Exposure Notifications

Off-Device Venue Check-Ins

Preliminary — Subject to Modification and Extension

September 2021

v1.0

Contents

Overview 3

Off-Device Venue Check-Ins 4

Revision History 8

Overview

This document details how a Public Health Authority (PHA) can use Off-Device Venue Check-Ins to alert users when someone reports a positive test result for COVID-19, the disease caused by the SARS-CoV-2 virus. This feature notifies other people who've been at the same indoor location at the same time as someone who reports a positive test result through an app on a phone or other device.

A PHA can include a feature in their Exposure Notifications app that allows people to Check-In at a specific venue, by scanning a Quick Response (QR) code or Near-Field Communication (NFC) device. This feature can notify people who were at the same venue around the same time as a potential exposure, even if Exposure Notifications didn't record them being near a person at the venue who reported a positive test result. This feature is not a part of Exposure Notifications and users must opt-in to use the feature in the PHA's app and separately opt-in to share their Check-In information after a positive test result.

There are two options for performing Check-Ins:

1. On-Device.

   Venue Check-Ins stay on the device. Human contact tracers identify venues where people who report positive test results have recently visited during a period when they were likely contagious. These contact tracers manually enter venue identifiers and approximate date and time when someone who tested positive was at the venue, into the PHA server. The server periodically pushes this information to the device, and the device compares the information against the local Check-Ins. If there's a match, the system warns the user they may have been exposed.

2. Off-Device.

   A device uploads cryptographically protected venue Check-Ins to the PHA server only when the user tests positive and agrees to sharing. The Check-Ins are anonymously sent off the device, and then pushed from the server to all devices for matching. If a Check-In from the server matches a Check-In on the device, the system warns the user they may have been exposed. This option doesn't need human contact tracers to input any information.

Off-Device venue Check-Ins require positive users to share their Check-In information, and remove the need for human verification; matching occurs automatically. On-Device Check-Ins stay on the device, which gives users more control over their private information.

When a user opts-in to sharing their venue Check-In information, the PHA app sends only limited data to the PHA's server, for example, a venue identifier, the approximate date and time when the user was at the venue, and a transmission risk value. It must not send any personally identifiable information about the user when it sends the Check-In information. The PHA's app cryptographically protects the data it sends to the PHA, and must not send any data to Apple or Google. Neither Apple, Google, nor the PHA have access to the protected Check-In information.

Definitions

- Check-In Mechanism — A QR Code, bar code, NFC device, or any other automated mechanism people can use with their phones to Check-In at a venue.

- Check-In — The event when a person uses the Check-In Mechanism to record on their phone the fact that they were at the venue at a specific time.

- Check-In Protected Report — A cryptographically protected report derived from a Check-In.

- EN — Exposure Notifications, as developed jointly by Apple and Google.

- TRL — Transmission Risk Level: a coarse value based on testing and venue information.
- Venue — Any event, venue, or other location that offers Check-In.

Off-Device Venue Check-Ins

When a person uses the Check-In Mechanism, their phone receives and records data provided by that mechanism. That data must include the venue's cryptographically secure random seed value, which must contain at least 128 bits of entropy. The data received from the Check-In Mechanism can optionally include other information in addition to the cryptographic seed, such as a venue name, address, or a default attendance duration. The venue must generate the cryptographic seed, locally, on device and must not share it with any central entity or database — including any owned or managed by the PHA — nor share any other value from which a central entity or server might derive the seed.

If a PHA uses an NFC device or any other device capable of two-way communication, the phone must not send any user-identifiable or device-identifiable information back to the Check-In Mechanism.

User Check-In

When a person arrives at a venue that supports the Check-In System, they can use their phone to Check-In. When they use the Check-In Mechanism, their app must securely store the venue's cryptographic seed along with the date and time of the scan. During Check-In, the venue must not record any information.

Check-In data must only be stored locally on the user's device. The app must delete a Check-In permanently when 14 days have passed since the Check-In. iPhone backups do not include Check-In data, and users can view all Check-In data in the app. The app must provide a way for the user to delete individual Check-Ins, as well as have an option to delete all Check-In System data.

