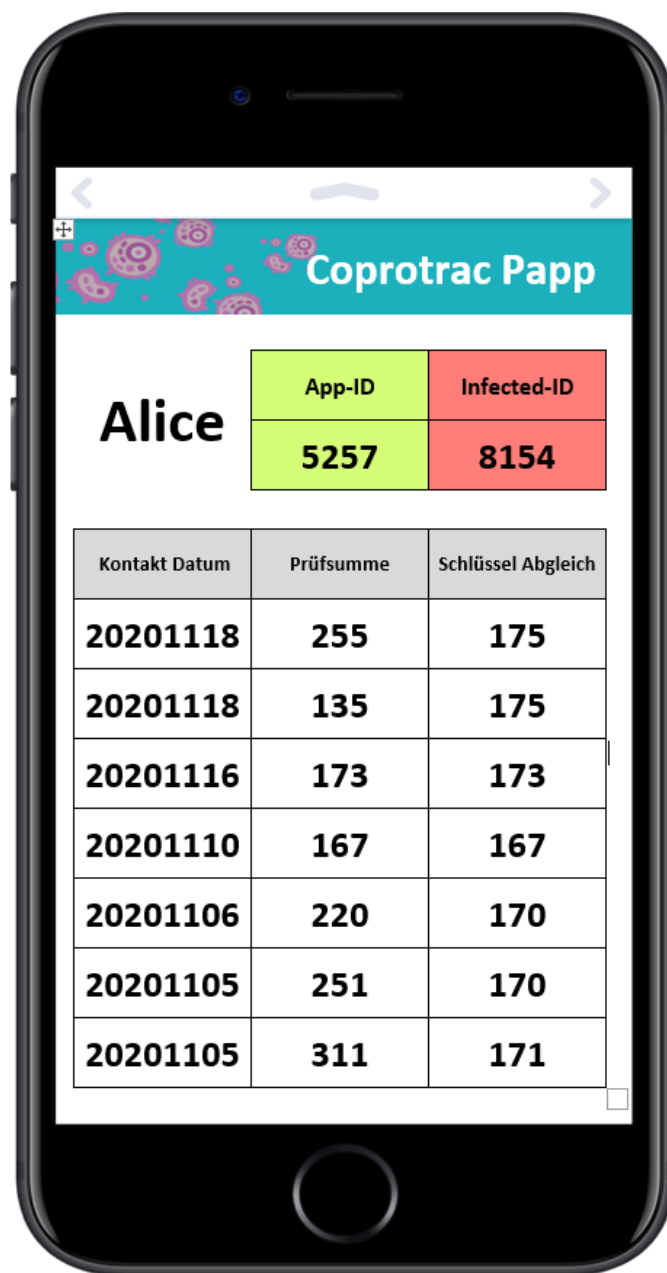


Corona Proximity Tracing Paper App

Spielanleitung



Coprotrac Papp

Spielidee

Sogenannte Proximity-Tracing-Apps registrieren, wer zu welcher Zeit mit anderen (infizierten) App-Nutzern in Kontakt war. Die Basis dieses möglichst anonymen Austausches von Informationen bilden kryptographische Verfahren.

Dieses Spiel zeigt Schritt für Schritt, wie solche Apps technisch funktionieren.

Die Coprotrac Papp zeichnet wie eine echte Smartphone-App Kontakte auf und zeigt an, falls ein gespeicherter Kontakt später einen positiven Covid-19-Test eingibt.

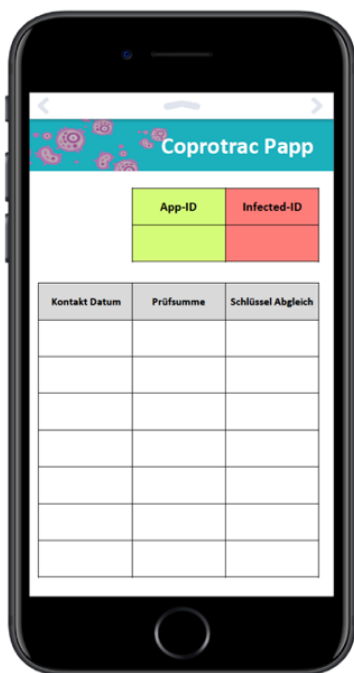
Vorbereitung

Coprotrac Papp ist ein Spiel für maximal 12 Personen plus die Spielleiterin oder den Spielleiter, ist aber auch mit weniger Leuten durchführbar.

