Vol.2021-DPS-186 No.8
Vol.2021-CSEC-92 No.8
2021/3/15

Generalization of Contact Tracing

Mathieu de Goyon[1,3,a)] Atsuko Miyaji[1,2,b)] Tian Yangguang[1,c)]

概要：With the Covid-19 pandemic appearance in December 2019, many research works started to develop contact tracing applications to monitor the spread of the virus. Most existing solutions focus on two-party settings: a non-confirmed user interacts with another user (both of them called close contact). They only focus on the case where two persons meet at a place. However, such cases where two or more persons meet occur often. Current system does not deal with such cases. As a result, a simple extension to multiple-party case does not scale well – as the number of close contacts increases, the workload between users becomes a bottleneck. In this paper, we propose a new contact tracing protocol that works in a multi-party setting. We evaluate our scheme from the point of view of communication, computation, transaction complexities and also discusses the risk of transmission depending on the number of people.

キーワード：Contact Tracing, Multi Signature, System Security

1. Introduction

Since the beginning of the pandemic in December 2019, many countries had to take extreme measure, such as lockdowns or curfews, to slow down and attempt to stop the spread of the virus. In order to get out of those measures, it became necessary to monitor the spread of the virus by finding the people who have been in contact with contaminated patients and take appropriate actions. Because it is a very hard task to do completely manually, many researchers opted to develop contact tracing apps, usable on smartphones or tablets to automate this task.

The main function of these contact tracing apps would be to alert its user if someone the app deems as a close contact was later found to have the covid-19. The user would then be able to self-isolate and monitor his symptoms, or even contact the health authorities to get tested. If implemented correctly, this could drastically slow down the spread of the virus. But such an automated process raises some important issues. Firstly, does those contact tracing apps really ensure the privacy of its users. If handled incorrectly, people could find out the identity of contaminated patients, and even who they have been in contact with. Secondly, is an application really able to accurately measure the risk of transmission. Indeed, the only way to contract the COVID-19 is by having direct or indirect contact with someone contaminated. Unfortunately, it is not technically possible to our knowledge to measure the risk of an indirect contact using a smart device such as a smartphone. However, contact tracing apps would still be an important help if they were able to accurately measure the risk during a direct contact. That is to say, being able to alert its user if the contact was enough for there to be a reasonable chance of a COVID-19 transmission.

However, it does not seem to be always the case for some of the apps being used right now. The vast majority of contact tracing apps are currently using Bluetooth Low Energy to communicate with other users. They then use the Received Signal Strength Indicator (RSSI) to calculate the distance between the users and deduce the risk of transmission [1].

Such a protocol is however impractical because Bluetooth is sometimes unable to provide an accurate measure of distance. Indeed, many sources of errors could interfere with a measurement such as the absorption by human bodies, different environmental effects, the orien-

[1] 大阪大学
Osaka University
[2] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
[3] IMT Lille Douai, France
[a)] mathieu@cy2sec.comm.eng.osaka-u.ac.jp
[b)] miyaji@comm.eng.osaka-u.ac.jp
[c)] jack@cy2sec.comm.eng.osaka-u.ac.jp

tation of the antenna, or the specific behaviour of the device. All of these external factors could make a device 2 meters away look like it was 20 meters away [2]. In fact, some experiments show that in specific environment such as tramways, contact tracing is equivalent to picking close contacts randomly [3]. Some recent works have started tackling this problem by using other technologies paired with Bluetooth to more accurately measure the distance [4][5][6].

Another problem is that most current works consider the basic situation of someone meeting someone else. Yet, a meeting with more than 2 participants occur very often. In current solutions on the market, a meeting between several people at the same time is only considered as a repetition of this basic protocol until everyone has interacted with everyone else [7][8].

However, a meeting between more than 2 people is and should be considered as completely different from several meetings between 2 people. Studies show that the number of people is also an important factor of Covid-19 transmission that should be considered during the risk assessment [9]. This point will be further discussed in section VII.