After a Check-In, with the user's permission, their phone will periodically download cryptographically protected data from the PHA's server so that the PHA's app can check whether a person at the venue around the same time as someone who reports a positive test by uploading Check-In information, which might indicate a potential exposure.

Check-In Protected Reports

A PHA that supports the Check-In System can optionally allow users who have tested positive to share the Check-In information stored on their devices. If a PHA offers this feature, anyone with a Check-In who receives a positive test result for COVID-19 can choose to share their cryptographically protected Check-In information with the PHA's server. This allows other users to be notified when their device detects that they were at the venue around the same time as a user who shared a positive test result. The app must not use or send any identifying information as part of this process.

When a user with a previous Check-In tests positive and consents to sharing their information, the app generates a Check-In Protected Report for any Check-In from the previous 14 days that the user selects. The Check-In Protected Reports contain a venue identifier along with relevant metadata, such as the approximate date and time of the user Check-In, their calculated infectiousness, and their transmission risk level (TRL). With the user's permission, the app sends the selected Check-In Protected Reports to the PHA's server.

To create a Check-In Protected Report, the app first creates a presence record by applying a one-way cryptographic function to the Check-In data, which must include the venue's cryptographic seed received from the Check-In Mechanism. The app requires the original cryptographic seed received from the Check-In Mechanism in order to access the presence record. That requirement means that other devices that Check-In at that venue can link the report back to the venue. The PHA and devices that didn't Check-In at the venue don't have the cryptographic seed required to link the report back to a venue or location.

Once the app generates a presence record, it creates the Check-In Protected Report. The app can do this, for example, by encrypting the presence record along with any relevant metadata, using the presence record as an encryption key derived from the venue's cryptographic seed that it receives from the Check-In Mechanism.

> **Note**: Apple recommends using an authenticated encryption scheme that uses AES-CTR plus HMAC-SHA256 for integrity protection. The overhead is at most 48 bytes (16-byte IV and 32-byte HMAC).
>
> ```
> PresenceRecordAES || PresenceRecordHMAC = HKDF(Cryptographic Seed)
> IV = CRNG(16bytes)
> EncryptedCheckin =  AES-CTR(PresenceRecordAES, IV, Visit Start ||
>     Visit End || TRL)
> CheckInProtectedReport = IV || EncryptedCheckin ||
>     HMAC - SHA256(PresenceRecordHMAC, IV || EncryptedCheckin)
> ```
>
> The information the app supplies to the key derivation function must include the venue's cryptographic seed, but may also include other information. iOS offers native APIs for all cryptographic functions above. For detailed information on these algorithms see https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-CryptographySpecificationv1.2.pdf and https://doi.org/10.6028/NIST.FIPS.198-1 (for HMAC-SHA256).

Apps must follow additional requirements when generating a Check-In Protected Report:

1. The app must not send a user's Check-In data or any data associated with, or derived from that Check-In data, without the user's explicit and informed consent.
2. The app must protect all venue information by using the cryptographic one-way function discussed above.
3. Check-In Protected Reports can only contain metadata as described below.
    A. An app should protect all metadata by using the cryptographic one-way function described above. Metadata must only be accessible using the venue's cryptographic seed that the user's device receives from the Check-In Mechanism.

    B. The only metadata allowed is limited to the Check-In start and end time, and limited information derived from the user diagnosis, such as infectiousness, report type, and transmission risk level.

    C. The metadata must not contain any personally identifiable information, device-identifiable information, location information, or information produced by Exposure Notifications.

    D. The app must compute Transmission Risk Level only from information related to the Covid test, test date, symptom onset, and venue configuration.

    E. If the metadata includes start and end times, the app must specify those times using increments of not less than 5 minutes.

        1. If the app doesn't protect the metadata by using the cryptographic one-way function described above, then it must specify the start and end times using increments of not less than 24 hours.
        2. The only timing information that the app may specify is a particular calendar day. No information on the time of day may be used.

    F. The app must not share any metadata or cryptographic material that allows access to venue information with any central entity, including the PHA.
4. Apps must send Check-In Protected Reports in batches to the PHA's server. Additionally, the app must also send random data to help prevent any third party making a precise estimation of the number of valid Check-In Protected reports in the batch by observing the encrypted communication channel.