In einer Schulklasse mit 24 Schülerinnen und Schülern kann Coprotrac Papp auch in zwei unabhängigen Gruppen plus einer Lehrperson als Spielleiter/in gespielt werden.

Es ist keine Voraussetzung, aber die Spielleiterin, der Spielleiter bzw. die Lehrperson kann sich über Proximity-Tracing-Apps schlau machen mit der Lektüre des Republik-Artikels [So funktioniert eine Corona-Tracing-App, die Ihre Privatsphäre schützt.](#)

Die Spielleitung begleitet die Spielerinnen und Spieler Schritt für Schritt mit Erklärungen zum Ablauf und Vorgehen (in dieser Anleitung bis zur Stelle «Gleich noch einmal» und wieder ab «Ich wurde infiziert»).



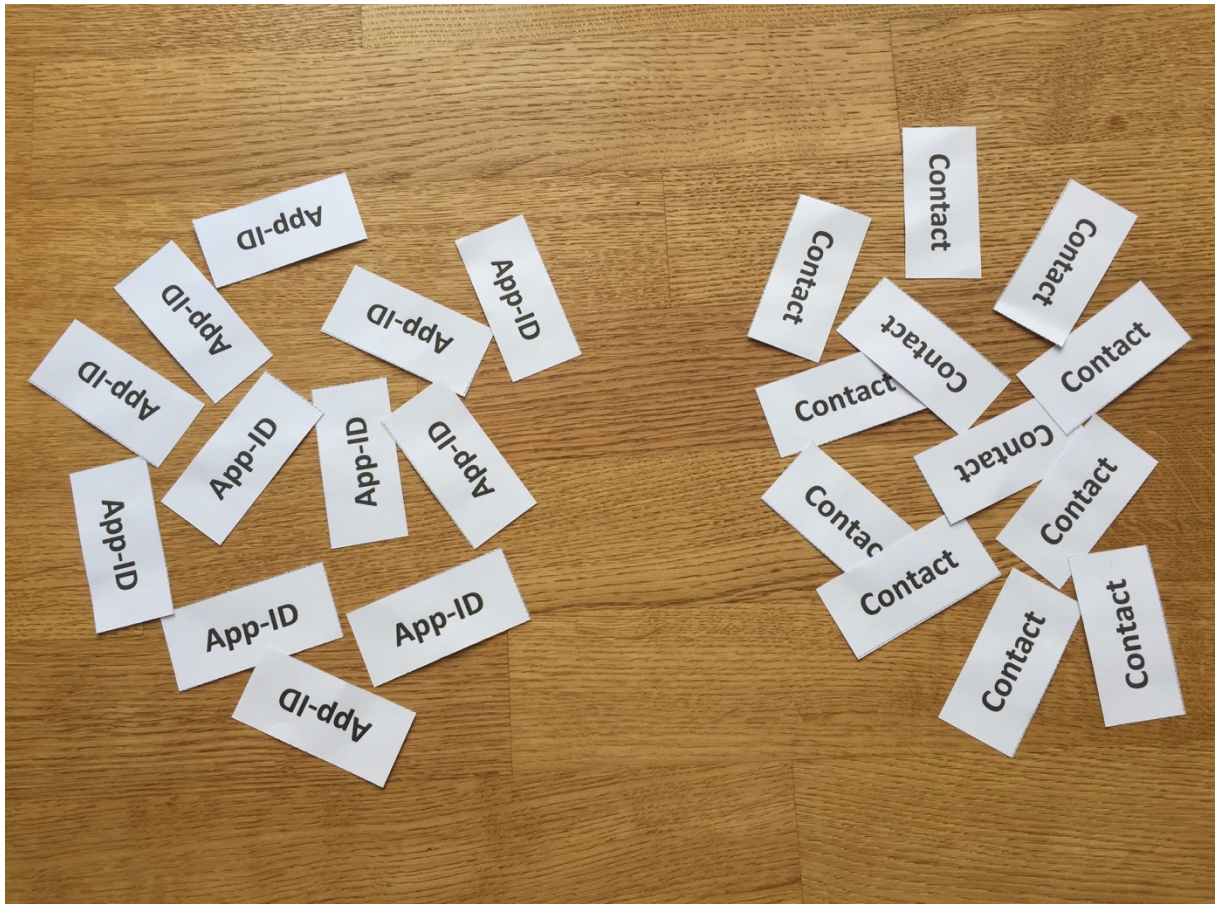
Jede/r Spieler/in erhält die auf Papier ausgedruckte PDF-Datei *Coprotrac_Papp_Phone_Screen.pdf*

