Weekly Blog 2

Sree Ram Boyapati	a1775690

What have I done this week? Tangible Outcomes

As part of the project, I have divided my tasks as the following - How to make a APP-Compliant server infrastructure taking costs and scalability into account.

The Australian Government has provided many guidelines on how to adhere to the privacy laws and mitigate privacy concerns through Australian Privacy Principles. https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-quick-reference/1

As part of the server infrastructure, Following principles have to be obeyed

1. Anonymity (Data Collection)
2. Dealing with unsolicited personal information.
3. Cross-border disclosure of personal information.
4. Security of personal information. (Identity and Authorization)

To address (3) and (4), We have decided to host the entire storage and compute infrastructure on Google Sydney Data Centres.

To specifically address concerns (4), CMEK (Customer Managed Encryption Keys) i.e (keys generated by me rather than google) will be used to encrypt data at rest using Google Cloud KMS.

To address (1) and (2), Data in the last 14-21 days is stored in the cloud and only pseudo-anonymized data which has been collected from the GAEN protocol based android client is being used. Cloud Functions support scheduled actions to do the job.

Diagnosis verification servers are hosted using Google HealthCare API which are in a different network and communicate with Exposure Notification Service for diagnosis verification. Diagnosis Verification Servers need to be HIPAA compliant. Google Healthcare API manages de-anonymisation of records for analysis and full-manages access to data which might include clinical reports of patients after testing. These servers are beyond the scope and hence have been replaced with mock keys of users.

Privacy Notice and Architecture of Covid-Warn Android App has helped us in scoping our infrastructure.
https://www.coronawarn.app/assets/documents/cwa-privacy-notice-en.pdf

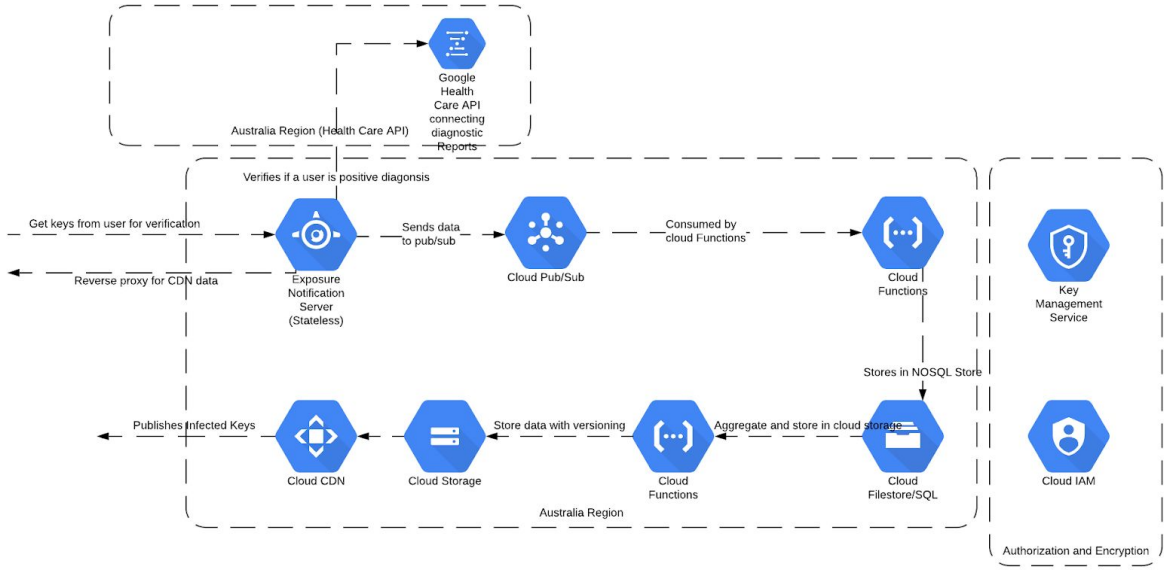

Figure 1. Reference Infrastructure of CovidGuard-F

Name	**Data Type**
identifiers	Array storing the last 14 days RPIs generated
expired_at	Epoch Timestamp
created_at	Epoch Timestamp

Table 1 - Data model of the document to be stored in FireStore

Data in Table 1. will be streamed through the pub/sub as json data. Data in the firestore can be encrypted using client side encryption however, I am treating it optionally as firestore already supports server-side encryption.

Interaction with Diagnosis Verification servers is presently mocked and some tokens are hard coded as valid. However, We have added interaction with the diagnosis verification modelled in our architecture. API calls to Diagnosis servers will be signed using secret keys known only to health authorities and data controllers of ENS servers stored in Cloud KMS.

Properties of App Engine

1. TLS 1.3 enabled on google app engine
2. Version based deployment to App Engine

Public RPIs of infected users

1. Data from the firestore is periodically aggregated and stored in the cloud storage which is exposed through a cloud CDN for the applications to download a list of infected users.
2. Aggregation is performed by cloud functions and firestore scheduled actions. https://firebase.google.com/docs/functions/schedule-functions

We have been granted access to Google Cloud Free Tier and designed our architecture using Always Free components. My total estimate is 2 dollars per AUD and we got a 300 dollar credit. We took a baseline of 1 GB data in storage.

