

# Cryptic COVID Contact Apps

...

What is the state of the art for  
Coronavirus contact tracing apps?

# What this talk is:

An appraisal of the current proposed methods for contact tracing apps, focusing on those utilising Bluetooth or BLE.

We focus on three main implementations:

1. NHS Colocate
2. BlueTrace
3. Apple/Google Contact Tracing



## What this talk is not:

Privacy

# Secret NHS files reveal plans for coronavirus contact tracing app

Documents left unsecured on Google Drive reveal the NHS could in the future ask people to post their health status to its Covid-19 contact tracing app



What this talk is not:



# Central Focus of this talk - Cryptography, Privacy, and Design

We focus on the implementation of cryptography in these proposed/supplied solutions with the following focal points:

## **Cryptography and protocol construction**

How are these  
protocols put  
together, and what  
primitives do they  
use?

## **Privacy Preservation and Considerations**

How well are privacy  
concerns handled? Are  
there areas in need  
improvement?

## **Implementation issues and fidelity**

How faithful are the  
apps to their original  
designs, and what  
implementation issues  
are there?

# But Why are we developing these?

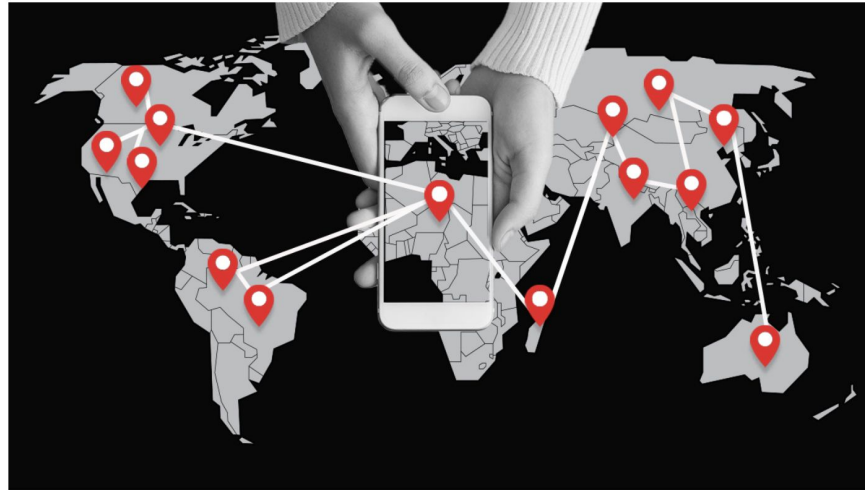
HEALTH

## How Digital Contact Tracing Slowed Covid-19 in East Asia

by [Yasheng Huang](#) , [Meicen Sun](#) and [Yuze Sui](#)

April 15, 2020

 Summary  Save  Share  Comment  Print **\$8.95** Buy Copies



HBR Staff/Getty Images

<https://hbr.org/2020/04/how-digital-contact-tracing-slowed-covid-19-in-east-asia>

# Technology Options

## Passive Options

### GPS

Accuracy to 6ft diameter around receiver is not great for contact/proximity tracing

### SS7

The telco backbone network has capability for working out location, but to worse granularity

## Active Options

### WiFi

Consumes lots of power in AP mode, and probably too loud to be useful

### BLE

Low power and reaosnable to use, but has active privacy preserving properties (rolling MACs) to make it hard to track

# Side note: Bluetooth/BLE is not designed for this...

Phones

## If Bluetooth doesn't work for contact-tracing apps, what will?

The inherent limitations of Bluetooth will make Apple and Google's contact-tracing apps woefully imprecise, but right now it might be the best solution we've got



By **ALEX LEE**

Friday 17 April 2020

<https://www.wired.co.uk/article/bluetooth-contact-tracing-apps>



# There may even be some very core issues...

Photo illustration: Soohye Cho/The Intercept, Getty Images

## THE INVENTORS OF BLUETOOTH SAY THERE COULD BE PROBLEMS USING THEIR TECH FOR CORONAVIRUS CONTACT TRACING



Sam Biddle

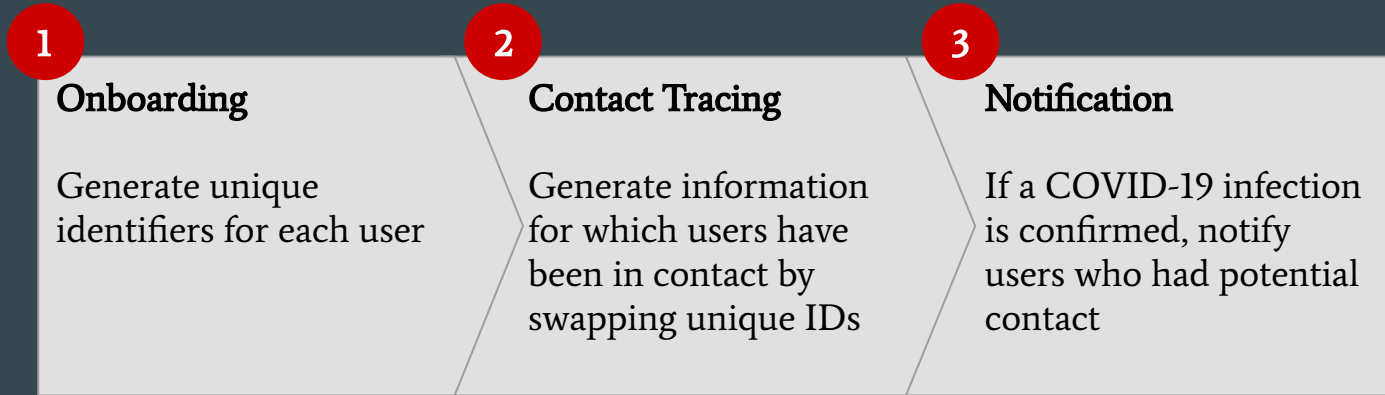
May 5 2020, 11:00 a.m.

**BLUETOOTH HAS SPENT** much of its life ignobly associated with crummy headphones, byzantine connection procedures, and car stereo systems that never quite seem to work right. Now this wireless technology concocted in the '90s to help PCs and mobile phones communicate is being asked to step up and save the planet from a global pandemic. According to its two co-inventors, there could be some issues.