Coprotrac Papp



Die Kärtchen für die App-IDs (App-IDs.docx) und die Contact-Events (Date-Stamped.docx) werden (falls möglich doppelseitig) ausgedruckt, ausgeschnitten und mit der Rückseite nach oben an zwei verschiedenen Stellen ausgebreitet.

Die Anzahl der App-ID Kärtchen muss der Gruppengrösse entsprechen (bei nur 10 Spielerinnen und Spielern also auch nur 10 App-ID Kärtchen).



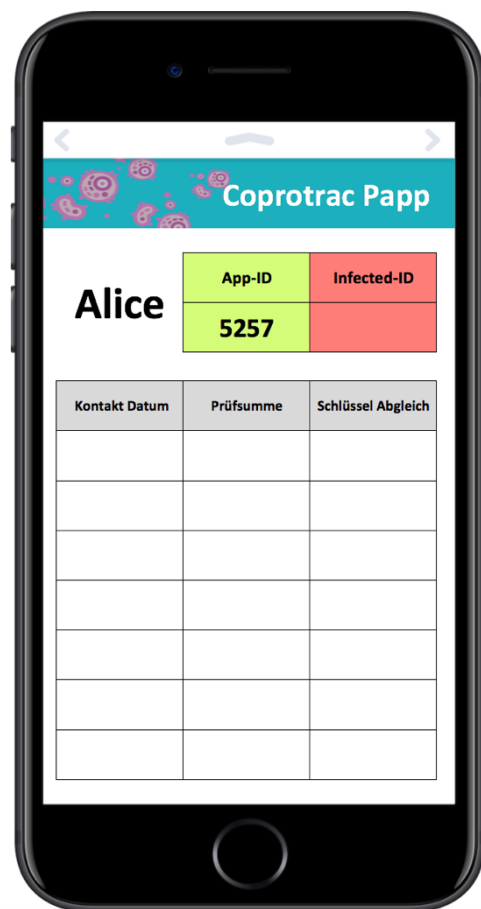
Coprotrac Papp

Spielverlauf

Persönlicher, aber anonymer Geheimschlüssel

Jede/r Spieler/in zieht eine App-ID und notiert diesen Geheimschlüssel in das leere grüne Feld **App-ID** der Coprotrac Papp. Das App-ID Kärtchen wird wieder verdeckt zurückgelegt. Die App-ID Kärtchen werden vorerst nicht mehr benötigt und können von der Spielleitung weggelegt werden.

Zum Beispiel zieht Alice den geheimen Schlüssel «5257».



Und in echt?

Jede Proximity Tracing-App generiert bei der Installation einen geheimen Schlüssel und speichert diesen lokal auf dem Smartphone ab. Dieser private Schlüssel ist rein zufällig und beinhaltet keine persönlichen Daten. So wird jede individuelle App-Installation und darum auch die Person anonym identifiziert.

Hier das reale Beispiel einer App-ID:

«44c6dfb4cbdbea397122d47f9e0bfe397aafcad3a38db91a13d617ec4a3cfa19».

Coprotrac Papp



Hallo Nachbar

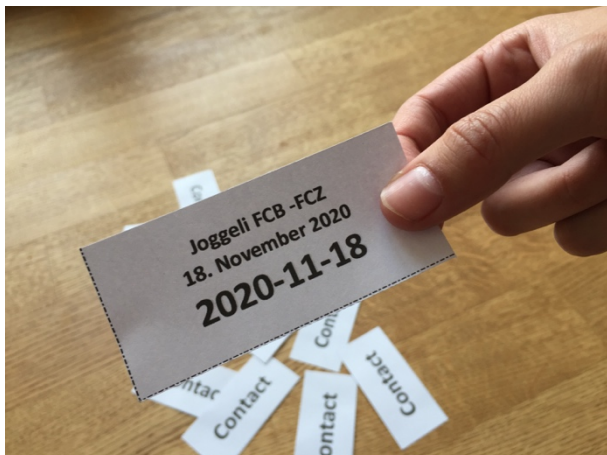
Nun findet ein Kontakt mit einer anderen Coprotrac Papp in der Nähe statt.

