Chapter 1
Introduction and Preliminaries

Abstract In this chapter, we will discuss the basic understanding of contact tracing software and the related cryptographic techniques. The underlying models of computation and communication will be explained. A standard smartphone can be considered as the basic device and there will be communication among those devices using Bluetooth technology. The data transfer between the mobile device and the backend server (might be controlled by governmental authorities) will be taken care of by the standard data communication channel provided by the service provider. Social issues related to privacy will also be touched upon in this introductory chapter.

1.1 Introduction

Consider the design of an application such that if you come locationally close to one of your friends then your mobile phone will give you a signal. There could be two different approaches to implementation. In one approach, the location of the phones could be continuously analyzed by some location tracking system. Whenever the system finds two phones in close proximity of each other, it would check whether one phone number is in the contact list of the other. If found, then both the phones will be notified with some characteristic sound, vibration, or a message. What are the problems in this approach?

1. First of all, in a location, there are a large number of mobile phones and therefore such a matching algorithm would possibly have huge costs associated with computation and communication.
2. The complexity of the algorithm would get compounded due to the dynamic and real-time nature of the process.
3. In addition to the computational issues, there is a serious privacy concern. If the application is running in a central server, the server would have the location histories of everyone using such an application.
**Bluetooth-based Solution.** There is another clever option that may not need to depend upon any location tracking systems. In this option, the communication happens through the Bluetooth technology. Hence, the two devices can communicate with each other only when the friends who own the two devices come in close proximity to each other, say, for example, within a distance of 10 m. Since in this approach the phones would use the Bluetooth technology, the devices would not have to depend upon the mobile network services. This is exactly the approach of Bluetooth-based [49] automated contact tracing [32, 34] solutions. If a person is diagnosed SARS-CoV-2 positive, then the people around him/her can be informed about their risk of getting infected as long as all of them have the contact tracing app running on their mobile phones. The protocols that are used in such Bluetooth-based notification systems are referred to as digital contact tracing protocols. Implementing such a system is rather straightforward given the maturity of Bluetooth technology. That is why in the present scenario, many countries and certain private entities have planned to launch such a system on priority. In some of the countries, such systems have already been rolled out. Given the level of urgency due to the ongoing pandemic, and the severe competitive and volatile business environment, the possibility of standardization of such software systems around the world seems elusive. Also, some of the privacy and security issues could be country-specific. Under this scenario, our main motivation is to explain the algorithms of contact tracing protocols, identify the cryptographic primitives used, and analyze the security and architecture of these systems. We present our analysis of the apps and protocols that are already being used in Singapore, Australia, the USA, India, and some of the European countries. Technological giants Google and Apple together have come up with a set of APIs [4] for exposure notification and several researchers across the world have proposed a formidable range of sophisticated cryptographic protocols and frameworks. It is understood that for travel and for many other purposes, some kind of passports related to SARS-CoV-2 could become essential [36] in the coming days. For example, the Government of India has proposed integration of e-pass functionality, which is meant for any interstate travel during the restricted movement phase, within the Aarogya Setu contact tracing app. Users of the app may also be allowed to apply and get approval for the e-passes through Aarogya Setu app’s user interface.

### 1.2 Background

Before we understand or analyze the digital contact tracing systems, let us delve deep into the question of “why is contact tracing needed.”

First, contact tracing is a tool that primarily helps health agencies and state health departments to predict and control the growth trajectory of an infectious disease. The use of such a tool is essential in an outbreak, epidemic, or pandemic situation.¹

---

¹Due to the epidemic spread of SARS-CoV-2, the WHO (World Health Organization) has declared it a pandemic in March 2020.
However, not all infectious diseases can be tracked using contact tracing. For example, dengue is transmitted by mosquitoes that act as carriers of the virus and any contact tracing effort for dengue would involve tracking the movement of infectious mosquitoes—which can be an exciting topic for a sci-fi movie but no agency or state health department would be dreaming of doing that at this point of time. So when we talk of contact tracing, we are talking about contagious diseases that can be transmitted from human to human directly or via a passive surface like fomites (i.e., objects or materials that are likely to carry infection, such as clothes, utensils, and furniture). So, digital contact tracing solutions (along with the manual tracing) based on mobile applications could efficiently achieve epidemic control on a large scale.