Our contribution – To overcome these problems, we introduce a new multi-party setting protocol which can be used no matter the number of people. Instead of considering a meeting between n people as a repetition of n(n-1) meetings between 2 people, our protocol consider the meeting as a whole, saving the number of people as a parameter that can be used to assess the risk. Furthermore, all participants will generate proof of the meeting by using a multi signature scheme, the Pixel multi-signature scheme which was introduced last year by Drijvers [10]. We also discuss the number of computations and communications of our protocol during a meeting between several people and compare it to an existing scheme.

In this paper, we introduce previous works in contact tracing apps in Section II and define the system model in Section III. We propose our multi-party setting protocol with 3 different approaches in section IV and present a security analysis in section V. We compare our protocol to an existing scheme in section VI and justify our work in section VII. Finally, we conclude our work in section VIII.

2. Related Works

Trace Together was the 1st centralized Bluetooth based solution, it was deployed in Singapore in March 2020. DP3T (Decentralized Privacy Preserving Proximity Tracing) [11], PACT-East (Private Automated Contact Tracing) and PACT-West are some of the main decentralized systems that were released in April 2020.

In April 2020, Google and Apple jointly released the Google Apple Exposure Notification (GAEN) protocol which is a decentralized Bluetooth based protocol. In June, Japan released their own App, COCOA based on the GAEN Protocol.

In May 2020, Vaudenay released a paper discussing the benefits and demerits of centralized and decentralized apps[12], while proposing some possible future directions.

In June 2020, Liu released a proposition of a centralized Privacy Preserving Protocol using a Zero Knowledge Proof [13]. This app allows the identity of the close contacts of the patient to be hidden from everyone including the government, despite using a centralized approach.

The French Government released a new version of their app TousAntiCovid, previously named StopCovid, using a centralized approach. They consider people as close contact when 2 people have been within 1 meter for more than 5 minutes, or within 2 meters for more than 15 minutes.

The first solution to pair Bluetooth with Ultrasound to measure the distance more accurately, called Novid, was released in April 2020 by Loh [4]. Another similar solution using both Bluetooth and Ultrasound was released in June 2020 by Luo [5]. This app shows an improved accuracy compared to Bluetooth based ones, notably in a situation across a wall which is a situation where Bluetooth based solutions often have false positive results.

3. System Model

3.1 Entities

In the rest of the paper, we consider the following entities :

- User : A person using the contact tracing app on his smartphone. Since we are discussing a multi-party setting, we will consider n users being close enough and for long enough to be considered as close contacts by the app, n being a positive integer. Can also directly refer to the app itself.

- Government : It is the entity responsible of the users registrations with their app.

- Board : Used to publish public information. Can be accessed by anyone.

3.2 System Assumptions

We consider the following reasonable assumptions, to guarantee the functioning of the system :

a) Every user we consider has Bluetooth connectivity on his smartphone, and the app also has access to it.

b) Every user has Internet connectivity on his smartphone, and the app also has access to it.

c) After first downloading the app, the user accepts that the app will upload his information on the Board if he is tested as Covid-19 positive.

d) Every user will not willingly reveal their activities, including on social medias. They will also not share their secret keys.

e) The user has access to every information stored and communicated through the app, but he will not modify it.

f) We assume all adversaries are Honest-but-Curious. That is, they are following the defined protocols but they are curious to learn more than the allowed information.

g) We only consider cryptographic attack in our threat model.

3.3 Threat Model

Weak Contact Privacy – Here, we consider someone that would try to find out the identity of a close contact of a covid-19 patient. A contact tracing app has weak contact privacy if nobody except trusted entities (Government and Health authorities), can find out the identity of close contacts of a patient.

Strong Contact Privacy – In a similar way, a contact tracing app has strong contact Privacy if nobody, including trusted entities (Government and Health authorities), can find out the identity of close contacts of a patient.