Die Spieler/innen gruppieren sich zu zufälligen Zweiergruppen und jede Gruppe zieht ein Contact Kärtchen.

Bei einer ungeraden Anzahl Spieler/innen hat eine Person im Moment keinen Kontakt.

Zum Beispiel treffen sich Alice und Bob und sie ziehen folgendes Contact Kärtchen:

«Joggeli FCB -FCZ, 18. November 2020, 2020-11-18».



Und in echt?

Eine Handy Proximity-Tracing App sendet alle paar Sekunden per Bluetooth ein Datenpaket in die nahe Umgebung aus: «Hallo, hier ist die Coprotrac Papp. Is there anybody out there?»

Coprotrac Papp

Datenaustausch

Die Apps von Alice und Bob tauschen nun Daten aus, um sich diesen Kontakt zu merken. Dabei handelt es sich um das Kontaktdatum und eine sogenannte Prüfsumme.

Unser Datumsstempel des FCB-Matches wird in der Form 20201118 (für 18.11.2020) übermittelt. In echt besitzt dieser Time Stamp noch eine Zeitangabe, wie zum Beispiel «2020-11-18T20:13».

Alice und Bob tragen nun beide die Zahl 20201118 in das Feld «Kontakt Datum» ihrer Coprotrac Papp ein.

The image shows two smartphones side-by-side, both displaying the 'Coprotrac Papp' interface. The left phone is for 'Alice' and the right for 'Bob'. Both screens show a table for recording contact data. The 'Kontakt Datum' column in both tables contains the value '20201118'.

Kontakt Datum	Prüfsumme	Schlüssel Abgleich
20201118		

Coprotrac Papp

Berechnung der Prüfsumme

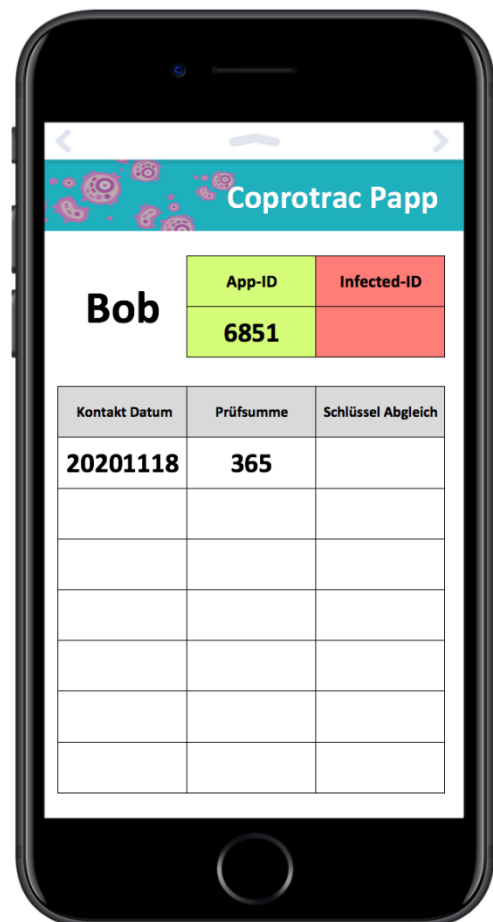
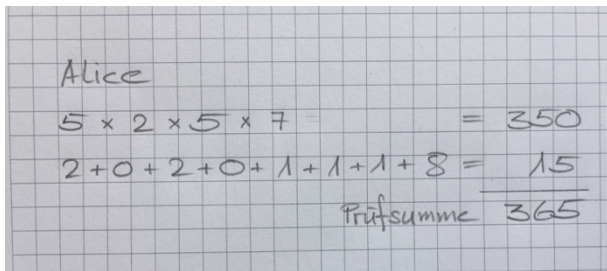
Die Apps von Alice und Bob machen sich an die Arbeit: Sie nehmen den aktuellen Datumsstempel und ihren persönlichen Schlüssel und geben beides in eine kryptographische Funktion ein, welche eine Verschlüsselung vornimmt. Heraus kommt eine Prüfsumme.