Second, contact tracing also helps the epidemiologists to study the transmission mechanism of a contagious disease. Usually, for a new disease like SARS-CoV-2, epidemiologists may refine the standard parameters of the system dynamic models (like SIR or SEIR) \[ 54 \] based on the stages through which the disease evolves while the contagion may go through certain mutations. These models help in predicting the way the number of people within a geographical region, under the epidemiological compartments, like Suspected (S), Exposed (E), Infectious (I), Recovered (R), etc., may vary with time.

Finally, and most importantly, contact tracing can help the general public in many ways. Those who might have come in contact with an infected individual(s) get to know of the exposure events and therefore their risks of infection through this process. In such cases, the health agencies also provide prophylactic and/or preventive care for such individuals. They are monitored on a daily basis for any development of the symptoms. Also, in case there is a need to quarantine infected or exposed (i.e., at risk) individuals, the health agencies and administration provide such measures based on the findings of the contact tracing process.

Thus, there arises the need for a contact tracing mechanism that does this job as precisely and swiftly as possible.

\subsection*{1.2.1 Did Contact Tracing Help in the Past?}

Indeed, there are a number of instances where contact tracing has helped in effectively controlling and limiting the spread of some diseases. For example, \[ 17 \] smallpox was eradicated by contact tracing and not by universal immunization. In the case of Ebola, it has been successfully used for many years. A CDC (Center for Disease Control and Prevention) report \[ 25 \] on 2014 Ebola outbreak response highlights that

\begin{quote}
contact tracing is a key part of the outbreak …it’s been used in each of the previous 20 Ebola outbreaks over the past 40 years to successfully control Ebola.
\end{quote}

Another recent article \[ 31 \] by two physicians Hart and Yarwood, mentions how contact tracing has helped to limit the spread in case of HIV and meningitis infections.
They consider contact tracing and isolation of infected or at-risk individuals as two important assets that may help in limiting the spread of SARS-CoV-2.

1.2.2 How Is Contact Tracing Done Manually?

The manual contact tracing process starts by identifying the initial set of infected individuals (called the index cases) who have been tested positive to the contagion under study and then interviewing each such individual to find out the possible close contacts, who might have contracted the infection while coming in close proximity with an index case. There is no strict definition of close contact or close proximity. We can borrow the definition [31] given by Queensland Health as per the guidelines of Communicable Disease Network, Australia that says

Close contacts are those who have had face-to-face contact with a confirmed case for a period more than 15 minutes, or those who have shared an enclosed space with a confirmed case for more than two hours.

A comparable definition comes from United State’s Center for Disease Control and Prevention which states in their website [18] that

For SARS-CoV-2, a close contact is defined as any individual who was within 6 feet of an infected person for at least 15 minutes starting from 2 days before illness onset (or, for asymptomatic patients, 2 days prior to positive specimen collection) until the time the patient is isolated.

However, there are observed incidents that contradict the above-mentioned understanding that it takes 15 min or more for the infection to get transmitted. For example, it has been reported [5] by a group of virologists that confirmed transmission took place between two asymptomatic individuals who came near each other for a brief period of time in the company cafeteria where one person passed a saltshaker to another.

In the manual contact tracing process, the contacts are monitored for an incubation period to see if any of them develop symptoms and eventually test positive, in which case he/she is isolated, treated and in turn interviewed to identify his/her close contacts or in case any of them does not show any symptom (or test negative), that person is declared to be not at risk. From an algorithmic point of view, this entire process is repeated for each index case. These data points of primary, secondary, or tertiary contacts starting with the index cases can also be used for analyzing the transmission dynamics, infection clusters, and different types of searches.

1.2.3 Challenges and Issues of Manual Contact Tracing

We briefly mention some challenges and issues which are to be faced in a manual contact tracing process.
1. The main challenge of manual contact tracing is that it is a painstaking process and at the same time resource intensive. It takes time and effort to do a detailed interview of an infected individual and still, it remains error-prone since it depends on how well the infected individual may remember the necessary information like the route map (places where he/she went in the last 14 days, in case of SARS-CoV-2) and the people who came in close contact with him/her in that route.

2. Privacy and confidentiality: Every individual has a right to the confidentiality of his/her medical condition. Hence, at the time of alerting an individual about the possible exposure that might have happened, the health practitioner is not expected to disclose the identity of the infected individual(s).

3. Ethics and legalities: The health agency or department has an ethical and (in most countries) legal obligation to share the information about the risk factor to the individuals who might have contracted the contagion from some other infectious individual.