Weak Patient Privacy – This time, we consider someone, that would try to find the identity of a patient that has been tested positive for the covid-19. A contact tracing app has weak patient privacy if an outsider cannot find the identity of a patient.

Strong Patient Privacy – Similarly, we consider someone that would want to find out the identity of a patient. A contact tracing app has strong patient privacy if an outsider cannot find the identity of a patient and a close contact has no better chance that randomly guessing. If we consider N the number of close contacts the adversary had that day, an app has strong patient privacy if the probability of the adversary finding out the identity of the patient P satisfies the following inequality : $P \leq 1/N$. We consider it is necessary for the Government and Health authorities to know the identity of the patient.

4. The Proposed Protocol

4.1 Overview

Our protocol is composed of 4 different phases.

In the **Registration** Phase, the Government will generate the different parameters and make them public. Every user will register on the government website via the app and generate the different public parameters.

In the **Meeting** phase, we will consider a meeting between n users staying close enough for every app to consider the others as close contacts, with n $\geq$ 2. Those users stay together for a long enough period of time that there may be a risk of transmission between users. The specific duration and distance will be discussed later in Section VI.

In the **Alerting** Phase, one of the n users to have been in contact is found to have the Covid-19. We will discuss 3 possible approaches for the patient's close contacts to be alerted that they have been in contact with the Covid-19. We will discuss the advantages and disadvantages of those approaches.

In the **Tracing** Phase, the user will be alerted that he has been in contact with a Covid-19 patient. The way he will be alerted depends on the approach used during the meeting phase.

4.2 Registration Phase

The Government will first generate the parameters. For the rest of the protocol, we will consider λ a security parameter, with $\lambda \in \mathbb{N}$. Let G_1, G_2 and G_T be cyclic group or prime order q such that q is a λ prime. Let g_1, g_2 be generators of G_1, G_2 respectively.

Let e : $G_1 \times G_2 \rightarrow G_T$ be a bilinear pairing.

Let H : $\{0,1\}^* \rightarrow \mathbb{Z}_q$ be a cryptographic hash function. Let F be a function taking a message as parameter and outputting an element of G_1. Let h be an integer. The Government also generates his parameters pk_G, sk_G its public and secret key respectively. The Government will then publish the public parameters { λ, H, F, e, h, g_1, g_2, q, pk_G}.

Every user i will register on the government website and receive an identifier Id and a signature σ. He will then generate his parameters pk_i, sk_i, as $y_i = g_2^{x_i}$. Those parameters will be updated each day and kept for 14 days (period of risk of transmission). The user will upload his public key on the government website.

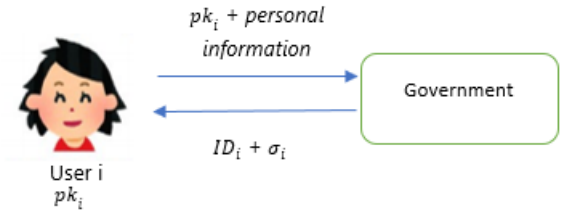

図 1 Representation of the registration phase

4.3 Meeting Phase

In the meeting phase, we generalize the 1-1 setting that is described in most protocols to a multi-party setting. We consider a meeting between n users with $n \geq 2$. Every user is close enough to be able exchange information using Bluetooth. Every app is broadcasting a package to the surrounding indicating their identifier Id and their signature σ of the following form : H(pk, σ, Id). All of the users have been together for long enough that they are recognized as close contacts. In the rest of this phase, we will refer the users by their numbers i,j with $i, j \in \mathbb{N}$.

First, the users will need to verify the identity of the others before recognizing them as close contacts, in a similar fashion to the protocol described by Liu [13] in their Verification phase.

Verification - For $1 \leq i \leq n$; $1 \leq j \leq n$; $i \neq j$, User i will generate a random challenge number r_i and send it to user j. User j will then generate a Schnorr signature on this number r_i :

(1) He will randomly generate a number $k_j \in \mathbb{Z}$, which he will also use with the other users during this meeting phase.