In unserer Coprotrac Papp berechnet sich die Prüfsumme folgendermassen:

- Multipliziere alle Ziffern der eigenen App-ID
- Berechne die Quersumme des Datumsstempels
- Addiere diese beiden Zahlen
- Übermittle diese Prüfsumme

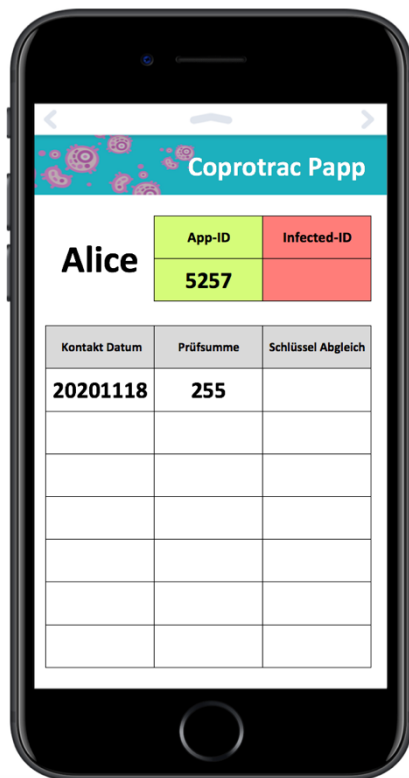
Alice rechnet also:

- Multiplikation der Ziffern der eigenen App-ID = $5 \times 2 \times 5 \times 7 = 350$
- Quersumme des Kontakt Datums = $2 + 0 + 2 + 0 + 1 + 1 + 1 + 8 = 15$
- Prüfsumme = $350 + 15 = 365$
- Alice sendet die Prüfsumme 365 an Bob, welche dieser in seiner App zum entsprechenden Datum speichert.



Bob rechnet:

- Multiplikation der Ziffern der eigenen App-ID = $6 \times 8 \times 5 \times 1 = 240$
- Quersumme des Kontakt Datums = $2 + 0 + 2 + 0 + 1 + 1 + 1 + 8 = 15$
- Prüfsumme = $350 + 15 = 255$
- Bob sendet die Prüfsumme 255 an Alice, welche diese in ihrer App zum entsprechenden Datum speichert.



Der Clou dieser kryptographischen Funktion ist, dass für den gleichen Schlüssel und Datumsstempel immer die gleiche Prüfsumme herauskommt – und für einen leicht anderen Schlüssel oder Zeitstempel eine ganz andere Prüfsumme. Die Funktion ist auch nicht umkehrbar: Man kann sie nicht zurückrechnen. Diese Einwegfunktion lässt darum keine Rückschlüsse auf Bob bzw. seine App zu.

Versucht es doch einmal: Aus dem Teil der Prüfsumme 240, die aus der Multiplikation der App-ID von Bob generiert wurde, kann die geheime App-ID 6851 von Bob nicht wiederhergestellt werden. Es gibt nämlich verschiedene Möglichkeiten, dass die Multiplikation der einzelnen Ziffern einer (vierstelligen) Zahl 240 ergibt: z.B. 1685, 6445, 2853, 5344, 8156, 4564 etc.

Man kann sich die Prüfsumme wie ein Schloss vorstellen, das nur mit dem richtigen Zeitstempel und dem richtigen Schlüssel geöffnet werden kann.

Und in echt?

Das verwendete Prinzip ist ein Code für eine Nachrichten-Authentifizierung, der auf einem Schlüssel und einer kryptographischen Hash-Funktion beruht.

Die Funktion nimmt als Eingabe einerseits einen Schlüssel und andererseits eine Nachricht (in unserem Fall den Datumsstempel) und liefert als Ergebnis eine Prüfsumme, die zufällig aussieht und unabhängig von Schlüssel und Nachricht immer gleich lang ist. Man spricht deshalb von einer *compression function*, die Informationen werden komprimiert.

Auf einem Smartphone sind die IDs und Prüfsummen natürlich nicht nur drei- oder vierstellig und können auch mit den schnellsten Rechnern der Welt nicht zurückgerechnet werden.