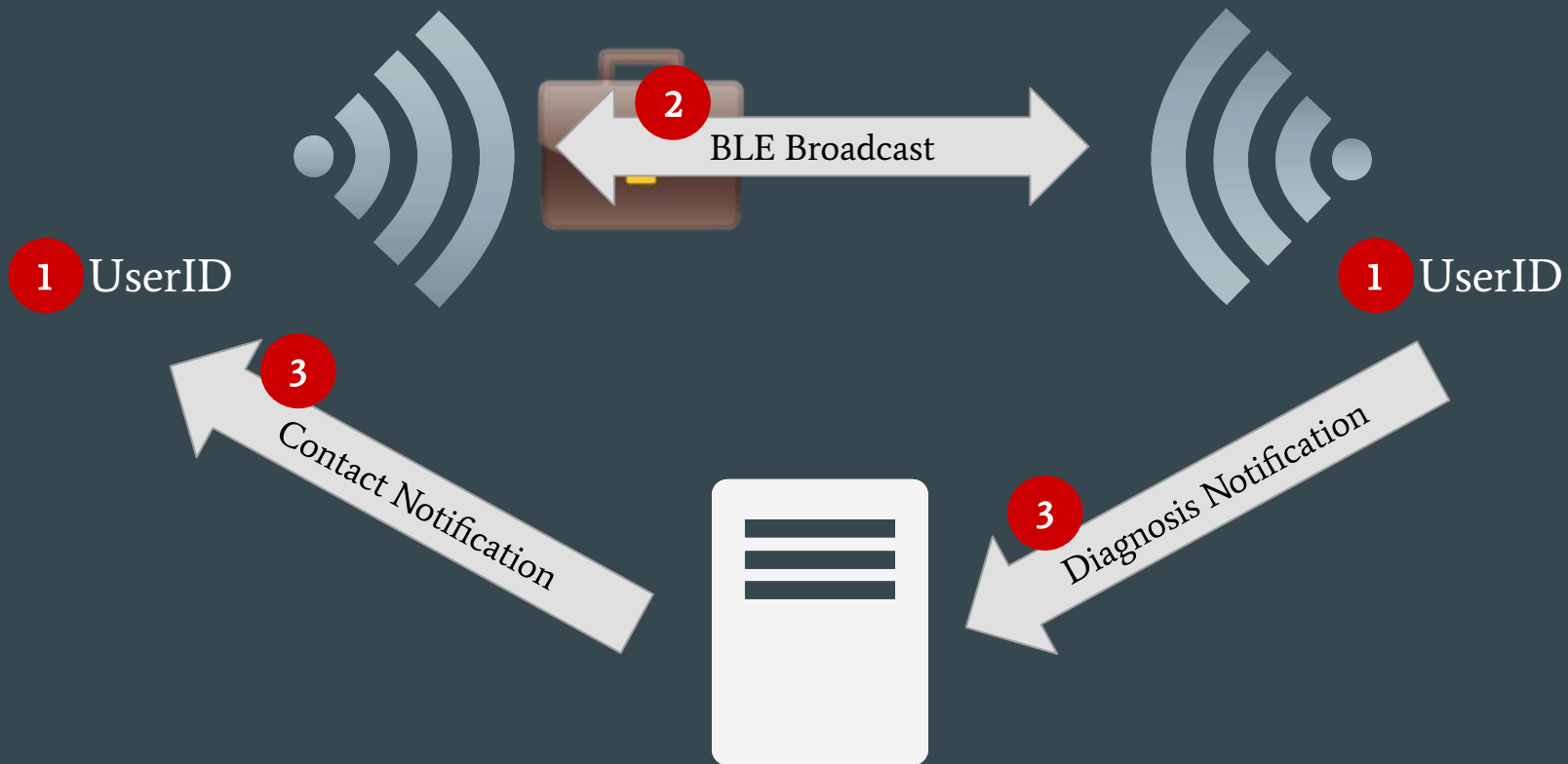
<https://theintercept.com/2020/05/05/coronavirus-bluetooth-contact-tracing/>

**NB - SARS-CoV-2 can be spread by people who are asymptomatic.**

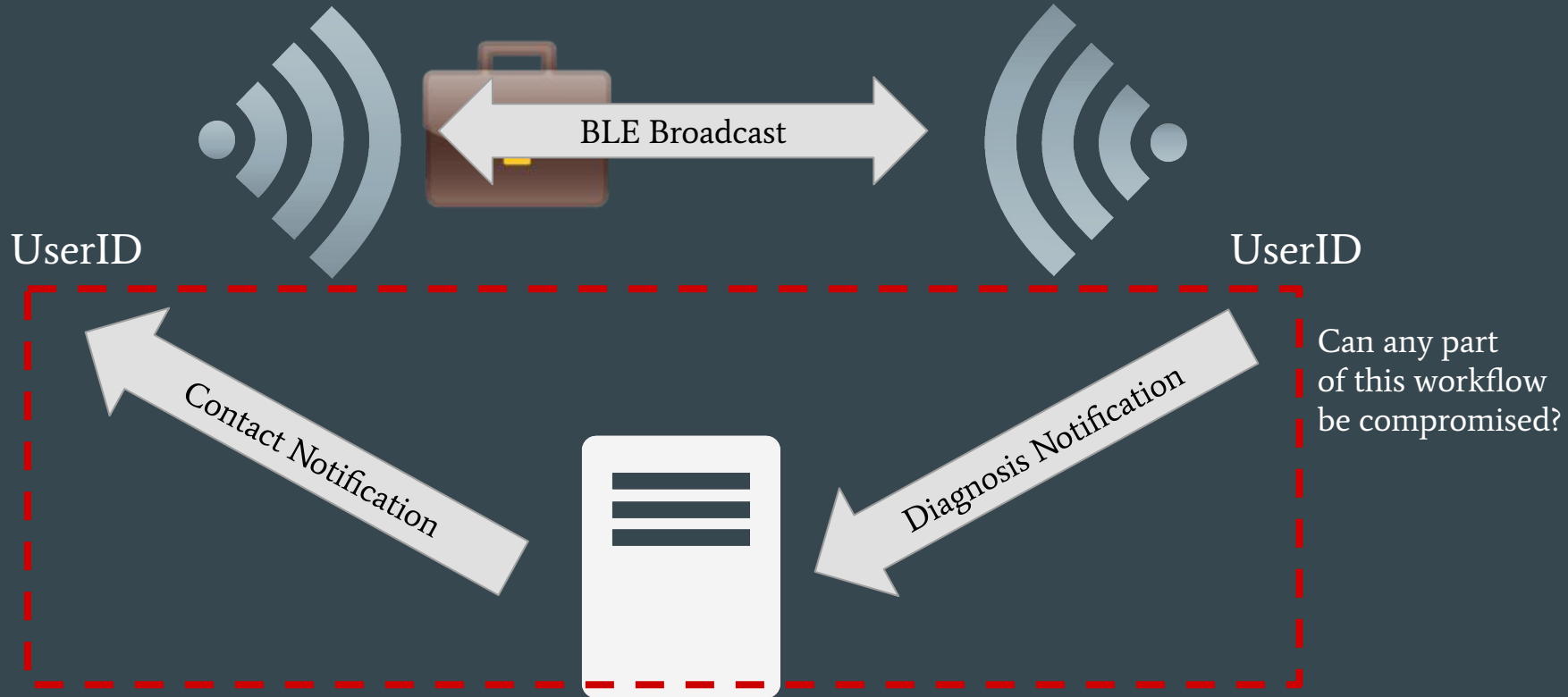
# The overall workflow for a contact tracing app:



# General idea of BLE-based contact tracing apps

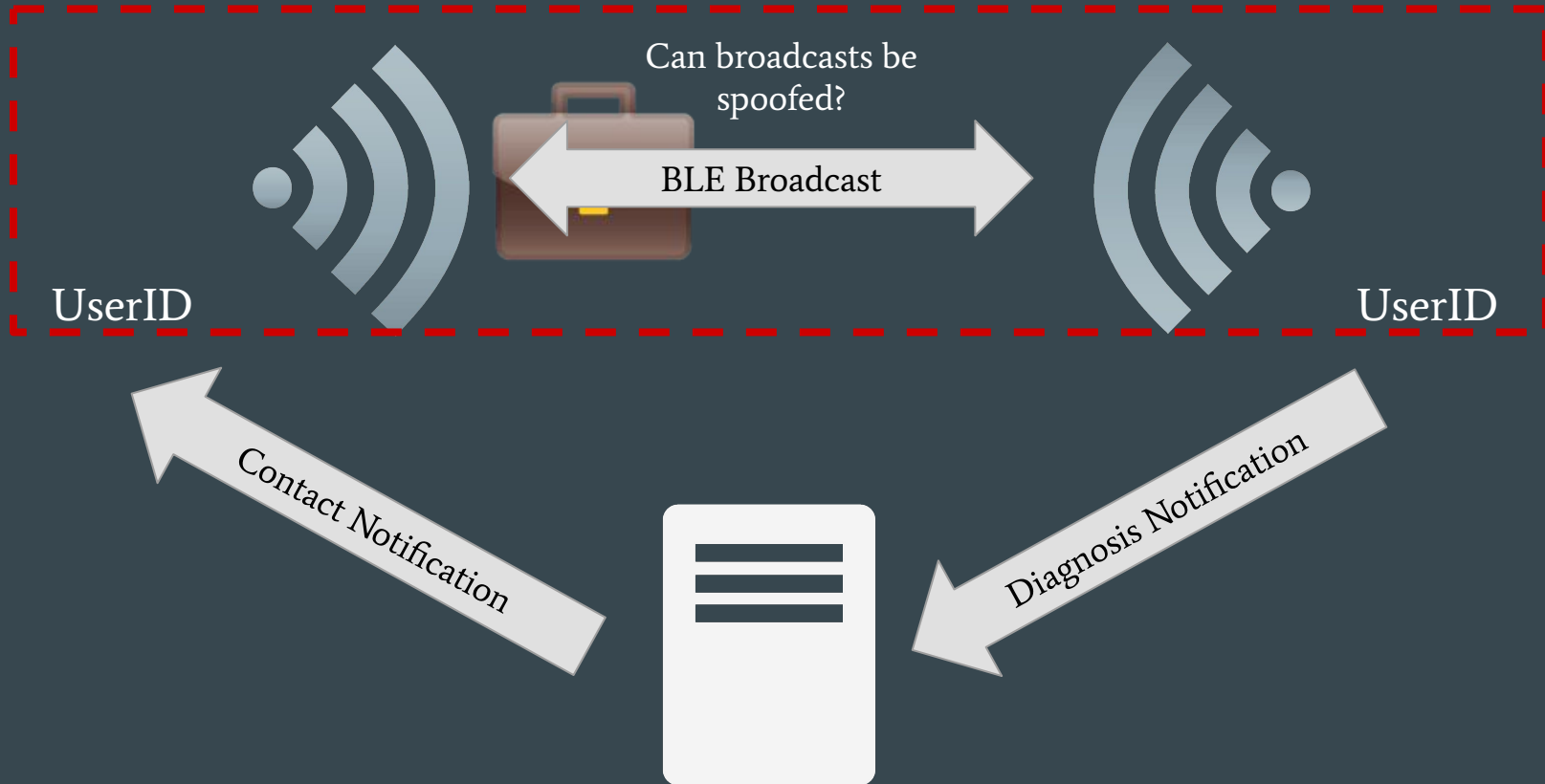


# Privacy and Security Considerations

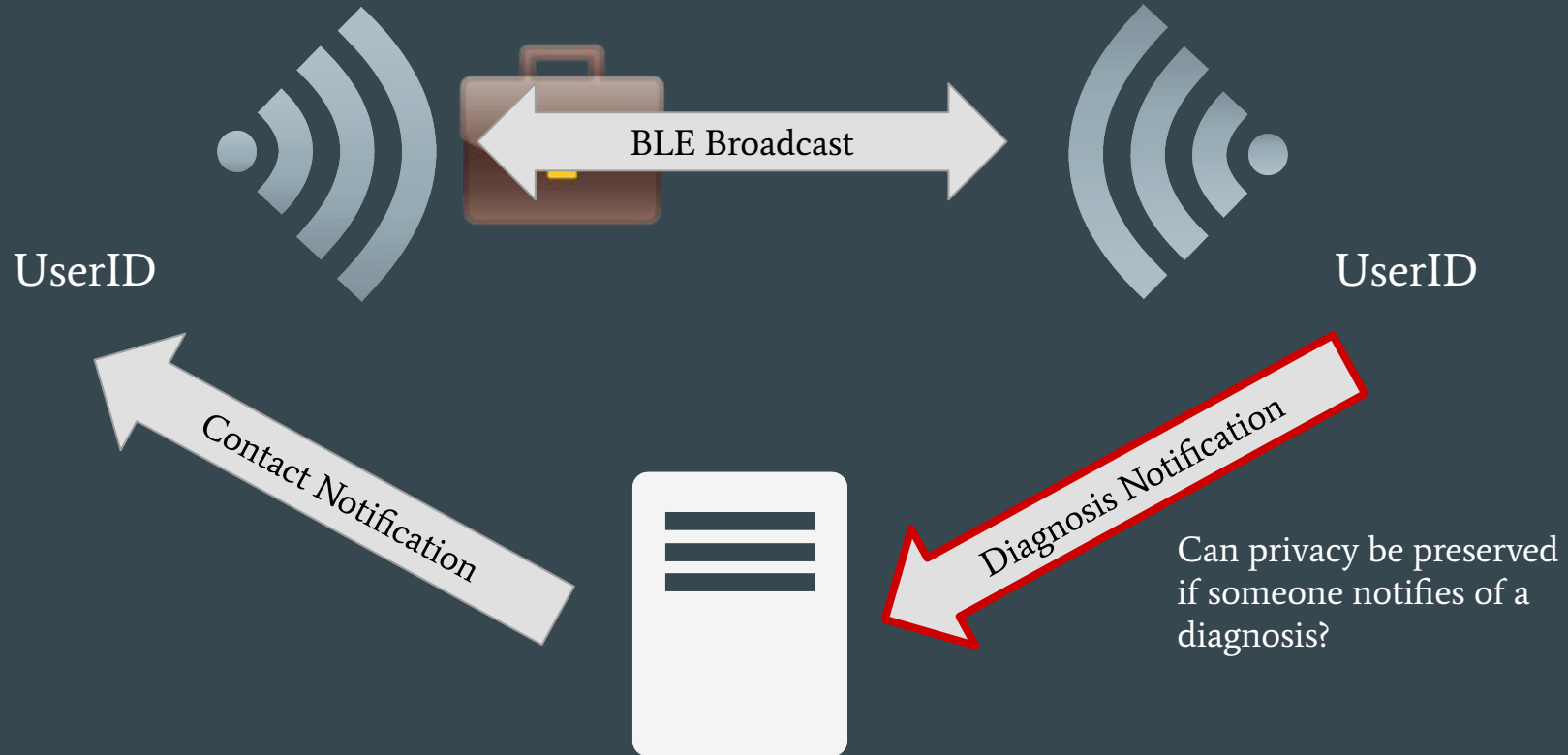


# Privacy and Security Considerations

Can a malicious actor determine who is who?



# Privacy and Security Considerations





NHS Contact Tracing App ([uk.nhs.nhsx.colocate](https://uk.nhs.uk/nhsx/colocate) version)