(2) He will then compute $t = H(g_1^{k_j}, r_i)$.

(3) Compute $s = k_j - x_j.t$ with x_j being his secret key.

(4) Output the Schnorr signature $\sigma_j^{'} = (s, t)$ for the challenge r_i

User i will simply compute t' $= H(g_1^s.y_j^t, r_i)$ with y_j User j public key and verify that t=t' . If it is, he will store σ_j and id_j. The same protocol will simultaneously happen for user i, with user j generating a challenge number r_j.

Evidence Generation - As stated before, it is important for the app to recognize that all of the users were together during the meeting. They will first compute the signature

$$\sigma_i^{''} = (h^{x_i}.F(M)^{k_i}, g_2^{k_i}) = (\sigma_{i1}^{''}, \sigma_{i2}^{''}) \qquad (1)$$

They will then generate the message M = $\{id_1, \cdots, id_n\}$, which is all of the identifiers of users involved in this meeting and broadcast $\{M, \sigma_i^{''}\}$ to confirm all of the participants on the meeting. They will then generate a multi-signature Σ on the message M as evidence of the meeting. This multi-signature Σ will be equal for every user, and will be computed as follow :

$$\begin{aligned}\Sigma &= (h^{x_1+\ldots+x_n}.F(M)^{k_1+\ldots+k_n}, g_2^{k_1+\ldots+k_n}) \\ &= (\sigma_{11}^{''}.....\sigma_{n1}^{''}, \sigma_{12}^{''}.....\sigma_{n2}^{''})\end{aligned}$$

With x_i the secret key of user i, k_i random number generated during the verification, F and h public parameters generated by the government and g_2 generator of G_2. In order to verify this signature, the user simply needs to use the bilinear properties of e. The user will first generate Agg, the aggregation of every user' s public key involved in the meeting. Agg = y_1. ... .y_n with y_i the public key of user i.

If we consider Σ as being of the form $\Sigma = (\sigma_1, \sigma_2)$, he will then use the bilinear properties of e to verify that :

$$e(\sigma_1, g_2) = e(h, Agg).e(F(M), \sigma_2) \qquad (2)$$

Indeed we have :

$$\begin{aligned}e(\sigma_1, g_2) &= e(h^{x_1+\ldots+x_n}.F(M)^{k_1+\ldots+k_n}, g_2) \\ &= e(h^{x_1+\ldots+x_n}, g_2).e(F(M)^{k_1+\ldots+k_n}, g_2) \\ &= e(h, g_2^{x_1+\ldots+x_n}).e(F(M), g_2^{k_1+\ldots+k_n}) \\ &= e(h, y_1.y_2.....y_n).e(F(M), \sigma_2) \\ &= e(h, Agg).e(F(M), \sigma_2)\end{aligned}$$

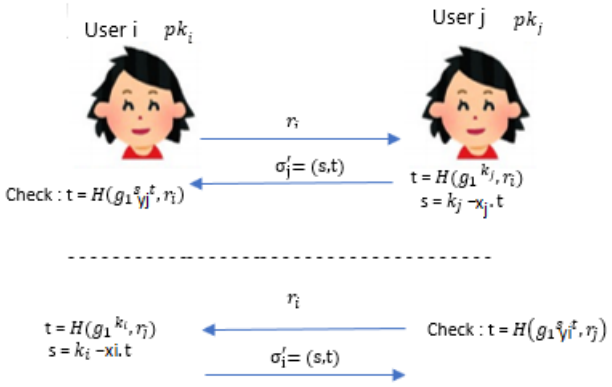

図 2 Representation of the Verification between user i and j. Same protocol happens simultaneously between every users

4.4 Alerting Phase

We consider that one of the participants of the meeting phase has been tested positive for the covid-19. We propose 3 possible approaches :

Decentralized - After the user has been tested positive, he will simply upload the list of all public keys he has used during the past 14 days $P_K = \{pk_1, pk_2, ..., pk_{14}\}$.