Reales Beispiel:

Prüfsumme (Schlüssel + Zeitstempel) =

44c6dfb4cbdbea397122d47f9e0bfe397aafcad3a38db91a13d617ec4a3cfa19

+ 2020-04-07T22:13:00 =

923a3b223fd620e788583abac3758410e724481a1756af514e588f3f7427317d



Ich bin ein Computer

Wir benutzen in unserer Paper-App viel kleinere Zahlen als eine «echte» Proximity Tracing Anwendung. Trotzdem darf uns beim Berechnen der Prüfsumme von Hand oder mit dem Taschenrechner kein Rechenfehler unterlaufen, da sonst unser Spiel nicht korrekt funktionieren würde.

Gleich noch einmal

Diese Kontaktrunde ist nun beendet und das Contact Kärtchen wird wieder zurückgelegt. Es entstehen neue Kontakte, indem sich die Spieler/innen in neuen Zweiergruppen treffen, eine Contact Karte ziehen und die Prüfsummen gegenseitig austauschen. Dies wird solange wiederholt, bis der Speicher unserer Coprotrac Papp (fast) voll ist.

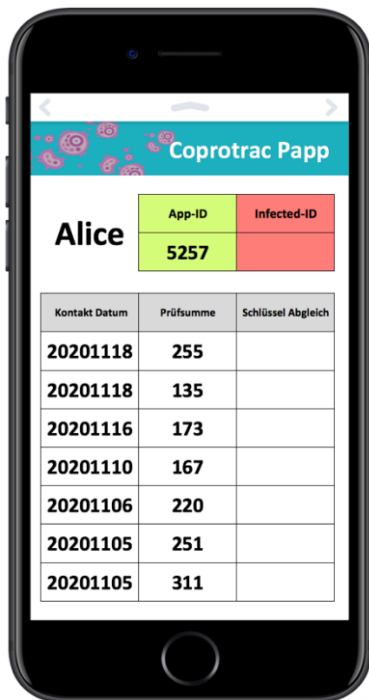
Es ist natürlich möglich, dass sich Alice und Bob wieder treffen bzw. sogar ein zweites Mal im Joggeli Stadion am FCB - FCZ Match. Dies wäre ja auch im echten Leben denkbar und würde von einer Smartphone App auch ein weiteres Mal protokolliert.

Und in echt?

So läuft dies auch für jede Begegnung und jeden Datenaustausch zweier Smartphone Apps ab. Mit der Zeit sammeln sich auf einem Handy viele hundert oder tausend Zeitstempel- und Prüfsummen-Einträge an. Gespeicherte Daten, die älter als 21 Tage sind, werden von der Software automatisch wieder gelöscht. Warum wohl? Was denkst Du?

Die gespeicherten Kontakt Daten und Prüfsummen auf dem Papier-Smartphone von Alice könnten nun folgendermassen aussehen:

Coprotrac Papp



«Ich wurde infiziert»

Nun wird eine Person der Spielgruppe positiv auf eine Corona-Infektion getestet. Sie schickt darum ihre App-ID an einen zentralen Server, welcher diesen geheimen Schlüssel wiederum an alle Personen mit einer installierten Coprotrac Papp sendet. Das ist bezüglich des Datenschutzes unproblematisch, denn der geheime Schlüssel ist nur eine zufällige Zeichenkette, welche keinen Rückschluss auf die infizierte Person zulässt.

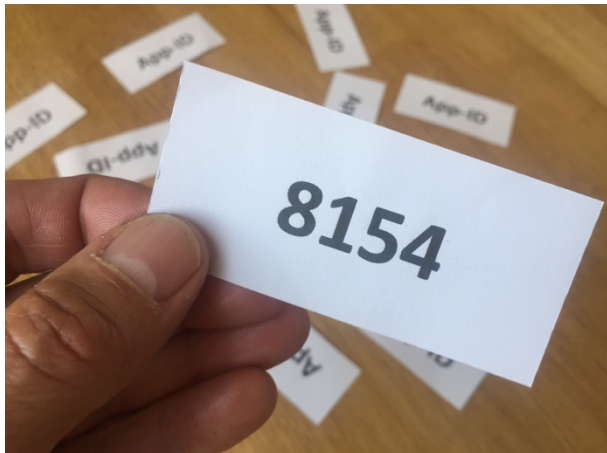
Und in echt?