4. Social stigma: Sometimes, the contacts may feel discouraged to share all information in the fear of stigma, discrimination, abuse, or even ostracization within the society.

1.2.4 The Transmission Mechanism

For analyzing digital contact tracing systems, it is useful to have some knowledge of the transmission mechanism of the SARS-CoV-2 virus. We present here a brief summary of our current understanding. Since SARS-CoV-2 is a newly discovered virus, the understanding about the mechanism of its transmission is expected to evolve over a period of time. As per the current understanding [58] of the World Health Organization (WHO), the virus can get transmitted from symptomatic and pre-symptomatic individuals to others who have come in direct or close contact with the infected individuals. Additionally, the transmission can happen through a passive surface that retains active virus landing on it in the form of respiratory droplets from symptomatic, pre-symptomatic or asymptomatic individuals. The shedding of the virus from the respiratory tract of symptomatic individuals appears to be highest in the first 3 days from the onset of the symptom. On average, a pre-symptomatic individual starts showing symptoms of SARS-CoV-2 after 5–6 days and in some cases as late as 14 days. A pre-symptomatic person may test positive to SARS-CoV-2, 1–3 days before the symptoms of the disease appear. The maximum amount of confusion lies in the case of asymptomatic individuals. In one of the latest updates [60], WHO has mentioned the following:

A subset of studies and data shared by some countries on detailed cluster investigations and contact tracing activities have reported that asymptotically infected individuals are much less likely to transmit the virus than those who develop symptoms.
In summary, the following points are so far known about the transmission mechanism:

- The chance of transmission is highest from symptomatic individuals in the first 3 days after the onset of the symptom.
- Symptomatic individuals remain infectious throughout the duration of disease in his/her body.
- Some pre-symptomatic individuals may be infectious on an average 5–6 days or in the worst case 14 days before the symptoms start showing.
- The transmission can also happen through passive surfaces [59] and the virus may remain active on such surfaces for a few hours and up to several days.
- Asymptomatically infected individuals are much less likely to transmit the virus as compared to symptomatic or pre-symptomatic individuals.
- There is also one more category, called oligosymptomatic transmission that has been mentioned in certain research publications (e.g., in [55]), which indicates that for some individuals the symptoms are so mild that they may even be misinterpreted as asymptomatic cases.

We should also remember that the nature or behavior of the virus may change over time. By analyzing virus genomes from several infected persons with SARS-CoV-2, researchers have reportedly [41] identified close to 200 recurrent genetic mutations in the virus. This reflects how it may be adapting and evolving to its human hosts.

### 1.3 Digital Contact Tracing Systems

Although, in the last few years researchers have conceived contact tracing systems via mobile phones (e.g., the published work of Farrahi et al. [27], the research published by Zhang et al. [62] and, that of Altuwaiyan et al. [3]), there was no significant implementation or experimentation done on digital contact tracing systems in any country prior to the SARS-CoV-2 outbreak. In the last few months, top researchers, organizations, and governments across the world have started proposing a number of competing protocols, frameworks, API systems, and apps. Many of these proposals have also envisaged the possibility of how various systems may be used across multiple countries or geographical regions. However, so far, none of the proposals has considered the co-existence and interoperability of multiple solutions and systems.

We briefly touched upon this in Chap. 4.

The primary driving factor behind the digital contact tracing systems is the usage of cutting edge technology to efficiently scale up a time-tested epidemiological method and thereby stop, control or at least reduce the break-neck speed with which SARS-CoV-2 is spreading across the globe and creating havoc in our lives. The central idea is to use an automated method of detecting contact events of two mobile phones coming in proximity with each other and acting as a proxy to the event of two individuals coming in close contact.
1.3 Digital Contact Tracing Systems

1.3.1 Challenges and Issues of Digital Contact Tracing

While the digital contact tracing systems can circumvent the main challenge of manual contact tracing systems of being painstaking and resource intensive, it faces some imminent new challenges as follows.

- What would happen to those individuals who do not have his/her own mobile device?
- Will the digital system support the feature phones or would it be only implemented for smartphones?
- How reliable and accurate will be such alert systems?
- What happens if a significant part of the population does not install and/or enable the system in their devices due to some reason (e.g., suboptimal customer experience)?