Centralized - In the centralized approach, the user will not upload information about himself but about his close contacts. However, it is still possible to guarantee a strong contact privacy even with a decentralized approach. In May 2020, Liu proposed a new privacy preserving app which uses a centralized approach [13]. Their app guarantees a strong contact privacy by using a zero-knowledge proof approach during the alerting phase. That way, the Patient can convince the doctor without disclosing the identity of his close contacts. The doctor will publish the pseudo public keys of the close contacts, generated by the patient, instead of the actual ones.

Alternative - In fact, another possible approach would be to take advantage of the multi-signature instead of using the public keys. Since multi-signature are the same for every participant in a meeting, it can act as proof that a meeting happened. A patient that has been tested positive for the covid-19 could simply upload the multi-signatures on the board without uploading the public keys.

4.5 Tracing Phase

The Tracing Phase mostly depends on the approach used during the Alerting Phase.

In the **Decentralized approach**, the user will need to check every new entry and compare it to the public keys of their close contacts.

In the **Alternative approach** , the user will compare each new entry to the multi signatures on his app.

In the **Centralized Zero Knowledge proof approach**, the user has to test every new entry by using a exponentiation.

5. Security Analysis

In this section, we will present the security analysis of the proposed approaches.

Decentralized approach - Our Decentralized approach guarantees Strong Contact Privacy and Weak Patient Privacy.

The justification for Strong Contact Privacy is rather simple since the only information accessible by a user is the information available on the board : the list of public keys used by the patient in the past 14 days. This information does not leak any information on the close contacts of the patient. It is also the case for the government and health authorities since the alert will be emitted directly by the app after checking the board.

The justification is similar for the weak patient Privacy. The information on the Board will consist of the public keys used previously by the patient. A close contact will easily be able to find the identity of the patient since he has access to the public key inside the app. However, an outsider wouldn't have any way to know the identity of the person linked to this public key.

Alternative approach - Our Alternative approach guarantees Strong Contact Privacy and Weak Patient Privacy if the multi signature scheme is secure.

The reasoning is similar to the one used for the Decentralized approach. This time, the information available to both the government and the close contacts is the multi signature generated by the participant during the meeting. If the multi signature scheme is secure, an outsider is unable to find the identity of the patient, or his close contacts. The Government has access has access to all of the public keys of users of the app, as well as their ids. He also has access to the function F and the integer h. However, the multi signature only gives access to the aggregation of all of the public keys of the participants, so the Government is unable to obtain the identity of the close contacts if the scheme is secure.

The close contacts will be able to know that the patient is one of the close contacts they met when they generated the multi signature. Hence the probability of knowing the identity of the patient is 1/n with n being the number of participants during that meeting. Since we have $n \leq N$, we do not have Strong Patient Privacy most of the times.

Centralized approach - In their proposition of a privacy preserving contact tracing app : a zero-knowledge proof, Liu shows that a zero-knowledge centralized approach has both Strong Contact privacy and Strong Patient Privacy if the signature scheme is unforgeable and if the following assumption holds :

Suppose that $u \in G_1$, $g \in G_2$, $b \in \mathbb{Z}_q$, $Z_0 \in G_T$ *and* $Z_1 = e(u,g)^{b^{q'+2}}$. *When given* $(u, u^b, ..., u^{b^{q'+2}}) \in G_1$, *and* g, $g^b \in G_2$, *no PPT adversary can distinguish* Z_0 *and* Z_1 *with non-negligible probability.*

We would tend to recommend the alternative approach, since it guarantees a stronger patient privacy than the decentralized approach while keeping the strong contact privacy. It is also important to mention that while the centralized approach offers more privacy thanks to the zero-knowledge proof, the protocol during the alerting phase necessitates several computations and verifications that could make it harder to be used. The differences between decentralized and centralized are further explained by Vaudenay [12].