Google Cloud also provided us a lot of information on how to adhere to APP principles.
https://cloud.google.com/security/compliance/australian-privacy-principles

Audit Logs are collected for compliance reasons. Google Firebase was not used because of trackers. App Engine provides versioned releases with google provided third party libraries for server implementation which is very minimal thanks to GAEN decentralized approach.

Bandwidth (Egress) and Price Estimation

For 500k infected users, 8MB is the total message size served through cloud CDN.
If it is periodically downloaded and aggregated every 6 hours, It is very low for testing purposes.

However, for users in Australia with 10k Active cases, Payload size is 160KB (each identifier is 16 bytes). This is very permissible given that many users might be indoors and have got around 20 GB of bandwidth on average per month. This allowed us to take the GAEN approach for contact tracing and model our architecture accordingly.

Complete Cost to Host the infrastructure - **1.22 AUD per month**
https://cloud.google.com/products/calculator/#id=ad29963c-f017-4cee-9b41-b7285701014f

In the next few weeks, I plan to set this infrastructure up using Terraform and work with my teammates on server implementation.

Paper Review # 1

Liu, Joseph K. et al. “Privacy-Preserving COVID-19 Contact Tracing App: A Zero-Knowledge Proof Approach.” IACR Cryptol. ePrint Arch. 2020 (2020): 528.

Challenges and Why did I choose this paper?

In the GAEN approach, There is a lot of egress costs as the entire dataset of the infected users is shared with all the users to check if they have come in close contact. Risk Analysis and exposure score is derived in the device.

This approach has three issues -

1. High processing power required on the phone. (Drains battery faster)
2. High bandwidth usage as some countries may have expensive data plans and it might cost a lot to use the app in countries with a high number of active cases.

Hybrid Architecture

1. Instead of downloading the entire packet, Can the server collect all the contacts from infected users and store them so that users can ask the server if they have been infected?
2. Loss of privacy is minimal if the server can know someone is infected but cannot decipher the identity. Also, If the contact is in an indoor setting - Chances of contracting covid-19 is higher.
3. Egress costs are much lower and much better scope of doing analysis at the server side.

Strength of privacy - Number of false-positives that the server knows in a hybrid architecture. However, not all servers can be expected to be *Honest-but-curious*.

Hence I chose this paper to study -

1. Can ZKP be used to validate if there has been a contact by sharing with the government/medical staff without revealing their identity?

Conclusion of the Research:

1. We still need to send a massive payload of possible exposure contacts, however, in this approach, Government notification service can be localised based on city or state.
2. False-Positive cases are severely reduced as a Medical doctor signs the user as a positive case on behalf of a group and notifies government bulletin of possible contacts and users can verify their proof of contact they have shared earlier and that notification has indeed come from the medical professionals and government.

3. Social graph attacks can be prevented as well as linking attacks as proof of contact lies with medical professionals using pseudo public keys.

Outcome of the Research:

1. City level or State level bulletins are possible to reduce the payload size. This can be incorporated by user settings and having separate buckets of infected users aggregated. This should be further explored.
2. Medical professionals/Government not knowing the possible contact information of the infected patients may not be desirable as people try to escape lockdowns and travel from Victoria to Adelaide (as we have read in the news). GAEN also suffers from this pitfall. Medical Professionals have to know who the possible contacts are.
3. ZKP addresses the privacy of confirmed patients on a larger scale because by checking contact with infected users, users can decipher the confirmed patient if the number of contacts are very low. ZKP broadcasts your pseudo-public key. However, in the case of ZKP, egress costs are even higher.

In the ZKP approach, false-positives and linkage attacks surface far lower, and the social graph is difficult to construct as we are broadcasting possible contacts. However, it is not helpful in addressing egress costs.

Paper Review # 2

Vaudenay, Serge. “Centralized or Decentralized? The Contact Tracing Dilemma.” IACR Cryptol. ePrint Arch. 2020 (2020): 531. I am taking this as my third paper

Challenges and Why did I choose this paper?

In the previous papers, We have discussed the high egress costs of a decentralized solution in both ZKP-based approach and GAEN based approach. I would like to delve deeper on security and privacy aspects of a decentralized solution compared to a centralized solution which might be proactive to isolating positive cases. Centralized or Decentralized, Guidelines and infrastructure level support needed for data controllers to meet compliance has been sketched out.

One of the concerns of decentralized solutions is -

1. Efficiency - Time to identify a new positive case entirely depends on if the contact user still has the app. No information regarding contacts is being collected for health authorities to contain them *as soon as possible*.
2. Cluster Identification - Venues must be alerted and without location data, knowing if a venue is affected is very hard as cluster analysis is not possible with just RPI. VenueTrace partially solves the problem. However, covid-19 doubles in a week and at peak in a couple of days. VenueTrace adoption may not match the growth of COVID-19

Centralized Solutions suffer from -

1. Security of the data in the hands of few stakeholders.
2. Risk of surveillance and loss of privacy.

Given that preventing deaths is of utmost priority, We would like to consider centralized and decentralized solutions with priority for efficiency.