Other related issues that are applicable in manual contact tracing systems may continue to impact the digital version as well, like

- privacy and confidentiality,
- ethical and legal issues, and
- social stigma,

as discussed earlier. Thus, an important question is

Will digital contact tracing systems replace the manual contact tracing process?

A short answer to this question is no. With a little more elaboration, we can say that digital contact tracing systems may not replace the manual process completely at least in the short term. To understand the reasons, let us look at what WHO has mentioned as the three basic steps of contact tracing (we copy the whole text from [57] in the following three points).

- Contact identification: Once someone is confirmed as infected with a virus, contacts are identified by asking about the person’s activities and the activities and roles of the people around them since the onset of illness. Contacts can be anyone who has been in contact with an infected person: family members, work colleagues, friends, or healthcare providers.
- Contact listing: All persons considered to have contact with the infected person should be listed as contacts. Efforts should be made to identify every listed contact and to inform them of their contact status, what it means, the actions that will follow, and the importance of receiving early care if they develop symptoms. Contacts should also be provided with information about the prevention of the disease. In some cases, quarantine or isolation is required for high-risk contacts, either at home or in hospital.
- Contact follow-up: Regular follow-up should be conducted with all contacts to monitor for symptoms and test for signs of infection.
Once a health agency confirms that an individual (say $B$) is infected, it should immediately initiate the process of identifying the close contacts by interviewing $B$ and not wait for the apps installed in the mobile devices of those individuals to raise the alarm. Not only would it be a waste of precious time, but there is also a possibility that some of those close contacts (e.g., kids or aged relatives) may not carry separate smartphones. Additionally, in many developing or under-developed countries, the number of mobile devices per family (especially in rural areas) can be limited in number. In majority of those cases, mobile devices may just be feature phones instead of smartphones.

Hence, towards following the basic steps of contact identification, contact listing, and contact follow-up for the contacts who can immediately be identified by the infected person, manual process of contact tracing may remain the most efficient as well as the most effective approach. Another important factor is digital contact tracing systems in most of the countries would be introduced as a choice and not as a mandate. So every individual can voluntarily opt-in or decide to opt-out. The absolute number of people who prefer to opt-out can be significant even under a wide-spread adoption of digital contact tracing apps. In fact, an individual who decides to opt-out may have an expectation that state health agencies would pro-actively reach out to them in case it is found out that he/she came in contact with $B$.

1.3.1.1 Did $B$ Infect $A$ or Was It the Other Way Around?

Let us now briefly revisit the case of $A$ and $B$ where $A$ comes in close contact with $B$ and a few days later $B$ shows symptoms and tests positive to SARS-CoV-2. At this point, $A$ is expected to be alerted through the manual or digital contact tracing process that she may also watch out for possible symptoms and if required test herself (for SARS-CoV-2). At this point, it may appear natural to assume that $A$ has contracted the virus from $B$, but based on the analysis of the previous section we can logically deduce that it could as well be the other way round. For example, the situation might have been that—$A$ (a pre-symptomatic but infected individual) has come in close contact with $B$ (not infected at that point of time) and transmitted the virus during that contact. $B$’s symptoms have appeared earlier than $A$ and that is why it may incorrectly be concluded that $B$ has infected $A$. There are instances in the documentation of some contact tracing systems where this incorrect conclusion has been drawn and hence while using such systems in the future, users may be alerted with incomplete or incorrect information (of course unintentionally).

1.4 The Technical Framework

Necessity is the mother of all inventions and humanity’s desire of getting rid of the current crisis created by the worldwide pandemic, is the mother of all necessities. Hence, the research, technical, and professional communities from various fields
1.4 The Technical Framework

across the world are joining hands together to conceptualize, design, and implement the digital contact tracing systems at a rapid pace. Due to the involvement of renowned experts (especially from the cryptologic community), many of these systems include innovative protocols and security designs. At the same time due to the rapid pace, some design features appear to be less secure, error-prone, or contrary to their design goals. Over a period of time, we believe these inaccuracies will be taken care of. In the next two chapters (Chaps. 2 and 3), we analyze the pros and cons of various designs by comparing these proposals against their intended objectives and goals. Finally, in Chap. 4, we will present a generic proposal and aim to predict the future course of digital contact tracing systems and the evolving role of cryptology. The motivation of our proposal is to outline how a design could be attempted by balancing different types of (and sometimes contradicting) requirements.

Any innovation happens at the intersection of three factors:

- desirability,
- viability, and
- feasibility.