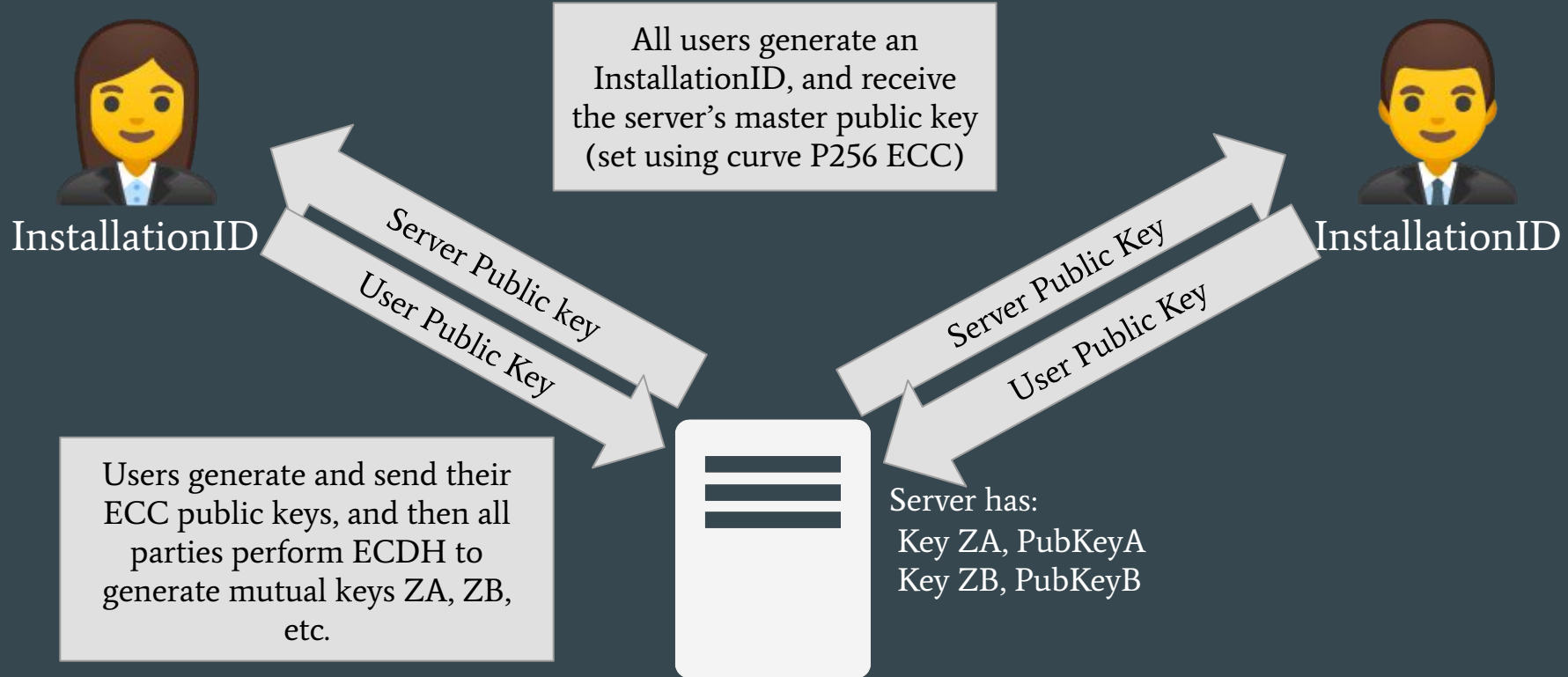
# Back story regarding the NHS app

App draft design document released late April 2020

First candidate apps for Android and iOS released on 6th May 2020 for Isle of Wight users


8th May - work on second app announced, using Google-Apple model, consulted on by a Swiss cryptography firm.

# All parties get the server's master public key, generate their own keys, and perform ECDH



# Alice derives her Broadcast Value (BV):

ZA  
InstallIDA

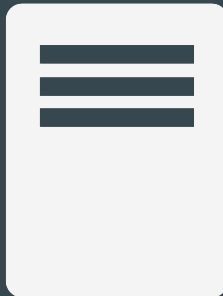


```
Key, IV = KDF(ZA, PubKeyA)
M = (Start||End||InstallID||Country)
Cipher,ICV=AESGCM(M, IV)
BV = (Country||PubKeyA||C||ICV)
```

ZB  
InstallIDB

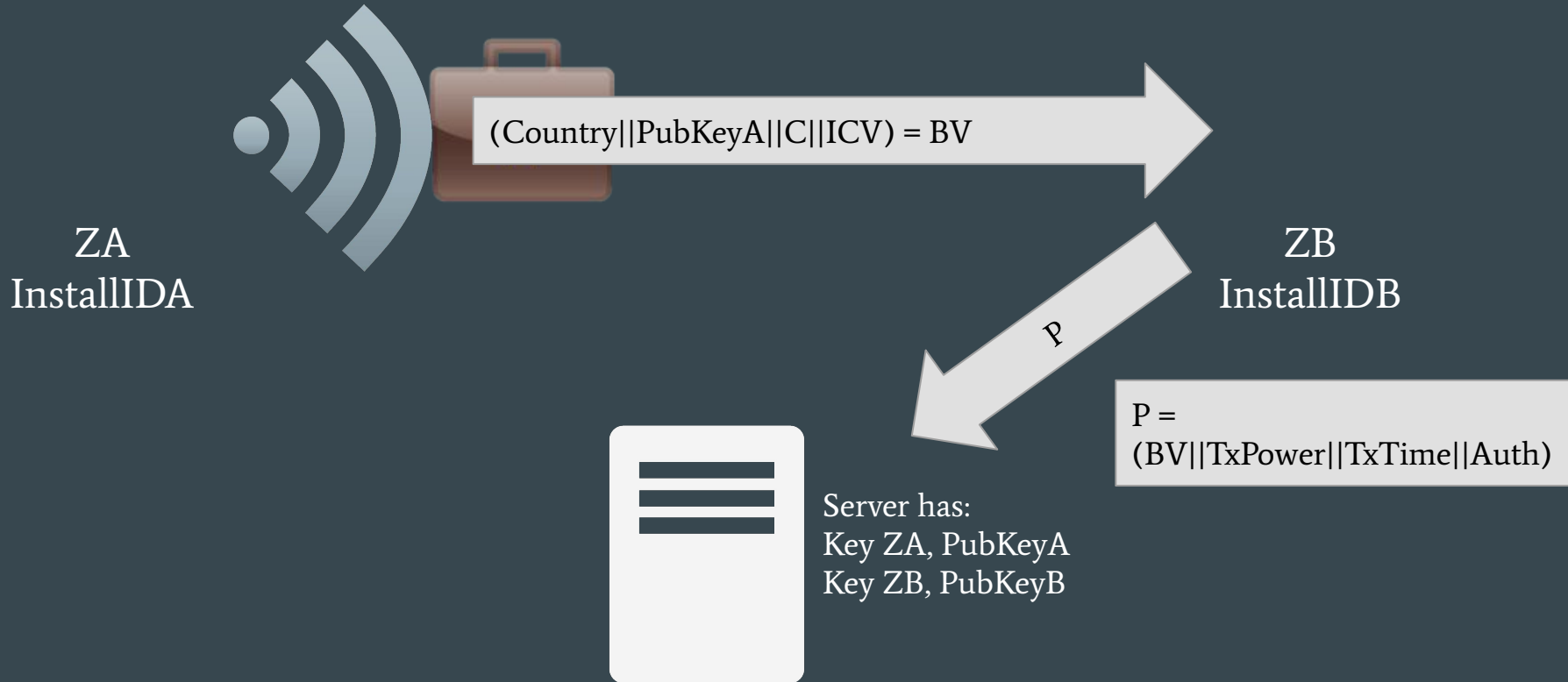
## Non-maths explanation -

Each user generates a value BV that is their public identifier for that day.

