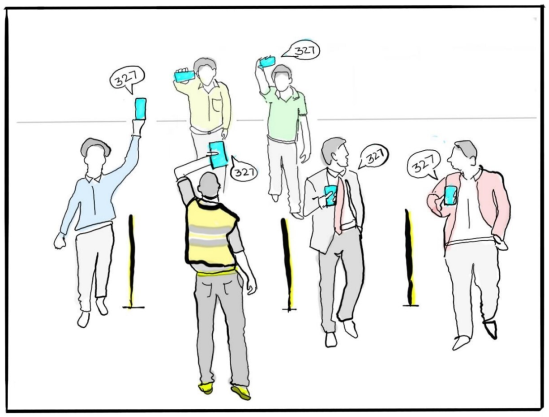

Figure 3: Depiction of the contactless group protocol where individuals perform information exchange over a wireless channel securely and generate a code on their smartphone app that proves their vaccination status to get access.

- Certificate signed by the *issuer* - The *issuer* can provide digital certificate for the public key of the venue and let *user* exchange keys with them.
- Self signing certificate by scanning venue QR code - The *user* can scan QR code available near the venue to obtain its digital certificate and trust it for further establishing the secure channel.
- Self signing certificate using key exchange - The *venue* can display a code that can be entered by the *user* in their app to trust a given certificate being broadcasted wirelessly. This code is tied to the public key used in the a digital certificate produced by the *verifier*.

In the following we describe an interactive protocol that occurs 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* first establishes a TLS connection with the *verifier*
- The *user* then sends $\mathsf{status}$ to the *verifier* on the secure channel.
- The *verifier* validates the message by performing $\mathsf{verify}_I(\text{status})$ to the user.
- The *verifier* responds with the challenge $k$ by sending $c = \mathsf{Encrypt}(k, pk_U)$
- The *user* decrypts the packet by $k = \mathsf{Decrypt}(c, \mathsf{sk}_i)$.
- The *user* then shows the value $k$ on the phone to the verifier.

### 3.2 Security

This protocol has all of the properties discussed in the paper-based protocol in the section 2.2 and further improves the security by introducing the idea of selective disclosure of PII. This enables a fine grained control on the personal information for an *user*. However, the previously discussed attack where the *venues* could create a database of User_PII is still possible. To circumvent such an attack, the app could provide anonymous credentials, which is the current direction of our ongoing work.

### 3.3 Non-traceable verification

The proposed protocol has the limitation of being traceable across venues. We fix the issue by integrating anonymous credentials into the verification protocol [4]. The anonymous credentials module can be implemented efficiently using the recent efficient versions of anonymous credentials [2]. The user would use different attributes for obtaining their credentials. During the verification phase, the verifier would request a random subset of attributes from the user for which the user has to present a physical proof. Post submission, the user would *proove* the signed message by performing the verification phase of anonymous credentials [2].

### 3.4 Assessing Health Outcomes of Safety and Efficacy

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

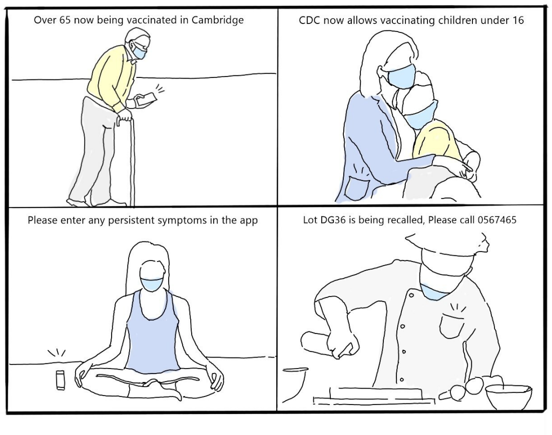
Over 65 now being vaccinated in Cambridge
CDC now allows vaccinating children under 16
Please enter any persistent symptoms in the app
Lot DG36 is being recalled, Please call 0567465

Figure 4: Health reporting and vaccination monitoring

**Upload:** Users can upload symptoms data at any point using their coupon code ($c$). The *issuer* can validate the upload using the coupon number and associate it with a verified vaccination by the *pharmacy*. *User* can also ask their doctor to submit an adverse reaction report using the same $c$. The upload of symptoms can be made privacy preserving by aggregating data using multi-party computation based aggregation methods [7, 6] on top of which differentially private mechanisms [9, 8] can be used to ensure privacy of 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 the user to reveal everything about themselves. Similar to GAEN [10] key server, the app can download the *adverse events data report* that is public for their state, every morning. The app 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 an adverse reaction to certain food allergy or immune health conditions)

**Aggregate view:** How can a vaccine maker, a US state or the CDC have a detailed or aggregate view about the population level statistics? (i) With anonymity, a user can be tracked using the provided coupon all through their journey. If we do not want the user coupon to be tracked across the journey, the user can upload the symptoms without the coupon code. (ii) If the user does not wish to upload symptoms in the raw, we propose to use a protocol based on secure multi party computation similar to Prio [6] to provide an aggregate statistic without revealing the privacy of any individual. (iii) For users without an app, they can log into V-SAFE [5] or VAERS [13] system using their coupon code. Similarly, a doctor updating adverse reaction report can use the coupon code.

## 4 Limitations and Attacks

Digital tools for pandemic response can have privacy and ethics issues at various fronts [11], therefore we discuss some of the issues and potential pitfalls for the proposed technology. The proposed protocol provides anonymity but not privacy if the *user* has to interact with the system. The ability to track the coupon $c_i$ 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 *users* to furnish a state ID or similar equivalent. Therefore, the name or some identifying information needs to be embedded as part of the record code. While we store this personal information in a hashed signature, the system does not protect against the notion of plausible deniability. This can be solved by having the *user* information stored exclusively with them while letting the *user's* app to prove the correctness of the information using zero knowledge proof. This is the direction of our current work.

In our proposed protocol, the *pharmacy* knows the information about the *user*, and hence in the worst case, can collude with the *issuer* to reveal personal information and identify a given *user*.

## 5 Acknowledgements

We would like to thank Priyanshi Katiyar for helping us out with the sketches.

# 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] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1087–1098, 2013.

[3] Dan Boneh and Victor Shoup. A graduate course in applied cryptography. *Draft 0.2*, 2015.

[4] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *International conference on the theory and applications of cryptographic techniques*, pages 93–118. Springer, 2001.

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

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018.

[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.