We have already discussed at length the need for digital contact tracing systems—the desirability factor is therefore addressed. Viability does not appear to be a challenge for this system as every Government and the Ministry of Health of every country is talking about its urgent need and would eventually promote and champion the usage of such systems. So what remains is to explore the factor of technical feasibility.

1.4.1 Technical Feasibility

The number of smartphone users and the number of mobile phone users in the world are estimated to be 3.5 billion and 4.8 billion respectively. Hence, there are about 3 billion people who do not carry any type of mobile phone. The implication of this is as follows.

- There is no technology at this point that can enable digital contact tracing system (DCTS) for everybody in the world. However, in many developed countries, the number of people using smartphones can be quite high, and hence in such countries, a successful roll-out of DCTS may prove to be extremely effective in controlling the spread of the disease. For example, in Singapore, smartphone penetration is estimated [37] to be around 78%.
- Smartphone app-based digital contact tracing systems can be implemented quickly to achieve most of the objectives of contact tracing as outlined by WHO.
- Feature phone-based systems would have technical limitations and can’t be as seamless as app-based systems. However, it is possible to envisage such systems.
- Realistically, Governments and Health Departments have to judiciously use a hybrid of manual and digital contact tracing systems to get the best possible outcome in the current situation.
• Feasibility of detecting transmission through passive surfaces can and must be explored after doing rapid prototyping and experimentation.

1.4.2 System Goals and Objectives

There can be some broad level system goals for such a system, including but not limited to-

• Achieving contact tracing objectives as per guidelines from WHO,
• Prompt notification of alert,
• Epidemiological analysis, prediction, and control,
• Robustness, reliability, and endurance,
• Precision and accuracy,
• System integrity,
• Confidentiality, security, and privacy,
• Interoperability (across regions or across device types or across implementation systems),
• Extensibility,
• Re-usability (in future scenarios), and
• Upward compatibility.

At present, some of the proposals are touching upon many of these goals. However, none of the proposals comprehensively addresses all of these. This is quite natural given the short span and urgency of the designs. It would be desirable that in the near future every system explicitly states their exact positions with respect to all these goals.

In almost all the systems, explicit focus has been given to privacy and security goals because, in many countries, the law is quite strict with respect to confidentiality, security, and privacy.

1.4.3 System Architecture: Options and Trade-Offs

The basic use case involves two individuals (A and B) and their mobile devices (say A and B respectively) acting as their proxies. When the two individuals come in close contact with each other the system must ensure that A records the presence of B and vice versa. From a theoretical perspective, it could be possible to envisage a system where it is sufficient to have only one device (say A) recording the presence of the other device (here B). However, none of the launched or proposed systems falls in this category—hence we would not analyze such solutions. At this point of time, there is a general agreement in the technical and research community that the most suitable option to achieve this functionality is to use the Bluetooth Low Energy (BLE) protocol [14]. BLE, earlier referred as Bluetooth Smart, is a wireless personal area network technology [12, 14], that has been designed and marketed by Bluetooth
Special Interest Group (SIG) for creating applications around fitness, smart homes, security, proximity, etc. It is different from the classic Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) protocol but both use the same 2.4 GHz frequency range and can co-exist in the same device. A variety of mobile platforms (iOS, Android, etc.) and generic operating systems (macOS, Windows 10, etc.) provide native support for BLE. BLE protocol [50] can operate in two different topologies—broadcast topology and connection topology. In broadcast topology, devices may advertise packets (as broadcaster) with a payload of 31 bytes to other devices that are repeatedly scanning (as receivers) to receive advertised packets in a non-connectable mode. There are no inherent provisions of security or privacy in broadcast topology and it is a fast and easy to use the technique for a device to send a short packet to a number of peers simultaneously. In case a device needs to send a payload larger than 31 bytes, there is a provision to use scan response technique to send an additional 31-byte secondary payload in broadcast topology. A device may periodically alternate between the broadcaster role and the receiver role. In the connection topology mode, devices can communicate bidirectionally by establishing a connection between a device playing the role of a central (master) and the other playing the role of a peripheral (slave). A device can act as a central and peripheral at the same time where one central may connect to one more peripheral and similarly a peripheral may connect to a number of centrals. In connection topology mode, devices can have a finer grained control on the data being shared and hence is more secure than the broadcast topology mode. The BLE specification ensures interoperability among multiple device types through two profiles—Generic Access Profile (GAP) and Generic Attribute Profile (GATT) that act as the entry points of the higher layer APIs to the BLE protocol stack.