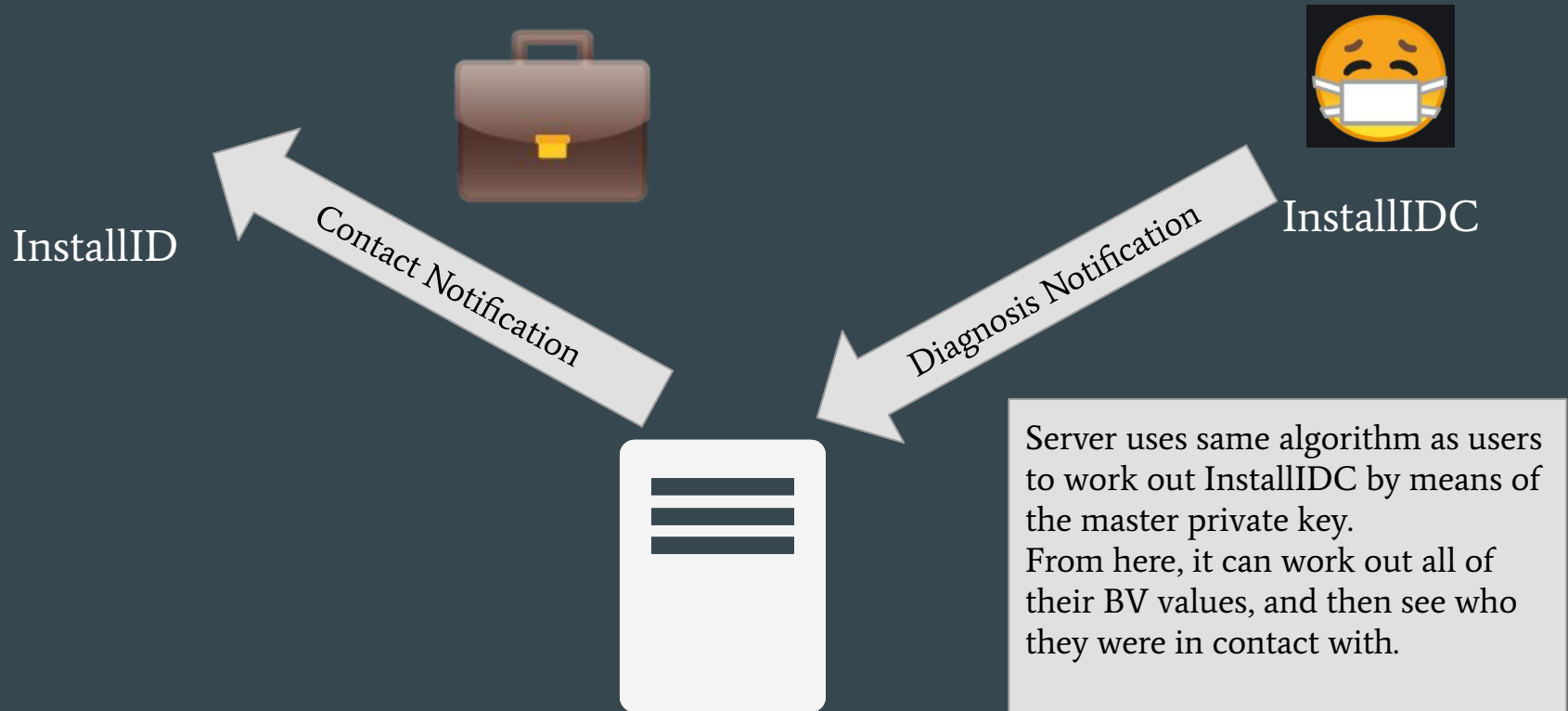


Server has:  
Key ZA, PubKeyA  
Key ZB, PubKeyB

# User derives the daily Broadcast Value and sends it out:



# Diagnosis Notification Workflow



# Benefits and Drawbacks of the NHSX approach

Pros	Cons
Central management takes care of individual notifications - data isolation means no potential for delayed compromise..	Very heavy processing required by the central services to derive and notify those who need to be after a diagnosis is declared.
Strong Cryptographic primitives are used (AESGCM, P256 ECC, etc.)	Single Master key for the whole system is a potential single point of failure.
	Much of the cryptography seems unnecessary.
	Broadcast Values (BVs) are static for a whole day - this could permit temporary tracking by third parties.

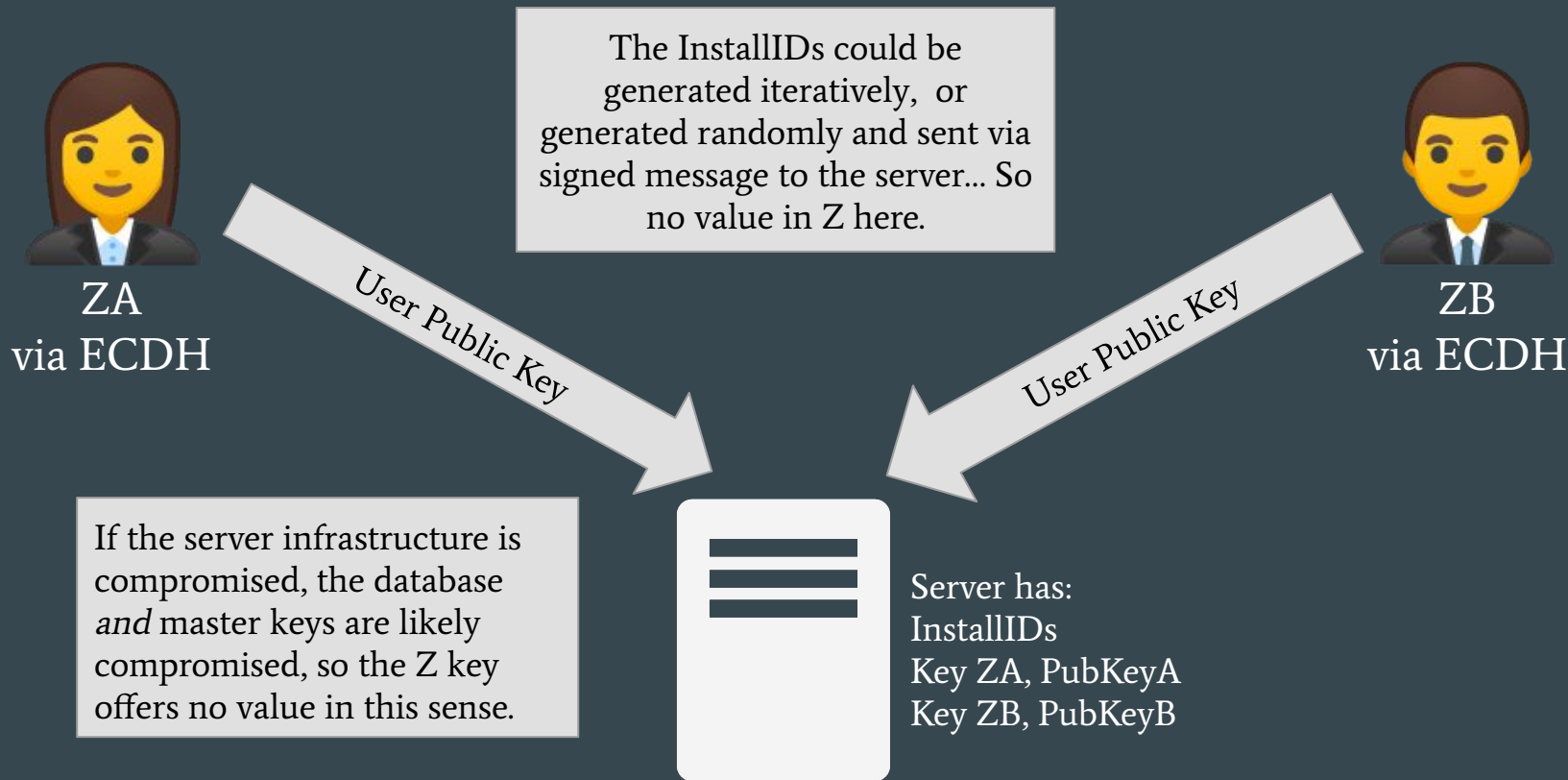
**The single master key design is archaic, and completely unsuitable for a modern system.**

**In this protocol and system, the loss of the master key would lead to total failure of the system - which is a bad design decision.**



The real danger is that such a bad design could likely be replicated and copied by developers unaware of the issues, but observant that 'it was good enough for NHSX'.

# There doesn't seem to be any benefit of the Z key:



# Notes on the Implementation

The app makes use of the BouncyCastle cryptographic library - they claim this is for 'point-value access to the ECC curve'.

However, the ECC implementation there makes use of `java.util.Random` (as opp. to `java.util.SecureRandom`)

```
protected synchronized int next(int bits) {  
    seed = (seed * multiplier + 0xBL) & ((1L << 48) - 1);  
    return (int) (seed >>> (48 - bits));  
}
```

However, BouncyCastle have indicated there should be no issue. But there does not seem to be a straightforward answer as to precisely why it is used for some parts of the cryptography, when the rest of the code seems to use the better JavaX codebase for cryptography.

My overall assessment of the NHSX  
system is that it is just...

...really odd.

# Tim Brown's gathering repo of resources

Tim Brown (@timb\_machine) is gathering information, resources, and articles about the NHSX app(s) here:

<https://github.com/timb-machine/nhsx-contact-tracing-app>

 **timb-machine** / **nhsx-contact-tracing-app**

 Watch ▾

5

★ Unstar

4

 Fork

1

 Code

 Issues 26

 Pull requests 0

 Actions

 Projects 0

 Wiki

 Security 0

 Insights

Tracker for independent privacy and security analysis of NHSX Contact Tracing application

 58 commits

 1 branch

 0 packages

 0 releases

 1 contributor

 CC0-1.0

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

# Resources for the NHSX app

Blog posts by Prof. Bill Buchanan:

<https://medium.com/@billatnapier/the-crypto-on-the-nhsx-covid-19-app-is-a-thing-of-maths-beauty-53b2434d13cb>

<https://medium.com/asecuritysite-when-bob-met-alice/the-uk-contact-tracing-app-version-2-69010db3394d>

NCSC Documentation on the Cryptographic Security of this app:

<https://www.ncsc.gov.uk/files/NHS-app-security-paper%20V0.1.pdf>

Good luck to Zuhlke Engineering in working on the next version of this app - enjoy the £3.8mil.

