Contact Tracing

Cryptography Specification

Preliminary - Subject to Modification and Extension

April 2020

Contents

Overview

This document provides the detailed technical specification for cryptographic key scheduling for a new privacy-preserving Bluetooth protocol to support Contact Tracing. Contact Tracing makes it possible to combat the spread of the COVID-19 virus by alerting participants of possible exposure to someone who they have recently been in contact with, and who has subsequently been positively diagnosed as having the virus. This specification complements the Bluetooth specification that contains further information about the scheduling of the advertisements and the higher-level lifecycle of the protocol.

External Functions

Concatenation

We use the symbol || to denote concatenation.

HKDF

HKDF designates the **HKDF** function as defined by IETF RFC 5869, using the SHA-256 hash function:

Output $\leftarrow HKDF($Key, Salt, Info, OutputLength$)$

HMAC

HMAC designates the **HMAC** function as defined by IETF RFC 2104, using the SHA-256 hash function:

Output $\leftarrow HMAC($Key, Data$)$

Truncation

Truncate defines a truncation function:

Output $\leftarrow$ Truncate(Data, L)

The `Truncate` function returns the first L bytes of the data. The input data size being greater or equal to *L* is a precondition.

DayNumber

Provides a number for each 24-hour window. These time windows are based on Unix Epoch Time.

$$\text{DayNumber} \leftarrow \frac{\text{Number of Seconds since Epoch}}{60 \times 60 \times 24}$$

DayNumber is encoded as a 32-bit (`uint32_t`) unsigned little-endian value.

TimeIntervalNumber

Provides a number for each 10-minute window in a 24-hour window as defined by DayNumber.

This value will be in the $[0,143]$ interval.

$$\text{TimeNumberInterval} \leftarrow \frac{\text{Seconds Since Start of DayNumber}}{60 \times 10}$$

where : Seconds Since Start of DayNumber ← Number of Seconds since Epoch % (60*60*24)

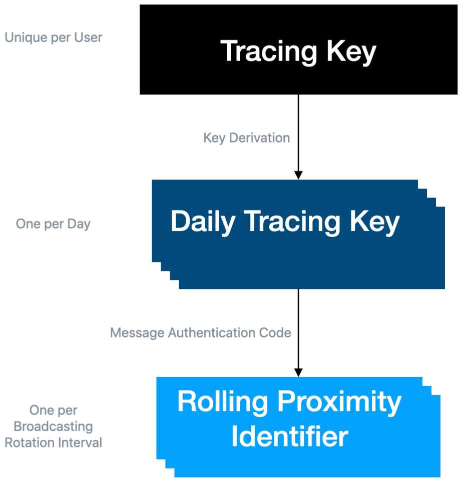
TimeNumberInterval is encoded as a 8-bit (`uint8_t`) value.

CRNG

CRNG designates a cryptographic random number generator.

Output $\leftarrow CRNG($OutputLength$)$.

Key Schedule for Contact Tracing


Unique per User
Tracing Key
Key Derivation
One per Day
Daily Tracing Key
Message Authentication Code
One per Broadcasting Rotation Interval
Rolling Proximity Identifier

Tracing Key
The *Tracing Key* is generated when contact tracing is enabled on the device and is securely stored on the device.

The 32-byte Tracing Key is derived as follows:

$$tk \leftarrow CRNG(32)$$

The Tracing Key never leaves the device.

Daily Tracing Key
A *Daily Tracing Key* is generated for every 24-hour window where the protocol is advertising.

From the Tracing Key, we derive the 16-byte Daily Tracing Key in the following way:

$$dtk_i \leftarrow HKDF(tk, NULL, (\text{UTF8("CT-DTK")} \,||\, D_i), 16)$$

where D_i is the DayNumber for the 24-hour window the broadcast is in.

Upon a user testing positive, the Daily Tracing Keys for days where the user could have been affected are derived on the device from the Tracing Key. We refer to that subset of keys as the Diagnosis Keys. If a user remains healthy and never tests positive, these Daily Tracing Keys never leave the device. The use of 16-byte keys allows to limit the server and device requirements for the transfer and storage of the Diagnosis Keys while preserving low false positive probabilities.

Rolling Proximity Identifier
The *Rolling Proximity Identifiers* are privacy-preserving identifiers that are sent in Bluetooth Advertisements.

Each time the Bluetooth MAC address changes, we derive a new Rolling Proximity Identifier:

$$RPI_{i,j} \leftarrow \text{Truncate}(HMAC(dkt_i, (\text{UTF8("CT-RPI")} \,||\, TIN_j)), 16)$$

Where:

- TIN_j is the TimeIntervalNumber for the time at which the BLE MAC address changes.

The 16-byte Rolling Proximity Identifier is broadcasted over Bluetooth LE. The use of 16-byte identifiers yields a low probability of collisions and limits the risk of false positive matches, while keeping device storage requirements low.

Matching Values from Users Tested Positive

Upon a positive test of a user for COVID-19, their Diagnosis Keys and associated DayNumbers are uploaded to the Diagnosis Server. A Diagnosis Server is a server that aggregates the Diagnosis Keys from the users who tested positive and distributes them to all the user clients who are using contact tracing.

In order to identify any exposures, each client frequently fetches the list of Diagnosis Keys. Since Diagnosis Keys are sets of Daily Tracing Keys with their associated Day Numbers, each of the clients are able to re-derive the sequence of Rolling Proximity Identifiers that were advertised over Bluetooth from the users who tested positive. In order to do so, they use each of the Diagnosis Keys with the function defined to derive the Rolling Proximity Identifier. For each of the derived identifiers, they match it against the sequence they have found through Bluetooth scanning. Additional validation can be used to confirm that the advertising happened in a time window comparable to the one expected based on the TimeIntervalNumber.

Matches must stay local to the device and not be revealed to the Diagnosis Server.

Privacy Considerations

- The key schedule is fixed and defined by operating system components, preventing applications from including static or predictable information that could be used for tracking.
- A user’s Rolling Proximity Identifiers cannot be correlated without having the Daily Tracing Key. This reduces the risk of privacy loss from advertising them.
- A server operator implementing this protocol does not learn who users have been in proximity with or users’ location unless it also has the unlikely capability to scan advertisements from users who recently reported Diagnosis Keys.
- Without the release of the Daily Tracing Keys, it is not computationally feasible for an attacker to find a collision on a Rolling Proximity Identifier. This prevents a wide-range of replay and impersonation attacks.
- When reporting Diagnosis Keys, the correlation of Rolling Proximity Identifiers by others is limited to 24h periods due to the use of Daily Tracing Keys. The server must not retain metadata from clients uploading Diagnosis Keys after including them into the aggregated list of Diagnosis Keys per day.

Test Vectors

Test vectors for interoperability testing between implementations of this specification are available upon request in a machine readable format.