ASSUMPTIONS. Devices using BLE let the other devices know about its presence by broadcasting discovery packets (in certain intervals through one of the approved channels) and scanning/recording the broadcast packets received from the other devices. The intent is to use the information contained in the packets to detect at a later point of time whether A (mobile device of A) came in close proximity to B (mobile device of B who eventually got tested as SARS-CoV-2 positive) for a duration that is sufficient long to transmit the virus from one to the other (it can be either B to A or A to B) assuming one of them was not infected at the time when they met. In this scenario, the assumptions are as follows.

- Both the devices A and B were switched on with Bluetooth enabled or Bluetooth scanning enabled (and in some situations, there is an additional constraint of using the app in the foreground mode).
- Each of the devices has the same app installed. In each device, the app takes care of broadcasting and recording the discovery packets sent and received via BLE interface.
- The BLE protocol correctly captures the close proximity situation and devices do not capture packets from other devices that are either too far off or have stayed close only for a brief period of time. Here, we must highlight that certain aspects would not be detectable like, whether A and B were wearing masks at the time
when they met or whether an infected person was present some time back and touched a surface through which $A$ or $B$ might have got exposed to the contagion.

Central Server. Since the detection of the event that “$B$ is positive” happens at a point of time much later than when $A$ and $B$ met, the question comes as to how device $A$ would come to know of it. It is unlikely that at the point of time when this information becomes known, $A$ would be in close range of $B$—hence alerting mechanism can’t be built using BLE. If somehow $B$ stores the phone number of $A$ (or at least one of the phone numbers in case of dual-SIM phones), $B$ may be able to intimate $A$. However, that might not be appropriate from the security and privacy perspective. Ideally, $B$ should not learn any extra information (like identity, phone number, IMEI number, etc.) about $A$ or her phone, other than storing some packet values from $A$. The same expectation holds for packets received by $A$ from $B$. Hence, to solve this deadlock, we may introduce a (trusted or semi-trusted) third entity (called $C$ as the server that includes a database) which would facilitate the process that leads to alerting $A$ about the fact that she came in close contact with an infected individual $B$ some days back and hence she must consult a health professional to seek guidance or get checked.

Types of Data Communication in Meeting Phase. In the context of two devices $A$ and $B$ along with the intermediate server $C$—there could be three possible options for alerting $A$:

1. $C$ is aware of the proximity information between two users $A$ and $B$ after $B$ is being reported positive and confirmation about that fact comes from a trusted and authorized source (e.g., health authority). Therefore, $C$ arranges to send an alert to $A$.
2. $C$ receives some data owned by $B$ from other trusted and authorized source after $B$ is being reported positive. $C$ becomes aware of the proximity information between $A$ and $B$ only when $A$ sends a query to check post which $C$ arranges to send an alert to $A$.
3. $C$ is aware of the list of individuals who have been tested positive and/or some identifiers from which their beacon packets can be identified. It periodically sends the information to all the devices. $A$ matches the received information to find out if it came in close contact with any such individual and alerts its owner with the associated risk level (or risk score). There can also be another variation of the third approach where $C$ is used as a post-box by both the parties ($A$ and $B$) to secretly exchange the information about their proximity without revealing the identity of one party (say $A$) to the other parties ($B$ and $C$). The post-box may have centralized or distributed implementation and the secrecy can be maintained via different cryptologic approaches and algorithms.

Among the three possible options mentioned above, the first option can be viewed as a centralized version, the third option as the decentralized version, and the second option as a hybrid of centralized and decentralized versions. Clearly, there are trade-offs for each of the versions and these trade-offs would be discussed in this book when we explain a concrete specification (like DP-3T) that may fall in any of these three architecture options.
IDENTIFIER. It must be noted that at the time of sharing of BLE packets between A and B, the payloads are expected not to reveal the identity of A or B in any way, i.e., from these packets, it would not be possible for either A’s app or B’s app (or any other party with malicious intent) to identify the device that has shared the packet. This aspect is usually achieved by using pseudorandom numbers as pseudonyms for the device which may either be a fixed identifier (not preferred as it would become easy to identify and track the device later) or a rotating identifier that changes in value every 15 min or so. There could also be ancillary data like exact or approximate time of the day, distance between the devices, relative speed, etc., stored along with the pseudonymous constantly rotating identifiers. Many times the expectations of privacy may need to be balanced against social responsibility or ethical considerations (individual’s freedom vs collective good)—for example, if B is given freedom to decide about disclosure of his contact information for the past 14 days (from privacy perspective), there could be many other individuals adversely affected to not have the information about their possible infected states (which in turn would affect the collective well-being) in case B chooses not to disclose.