## NHS tracing app in question as experts assess Google-Apple model

Swiss firm hired to test mainstream software despite launch of go-it-alone system



The NHS Covid-19 tracing app started trials on the Isle of Wight this week © Reuters/FT Montage



BlueTrace - ref. Implementation OpenTrace

# Background to the BlueTrace Protocol

Developed by the Singaporean government for the TraceTogether coronavirus tracing app.

Works differently from the NHS app by the fact that the HA supplies each user with a traceID.

Utilises BLE in a similar way to other protocols and methods.

The open source implementation OpenTrace is not to be confused with the OpenTracing debug s/w.



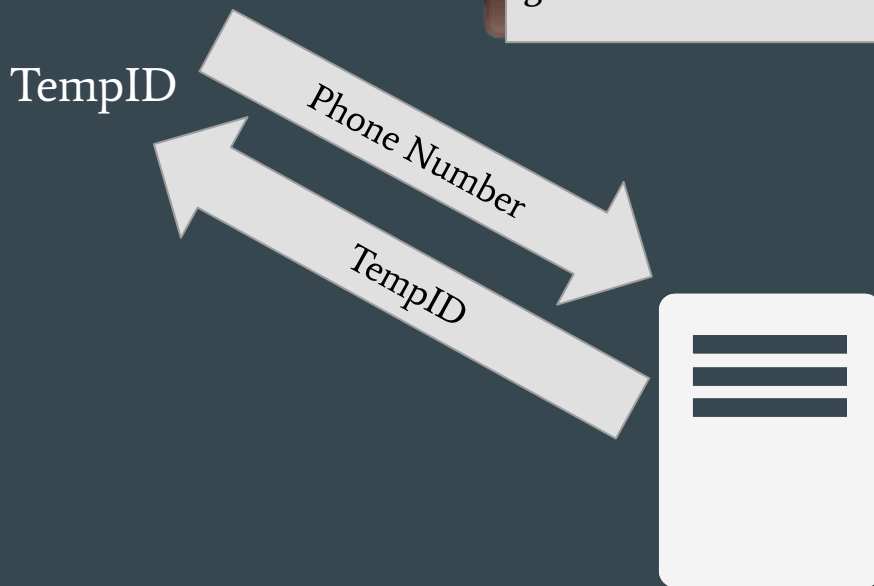
# User Registration

Each user submits their phone number to the main server, which then assigns them a unique, randomly generated UserID.

Each User is given by the server:  
 $C, Auth = AES256GCM(UserID || Start || End)$   
 $TempID = (C || IV || Auth)$

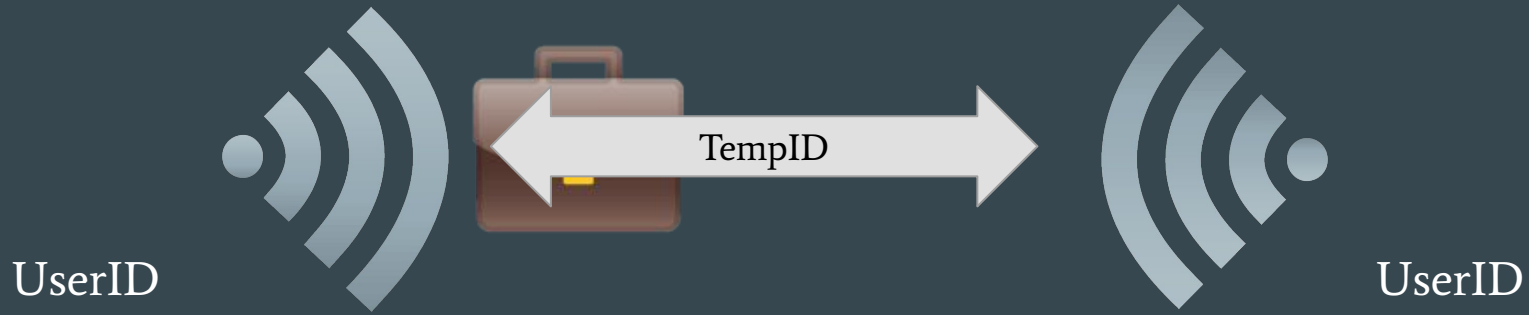
Only the HA knows the secret master AES256 key that is used to encrypt these values.

Temp IDs are time limited, as determined by the server.

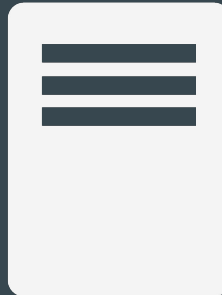


Server has:  
UserID  
PhoneNum

# TempIDs Recorded by Users they come into contact with

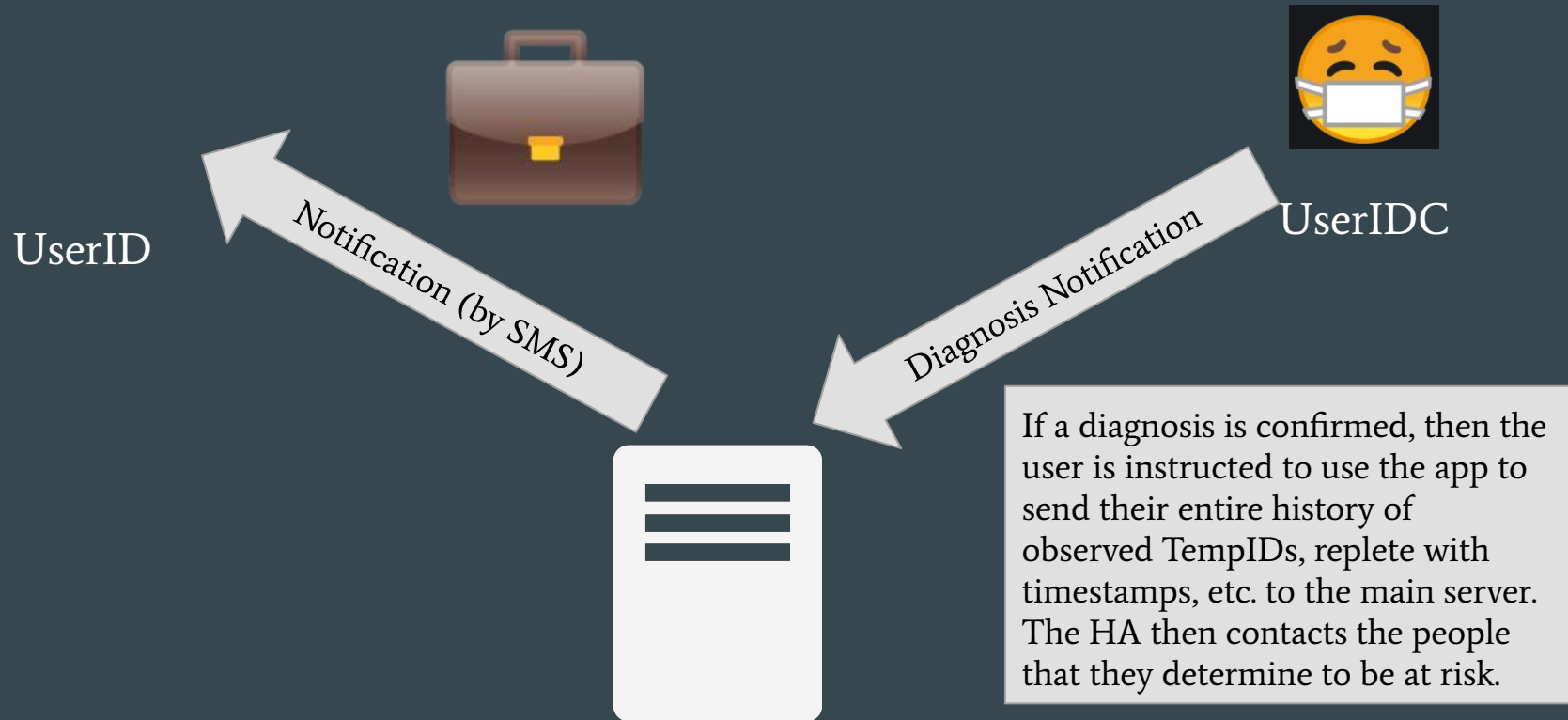


Each user logs their interactions by storing the TempIDs, the RSSI of the BLE, the make/model of the phone, timestamp, etc.