Conclusion of the Research:

1. Tracking People - Risk of tracking people is much higher in centralized systems like ROBERT where pseudonyms are used to generate the centralized details. Decentralized systems like GAEN and DP3T generate keys every day. This intervals can be shortened to minimize the risk of a exposed key
2. The paper says the centralized systems are better at managing the risk of identification of diagnosed users. I disagree with this because ZKP and Pronto-C2 manage

decentralized bulletins and ZKP publishes pseudo public keys. Most of the attacks like malicious apps etc can be mitigated and are usually very high-effort.

3. Delayed Authentication (2FA process) to avoid replay attacks and generate false positives. In ZKP based approach, proof of contact is computed to avoid false-attacks which are computationally expensive. The author's concerns regarding decentralized systems in this aspect can be mitigated.

Decentralized and Centralized systems differ in two ways based on the server:

1. Decentralized systems have minimal servers and share large payloads to process it on device.
2. Centralized systems have heavy server functionality centralized.

If there were no privacy concerns like Social Graph reconstruction, Centralized systems fare better. However, That is a major concern as it can be used for surveillance in authoritative countries.

I really like the approach of Section 5.2 in the paper which says *Decentralized Architecture with Restricted-Access Server.* Status verification is done by client uploading its identifiers to the server and letting it know if the server. In this approach, The server only stores a list of infected users and the app periodically sends its contacts to the server.

This requires rate-limiting and the server needs to know if the account is indeed unique. The paper proposes the uniqueness of the account to be determined by health authorities. This may not be an ideal scenario. However, complexity of set intersection with the data stored in the server and data from the app is minimised using bloom filters or Flajolet-Martin sketch. This seems like a very plausible model to reduce egress costs and computational complexity of large scale data analysis.

Further research is needed on how to establish a unique app per user with the server and reducing the complexity of set intersection of users. During enforced lockdowns, Universal sets can be limited to an area.

Outcome of the Research:
It helped us validate our approach and we plan to check if we can use the restricted access version to reduce the egress costs of our server setup for the demo. I would like to discuss this with the supervisor and let him know of my plans.

Paper Review # 3

Bradford, Laura Rachel et al. “COVID-19 contact tracing apps: a stress test for privacy, the GDPR, and data protection regimes.” Journal of Law and the Biosciences 7 (2020): n. pag.

Challenges and Why did I choose?
After being convinced, GAEN decentralized way is the right way to go about contact tracing COVID-19 patients. Making a server compliant with the privacy laws especially GDPR and Australian Privacy Principles is paramount for privacy Impact Assessment and Security of the application.

We have two servers -

1. Exposure Notification Server - which talks to clients
2. Diagnosis Verification Server - Equivalent to Medical Doctor User secret key in our ZKP approach which says if the patient is a positive case or not.

Diagnosis Verification Server may have access to clinical reports and Exposure Notification Server needs to be tamper-proof with mitigations and playbooks built in case of privacy breaches.

The paper by Bradford et al. 2020 provides us information on how we can process data to make our servers compliant with privacy laws and healthcare laws of the state

Conclusion of the Research:

1. Personal data was well defined - Any identifier through which we can associate a person’s identity is defined as personal data. Rolling Proximate Identifiers are highly anonymous and the huge set of last 14 days is being collected for verification.
2. There is a valid concern raised that people who operate the ENS may not be honest-but-curious. To ensure compliance, supervisory authorities must be given access on how the servers performed.
3. Health care data concerns in HIPAA and CCPA summarize that through an identifier can an individual know about their health status. Positive diagnosis data of the individual is shared with consent from the app and only RPI data is shared with health care authority and ENS server validates the RPIs stored in the diagnosis verification server.
4. Is it legal for the state to ask for consent of the user on positive diagnosis as powers of the state to persuade the individual to share data are higher? GDPR and other laws provide information that in public interest as COVID-19 is a pandemic, it is acceptable.

Outcome of the research in our project:

1. The paper helped us in choosing the cloud provider who automated data access management and provided access to use our own keys to encrypt data at rest to prevent organizations from hacking the data. This is helpful in states where companies are state-owned.
2. In vetting privacy of covid-19 tracing apps, Concerns regarding firebase were shared. Hence we chose app engine since it provided serverless applications where version based deployment of server is possible. We have used cloud functions internally for the ephemeral nature of compute resources.
3. As per GDPR, Server Architecture developed by us categorizes us as a data controller. We need specific authorization from public health authorities to operate. In case of a diagnosis verification server, Whitelisted IPs and Public and secret keys might be shared to sign our payloads to request validation of data. These keys can be stored in Cloud KMS solution only accessible by app engine on request (which is logged). A whitelisted domain has to be bought for compliance or a static IP which has been noted in our cloud costs.
4. As a data controller, Transparency of processing is paramount. Hence, We shall create all our cloud resources using Terraform templates (Infrastructure as Code) which will be open source.