6. Comparison

In this section, we will compare the communication, computation and transaction complexities of our protocol and the protocol used in the zero-knowledge proof approach during the meeting phase and tracing phase. Once again, we do not consider a meeting between 2 people but a meeting between n different people at the same time.

In both protocols, the **verification** in the meeting phase between 2 users consists of 2 messages after the initial broadcast, 2 computations and one verification per user. Since the identity of every other users must be verified, each user will output 2.(n-1) messages, and perform 2.(n-1) computations and (n-1) verifications. This phase is linear and will depend directly on the number of participants n.

In the **zero knowledge proof**, an additional computation and verification is needed between each pair of users for the mutual commitment phase. In our protocol, every user can simply generate the multi signature after having verified the identity of the other users independently from n which reduces the burden on the phone.

Furthermore, the **tracing phase** in the zero-knowledge protocol uses a computation in order to check if the user is a close contact of a patient. The number of computations will directly depend on the number of new entry on the board, which can easily go up to several thousands.

Reducing the communications, computations and verifications to a minimum is especially important in a contact tracing app. Since the App needs to be constantly functioning in the background and communicate information to its surroundings, the phone needs to be constantly on. This can be rather hard on the battery of the phone so limiting the amount of power used by the app will allow the app to run for longer.

7. Justification of Multi-party Setting

7.1 Close Contact

In this section, we will discuss about the signification of "close contact" that is commonly used in contact tracing apps. As explained briefly in the introduction, a close contact should refer to someone that was in contact with the user in such a way that if that contact had the covid-19, there would be a non-negligible risk of direct transmission. In other words, for a contact tracing app to be truly considered as accurate, it should be able to label as close contacts every person susceptible to have been infected directly. To do so, the solutions used by health authorities in most countries use 2 parameters : the distance between the users and the duration of contact. This distance is calculated directly by using the Received signal strength Indicator (RSSI). The most common rule is that someone is considered as being a close contact after being at a distance of less than 2 meters for a period longer than 15 minutes.

Countries	Distance (meters)	Duration (minutes)
France	<1 <2	>5 >15
Korea	<2	>15
GAEN	<2	>15
USA	<1.5 (6 feets)	>15
Japan	<1	>15

図 3 Distance and Duration parameters used in contact tracing apps to determine someone as a close contact

As stated before in the introduction, Bluetooth can be unable to accurately measure a distance in some environments such as a Tramway. However recent research pair Bluetooth with Ultrasound to do it more accurately. However, the important question to ask is : is 2 meters, sometimes even 1, enough for there to be no risk of transmission ? Some studies tend to show that this 2 meter rule comes from very old studies and is not enough to guarantee the virus will not be transmitted. In fact, depending on the situation there could be a transmission with a distance of 4 meters [9]. Concerning the duration of contact, some studies also show that there could be a risk for relatively short contacts. We believe it could be necessary to fix the distance at below 3 meters and the duration at more than 5 minutes in our protocol to further reduce the risk.

7.2 Approach justification

In this section, we will further justify the need to generalize the meeting protocol between 2 users to a multi-party setting.

The primary usage of contact tracing apps is to be alerted if the user has been in contact with someone tested positive for the covid-19 in an automatic way because it is very hard to do manually. But it doesn't mean it

will not be done manually as well in many situations. Be it a coworker, a friend, a family member, there is huge probability those contacts would warn the user directly or indirectly if they have been tested positive for the Covid-19.

In other words, a contact tracing apps is useful because it also works in situations where the user is in contact with someone he doesn't know. Those situations will often occur in public places such as restaurants or trains, and very few will only involve 2 people.

But such a situation is not only the most recurrent it is also the situation which presents the highest risk of transmission. Indeed, studies shows that the number of people involved in a meeting is a direct factor if risk of transmission, with the distance and duration of contact.

That is why it is especially important for the app to know the number of people involved in during a close contact. Our protocol allows the app to have access to this information, while also generating evidence signed by every user involved in the meeting that they were there during the contact. The app could then use that information to calculate the risk of transmission instead of only using the distance and duration of contact.