Server has:  
UserID  
PhoneNum  
MasterAESKey

# Diagnosis Notification Workflow



# Benefits and Drawbacks of the BlueTrace approach

Pros	Cons
Complete central control make the user processing minimal - and out of band notification means the app is expendable.	Single Master key for the whole system is a potential single point of failure, as for NHSX.
Strong Cryptographic primitives are used (AES256GCM in particular.)	Opting out relies wholly on the central authority, which has PII for each user.
Users have complete control over whether they send data to the HA. (Good for privacy)	Users have complete control over whether they send data to the HA (bad for calculations and malicious activity detection).

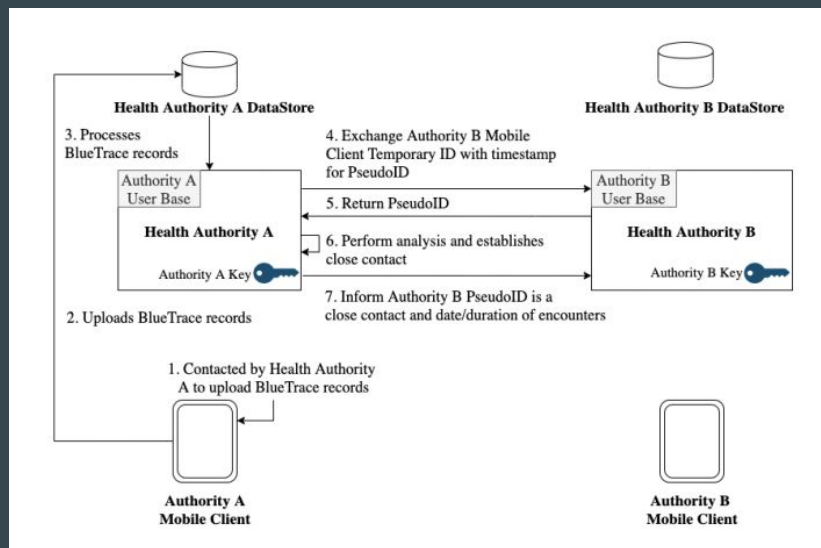
# Notes on the Implementation

The OpenTrace implementation is wholly written in Kotlin - and it should be noted that Kotlin **does not currently have** a good random number generator.

(see: <https://github.com/Kotlin/KEEP/issues/184>)

The Whitepaper has provision for the user to generate their UserID. At the moment they are using Java's RandomUUID through a Kotlin wrapper - this uses java.util.SecureRandom, so all is fine.

Given the centrality of storage, the whitepaper makes provision for the sharing of data between HAs.



# Resources for the BlueTrace Protocol

The BlueTrace Website:

<https://bluetrace.io>

The BlueTrace Whitepaper:

[https://bluetrace.io/static/bluetrace\\_whitepaper-938063656596c104632def383eb33b3c.pdf](https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf)

Opentrace FOSS implementation:

<https://github.com/OpenTrace-community>



The Gapple Protocol



The Appoogle Protocol





The Google-Apple Protocol

# Back story regarding the Google-Apple protocol

First announced on 10th April - everyone has a little cry...

<https://blog.google/inside-google/company-announcements/apple-and-google-partner-covid-19-contact-tracing-technology/>

Germany goes along with this technology on 26th April 2020, having been initially reticent to using it. Singapore cite this as their inspiration for their BlueTrace protocol.

The UK sidesteps the use of the protocol on 28th April 2020 -

<https://healthtech.blog.gov.uk/2020/04/24/digital-contact-tracing-protecting-the-nhs-and-saving-lives/>

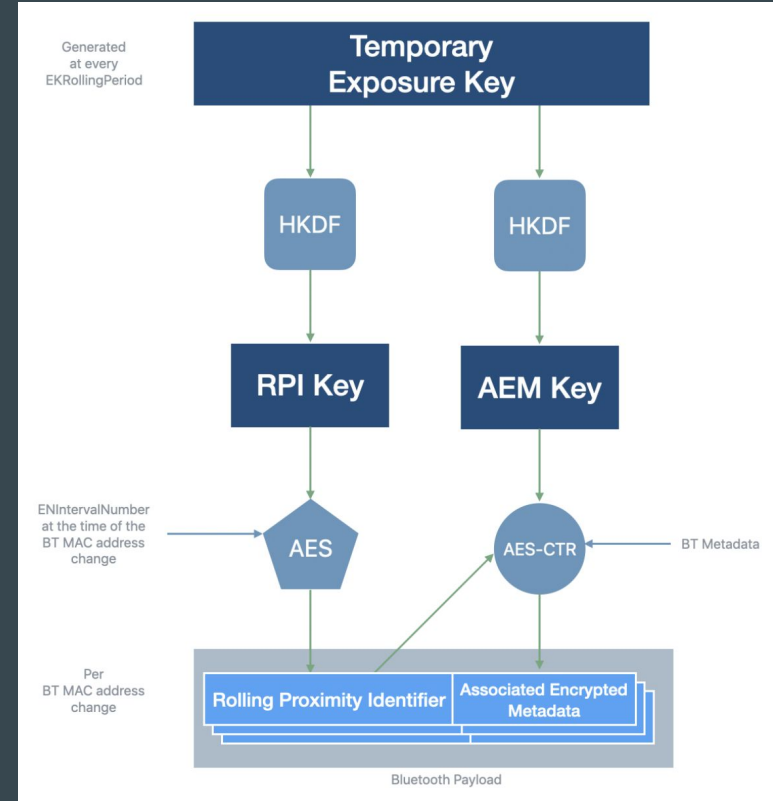
# Overview of Key Terms

- **Exposure Notification Service** — The Bluetooth Low Energy service for detecting device proximity.
- **Temporary Exposure Key** — A key that's generated every 24 hours for privacy consideration.
- **Diagnosis Key** — The subset of Temporary Exposure Keys uploaded when the device owner is diagnosed as positive for the coronavirus.
- **Rolling Proximity Identifier** — A privacy preserving identifier derived from the Temporary Exposure Key and sent in the broadcast of the Bluetooth payload.
- **Associated Encrypted Metadata (AEM)** — A privacy preserving encrypted metadata that shall be used to carry protocol versioning and transmit (Tx) power for better distance approximation. The Associated Encrypted Metadata changes about every 15 minutes, at the same cadence as the Rolling Proximity Identifier, to prevent wireless tracking of the device.

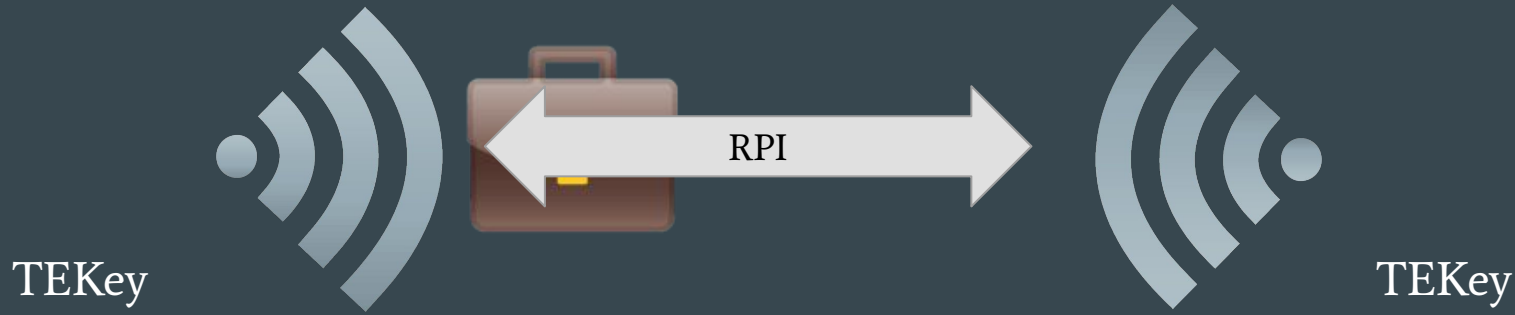
# How Tokens are generated - fully independent of any HA Infrastructure:



The users' devices each generate a Temporary Exposure Key (TEKey) that is fresh for each day. There is no contact with the server at all during this phase.