5. The app must shuffle the order of Check-In Protected Reports in the batch before it sends these reports to the server.

App UI Requirements

PHAs that implement the Check-In System must include additional user-facing functionality in their app.

1. If included in an Exposure Notifications app, the Check-In System must be offered as an optional feature that requires user consent. Exposure Notifications functionality must continue to work if the user does not grant permission for the Check-In System.
2. The app must provide the user with a readily available, human-readable list of all venues where they have checked-in, along with any relevant associated data that the app stores about that Check-In, such as its time and date.
3. The app must provide the total number of non-human-readable records currently stored on the device.
4. The app must provide a way for users to delete Check-In records.
   A. The App must provide the ability to delete individual records and to delete all the Check-In System data as a single action.
   B. When the user deletes a Check-In, it must be permanently removed from the device, along with any associated or derived data.
5. The app must prompt users for permission to check whether the user has checked in to a venue around the same date and time as a user who has reported positive and has chosen to share their Check-In Information. This must be a separate permission request from the one the app uses for Exposure Notifications key matching.
6. The app must request permission in a way that is neutral and does not guide the user toward a specific response. For example, the permission prompt must use the same size, color, font, and font attributes for both the "yes" and "no" options.
7. The app must disclose to its user before asking them for permission to submit a Check-In Protected Report that submitting the Check-In Protected Report risks revealing their identity and location.
8. When an app prompts the user to share venue Check-In information, all Check-Ins must be initially unselected. The user must explicitly select a Check-In to share it.
9. The app must tell the user exactly what data it will submit to the PHA's server prior to sending the data. It must prompt again every time it requests to submit Check-In data.

App Implementation Requirements

1. The Check-In System is a separate feature from EN. Accordingly, the app must use a completely separate flow for implementing the Check-In System.
2. The app must keep Check-In data exclusively on the user's device and must not send it to any server (except as part of a Check-In Protected Report), or back it up in the app's datastore.
3. Apps must check locally on the device whether the user has checked in to a venue around the same date and time as a user who has reported positive and has chosen to share their Check-In Information.
4. Apps must protect the information they use to do local Check-In matching by using encryption or hashing that uses the seed received from the Check-In Mechanism in a way that doesn't reveal user venue Check-In Information to other visitors who used the Check-In Mechanism at the venue.
5. The app must store all information related to Check-In activity with data protection Class A: Complete Protection or Class B: Protected Unless Open, except for data necessary to do on-device matching. For more information on

data protection classes, see https://support.apple.com/guide/security/data-protection-overview-secf6276da8a/web

6. The app must use local notifications to inform users about potential exposure at a venue. The app must not transmit Check-In data (including associated or derived data) to a server or other device except using Check-In Protected Reports as described in this document, and the app must not transmit data related to matching to a server or central authority.
7. The app must automatically and permanently delete any Check-In records and all associated and derived data once the Check-In is more than 14 days old.
8. The app must not share any data related to, or derived from Check-Ins, except with explicit consent of the user and as described in this document.
9. The app must not log or transmit IP addresses or device identifiers.

Server Requirements

1. Servers must shuffle the Check-In Protected Reports they receive from users so that Check-Ins from the same user are not grouped together. No two Check-In Protected Reports can be identifiable as coming from the same user.
2. Servers must not log IP addresses or device identifiers.
3. Server must not use any information submitted by devices to profile, locate or re-identify the submitting users.

Venue Requirements

The PHA is expected to make reasonable efforts to prominently notify Venues (unless otherwise required by applicable law) that:

1. Venues must not require attendees to Check-In using the app, or make services (such as access to public transit or buildings) contingent on a user Check-In using the app.
2. Venues must provide an alternate way to Check-In that doesn't require the use of the Check-In System feature.

Public Health Authority Requirements

1. PHAs must not attempt to profile or identify users, for example, by mapping the physical venue to the Check-In Mechanism cryptographic seed or by any other means.
2. PHAs must publicly publish a buildable copy of the source code for their EN app including any external libraries or dependencies. PHAs must update the published copy with every subsequent submission to the App Store.

Revision History

v1.0 - September 1, 2021

- Initial version.