Damit nur Personen, die tatsächlich positiv getestet wurden, ihren Schlüssel hochladen, könnte der Server zusätzlich ein Passwort verlangen. Infizierte Personen würden dieses zum Beispiel in Form eines anonymen Passwort-Coupons vom medizinischen Personal bekommen.

In unserem Spiel simulieren wir diese Meldung einer Infektion, indem die Spielleiterin oder der Spielleiter eine beliebige App-ID Karte zieht, zum Beispiel «8154».

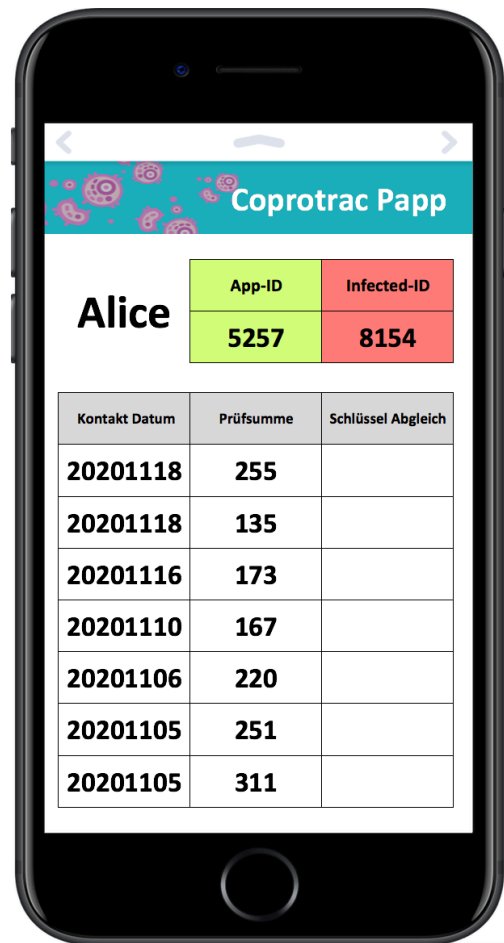
Alle Spieler/innen notieren sich nun die anonyme App-ID in das rote Feld **Infected-ID**. Wenn in unserem Spiel die infizierte Person unerkant bleiben möchte, muss sie die nächsten Schritte (unnötigerweise) auch zum Schein mitmachen.

Coprotrac Papp



Nun muss Alice herausfinden, ob es auf ihrem Smartphone einen Datumsstempel-Prüfsummen-Eintrag gibt, auf den dieser Schlüssel passt.

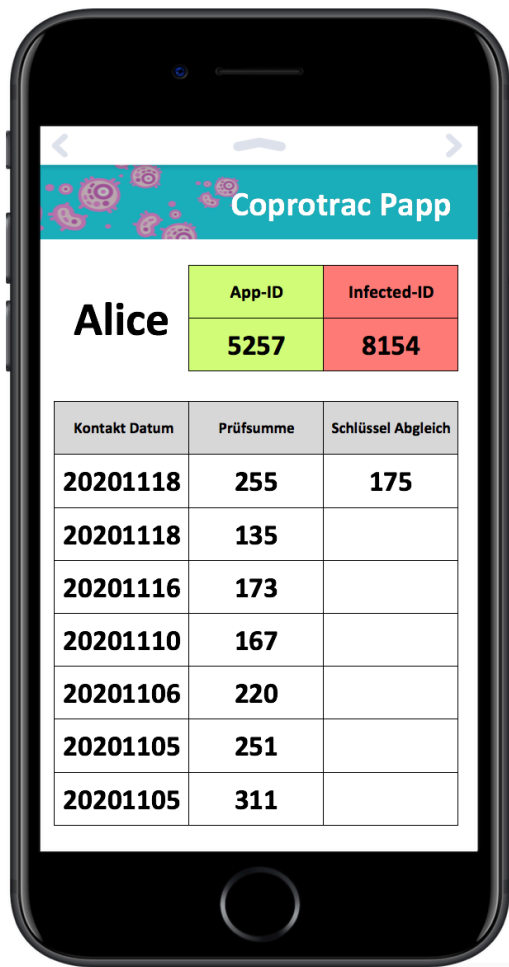
Alice nimmt dazu den Schlüssel «8154» und geht jeden Eintrag im Speicher der Reihe nach durch. Sie generiert mit dem jeweiligen abgespeicherten Kontakt Datum und der heruntergeladenen Infected-ID die Prüfsumme.