8. Conclusion

8.1 Conclusion

We proposed a generalization of the protocol of contact tracing apps to a multi party settings. We compared our protocol to another existing protocol. We presented 3 possible approaches used after a patient has been tested positive to the Covid-19. Our protocol is a first step for the app to use the number of people involved in a close contact as a factor to deduce the risk of transmission.

参考文献

[1] Google and Apple: Exposure Notification Bluetooth Specification.
[2] Kumar, S.: Technologie and Public health Perspectives on Private Automated Contact tracing (2020).
[3] Leith, D. J. and Farrell, S.: Measurement-based evaluation of Google/Apple Exposure Notification API for proximity detection in a light-rail tram, *PLOS ONE*, Vol. 15, No. 9, pp. 1–16 (online), DOI: 10.1371/journal.pone.0239943 (2020).
[4] Loh, P.-S.: Flipping the Perspective in Contact Tracing (2020).
[5] Luo, Y., Zhang, C., Zhang, Y., Zuo, C., Xuan, D., Lin, Z., Champion, A. C. and Shroff, N. B.: ACOUSTIC-TURF: Acoustic-based Privacy-Preserving COVID-19 Contact Tracing, *CoRR*, Vol. abs/2006.13362 (online), available from ⟨https://arxiv.org/abs/2006.13362⟩ (2020).
[6] Meklenburg, J., Specter, M. A., Wentz, M., Balakrishnan, H., Chandrakasan, A., Cohn, J., Hatke, G., Ivers, L., Rivest, R. L., Sussman, G. J. and Weitzner, D. J.: SonicPACT: An Ultrasonic Ranging Method for the Private Automated Contact Tracing (PACT) Protocol, *CoRR*, Vol. abs/2012.04770 (online), available from ⟨https://arxiv.org/abs/2012.04770⟩ (2020).
[7] Tang, Q.: Privacy-Preserving Contact Tracing: current solutions and open questions, *IACR Cryptol. ePrint Arch.*, Vol. 2020, p. 426 (online), available from ⟨https://eprint.iacr.org/2020/426⟩ (2020).
[8] Gvili, Y.: Security Analysis of the COVID-19 Contact Tracing Specifications by Apple Inc. and Google Inc, *IACR Cryptol. ePrint Arch.*, Vol. 2020, p. 428 (online), available from ⟨https://eprint.iacr.org/2020/428⟩ (2020).
[9] Jones, N. R., Qureshi, Z. U., Temple, R. J., Larwood, J. P. J., Greenhalgh, T. and Bourouiba, L.: Two metres or one: what is the evidence for physical distancing in covid-19?, Vol. 370 (online), DOI: 10.1136/bmj.m3223 (2020).
[10] Drijvers, M., Gorbunov, S., Neven, G. and Wee, H.: Pixel: Multi-signatures for Consensus, *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020* (Capkun, S. and Roesner, F., eds.), USENIX Association, pp. 2093–2110 (2020).
[11] Vaudenay, S.: Analysis of DP3T, *IACR Cryptol. ePrint Arch.*, Vol. 2020, p. 399 (online), available from ⟨https://eprint.iacr.org/2020/399⟩ (2020).
[12] Vaudenay, S.: Centralized or Decentralized? The Contact Tracing Dilemma, *IACR Cryptol. ePrint Arch.*, Vol. 2020, p. 531 (online), available from ⟨https://eprint.iacr.org/2020/531⟩ (2020).
[13] Liu, J. K., Au, M. H., Yuen, T. H., Zuo, C., Wang, J., Sakzad, A., Luo, X. and Li, L.: Privacy-Preserving COVID-19 Contact Tracing App: A Zero-Knowledge Proof Approach, *IACR Cryptol. ePrint Arch.*, Vol. 2020, p. 528 (online), available from ⟨https://eprint.iacr.org/2020/528⟩ (2020).