# How Tokens are generated - fully independent of any HA Infrastructure:

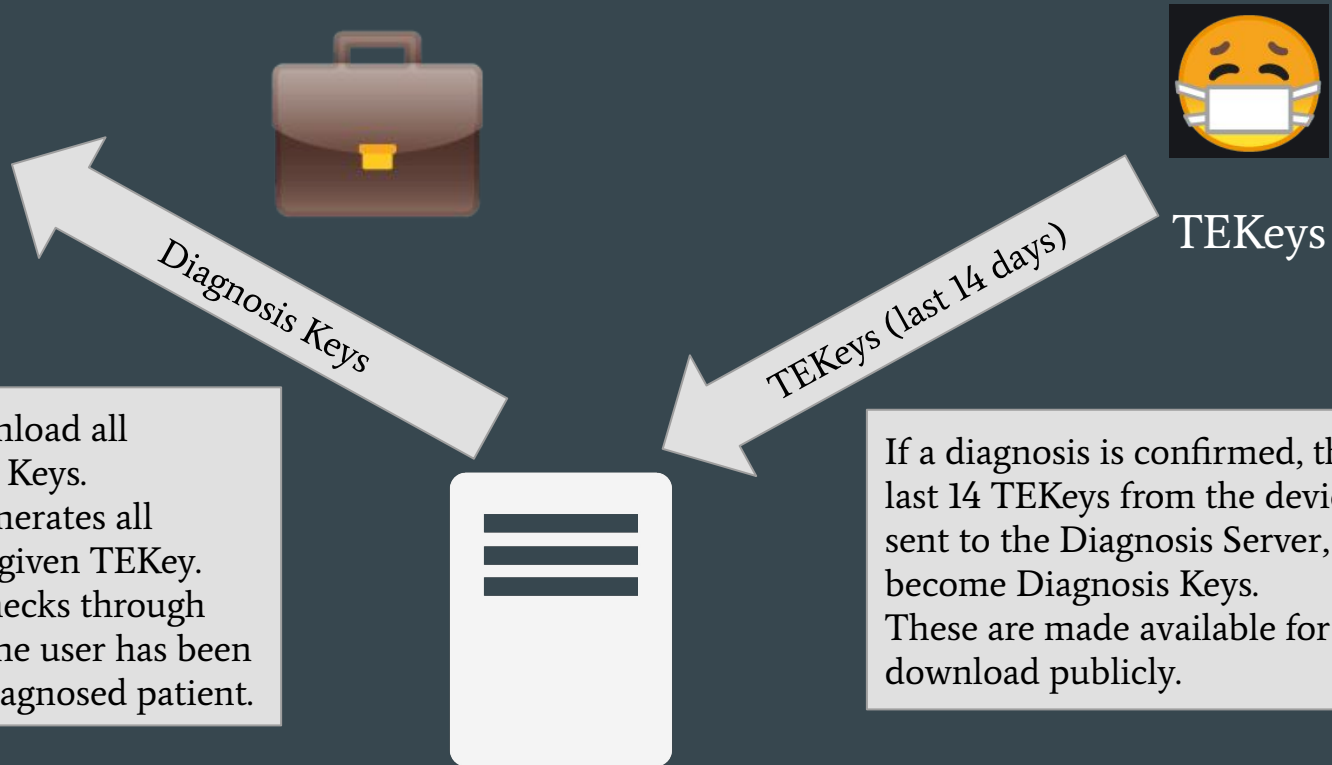


Each device generates Rolling Proximity Keys (RPIK) from the TEKey, which then generates a stream of Rolling Proximity Identifiers (RPI) from  $\text{AES128CTM}(\text{RPIK}, \text{PaddedData})$



Any detected RPIs are stored with 'Associated Encrypted Metadata', with a key that changes at the same rate as the RPIK.

# Diagnosis Notification Workflow



The user apps download all available Diagnosis Keys. The device then generates all possible RPIs for a given TEKey. The system then checks through the AEM to see if the user has been in contact with a diagnosed patient.

If a diagnosis is confirmed, then the last 14 TEKeys from the device are sent to the Diagnosis Server, and become Diagnosis Keys. These are made available for download publicly.

# Benefits and Drawbacks of the BlueTrace approach

Pros	Cons
The app relies on minimal server infrastructure.	Prone to replay attacks (but then again, so are all the others in this talk)
Strong privacy considerations - the app works mostly without any deliberate disclosure.	Relatively weak ciphers used (AES128-CTR) but not detrimentally so.
Suited for low power implementations owing to lightweight algorithms in use.	

An interesting article about issues with this, and other, contact tracing apps: <https://eprint.iacr.org/2020/428.pdf>

# Resources for the Google-Apple Protocol

Original Design Documentation site:

<https://www.apple.com/covid19/contacttracing>

Gavili, Yavron - “SECURITY ANALYSIS OF THE  
COVID-19 CONTACT TRACING  
SPECIFICATIONS BY APPLE INC. AND  
GOOGLE INC.”

<https://eprint.iacr.org/2020/428.pdf>

Apple-Google reference design on Github:

<https://github.com/google/exposure-notifications-android>

Announcement Article:







<https://www.google.com/covid19/exposurenotifications/>












# Overview of the Presented Solutions

Criteria:	NHSX Co-locate	BlueTrace	Apple- Google
Cryptographic Primitives in Use			
Cryptographic Design			
Privacy Considerations			
Faithfulness of Implementations			













# Overview of the Presented Solutions

Criteria:	NHSX Co-locate	BlueTrace	Apple- Google
Cryptographic Primitives in Use			
Cryptographic Design			
Privacy Considerations			
Faithfulness of Implementations			

# Overview of the Presented Solutions

Criteria:	NHSX Co-locate	BlueTrace	Apple- Google
Cryptographic Primitives in Use			
Cryptographic Design			
Privacy Considerations			
Faithfulness of Implementations			

# Overview of the Presented Solutions

Criteria:	NHSX Co-locate	BlueTrace	Apple- Google
Cryptographic Primitives in Use			
Cryptographic Design			
Privacy Considerations			
Faithfulness of Implementations			

## Further Reading:

- **DP-3T** - <https://github.com/DP-3T/documents>
- **PEPP-PT NTK** - <https://github.com/pepp-pt/pepp-pt-documentation>
- **ROBERT** - <https://github.com/ROBERT-proximity-tracing/documents>
- East Coast PACT - <https://pact.mit.edu/>
- West Coast PACT - <https://arxiv.org/abs/2004.03544>
- Anonymous Collocation Discovery: Harnessing Privacy to Tame the Coronavirus - <https://arxiv.org/abs/2003.13670>
- Temporary Contact Numbers (TCN) - <https://github.com/TCNCoalition/TCN>, <https://www.covid-watch.org/>
- TraceSecure: Towards Privacy Preserving Contact Tracing - <https://arxiv.org/abs/2004.04059>

# Acknowledgements

Anne Weine (@AnilamAnn)

Tash (@TashJNorris), Dawid  
(@insomnious0x01), and Emelie  
(@emeliejhp)

Colleagues - Dan, Ashley, Stu, Vlad,  
Evangelos, Adam, and Freddie.

## Questions or Comments?