Alice berechnet die erste Prüfsumme (Schlüssel Abgleich) folgendermassen:

- Multiplikation der Ziffern der Infected-ID = $8 \times 1 \times 5 \times 4 = 160$
- Quersumme des Kontakt Datums = $2 + 0 + 2 + 0 + 1 + 1 + 1 + 8 = 15$
- Prüfsumme (Schlüssel Abgleich) = $160 + 15 = 175$

Coprotrac Papp

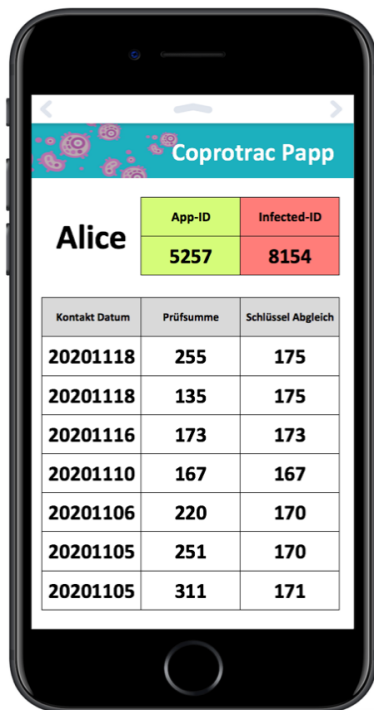


Ist dieser Schlüssel Abgleich identisch mit der zum entsprechenden Datum abgespeicherten Prüfsumme, weiss Alice, dass sie an diesem Datum mit dem Besitzer des heruntergeladenen Schlüssels «8154» Kontakt hatte.

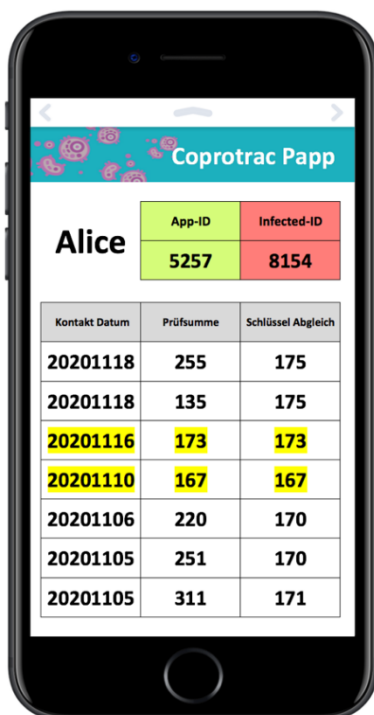
Dies ist im ersten Eintrag unseres Beispiels nicht der Fall ($255 \neq 175$).

Coprotrac Papp

Alice berechnet nun mit der Infected-ID auch die restlichen Prüfsummen in ihrem Speicher.



Alice findet zwei Einträge, auf die der Schlüssel passt. Das bedeutet, dass sie am 10. November und am 16. November 2020 Kontakt zur infizierten Person hatte. Dies würde eventuell auf ein höheres Infektionsrisiko hindeuten.



Coprotrac Papp



Mit Hilfe ihrer Agenda - in unserem Spiel mit den Contact Kärtchen - findet Alice heraus, dass diese Kontakte auf ihrer Zugreise nach Ascona und an der Geburtstagsparty von Alexis stattgefunden haben.

Wer die betreffende Person ist und ob sie sich beim Kontakt angesteckt haben könnte, muss sich Alice selbst überlegen.

Alle Teilnehmer/innen können mit ihrem Schlüssel Abgleich nun feststellen, ob sie in den letzten 21 Tagen Kontakt mit der positiv getesteten Person hatten.

Passt der heruntergeladene Schlüssel auf keinen Eintrag, hatte man keinen Kontakt mit dieser infizierten Person.

Auch in echt

Begegnungen speichern, Infekte melden, Alarm schlagen – ohne dass Namen oder persönliche Daten irgendwo zentral abgelegt werden: Proximity Tracing Apps können einen Lösungsweg aufzeigen. Alles, was es dazu braucht, sind ein gemeinsames Protokoll, ein wenig Kryptographie und ein Server, der ein Minimum an anonymer Information weiterleitet.