We would also discuss the adversarial scenarios and how in each of the situations, the possible attacks can be handled. It would also be important to consider certain fault scenarios and their implications on the system goals, e.g., if the BLE packets are somehow missed by any device, the completeness or comprehensiveness goal may get adversely impacted. The integration of the system architecture with the health authority is critical as in most of the proposals, there is an authorization step for B to upload his datasets with the confirmation indicator that he has been tested positive.

Finally, in technical architecture, it is important to point out that the digital contact tracing systems can be built in various layers. These layers are as follows:

- A specification that limits its scope at the communication protocol and data modeling level,
- A set of API libraries that are built on top of a communication protocol and data modeling layer,
- An app which uses proprietary or public-domain or open-source APIs,
- A framework or a system of apps, an interoperability infrastructure and a federated server systems to communicate across agencies and borders.

In each of these types of implementations, the designers may choose to make it open source or provide only a sample implementation without disclosing the backend source code.

1.5 Basics of Cryptology

For maintaining secrecy and privacy, we need cryptologic tools and our main motivation in this book is to have a look from that direction. We discuss a few primitives briefly in this section. For a more detailed introductory technical understanding, one may refer to [47].
1.5.1 Encryption

Encryption is the process of encoding information (plaintext) into a different form (ciphertext), through various means. Cryptographic encryption is implemented by using a function on the plaintext, along with another value, called a key. This encrypted data can then be decrypted using a decryption function on the ciphertext, possibly with the same or a different key. A secure encryption scheme is used to achieve privacy and confidentiality. In other words, an unauthorized user should not be able to get any information about the message from the encrypted message. Moreover, the secret key should be protected by any means, otherwise, the whole system will be completely broken. Cryptographic encryption can be broadly categorized into (i) public key (asymmetric) (ii) private key (symmetric) encryption, and (iii) hybrid encryption.

Public Key Encryption Public key cryptography involves the use of two kinds of keys—a secret key which is not revealed to any entity other than the user to whom it belongs, and a public key which is known to everyone. Any user (sender) who requires to encrypt any information must use the public key of the receiver. The receiver can decrypt the ciphertext using his corresponding secret key to get back the original information. This is possible by mathematically linking the public and secret keys in a certain manner. Examples of public key encryption include the RSA algorithm [43], the ElGamal encryption [26], various elliptic curve-based encryptions [11, 30, 52], etc.

Symmetric Key Encryption Unlike public key, it is assumed in symmetric key encryption processes that the two entities involved in the sharing of information (encryption and decryption) share the same secret key, which is unknown to any other entity. The common secret key is used for encryption as well as decryption algorithms. So, communication in symmetric key encryption is restricted to the authorized users, whereas in public key encryption, information can be sent to anybody having a public key and a secret key. However, symmetric key encryptions are much more efficient and lightweight. AES (a standardized algorithm by standard body of the USA, National Institute of Standards and Technology) is an example of symmetric key encryption which can encrypt 128-bit messages. CBC Encryption or counter mode encryptions are encryption algorithms which can encrypt messages of arbitrary length.

Hybrid Encryption A hybrid scheme first determines a common shared secret key using a key-exchange mechanism (a close variant of a public key encryption). For example, the sender can encrypt session key and send it to the receiver. Using the session key, the users can then communicate through a more efficient symmetric key encryption algorithm. The Diffie-Hellman key-exchange protocol [23] is a popular example which is used in SSH [61] and TLS [21, 22, 42] protocols.
1.5.2 Hash and MAC

Hash Function A cryptographic hash function takes as input, a message of arbitrary length, and returns a fixed-length output. Hash functions are usually very fast algorithms. So, they are also used as a preprocessor of some mechanisms that require the short fingerprint of a message that they produce. A cryptographic hash function has some desirable properties, which make it useful in various cryptographic constructions. These properties include collision resistance (it should be difficult to find two different messages with the same hash values), pre-image resistance or onewayness (it should be difficult to find a message which has a given hash value), etc. SHA2 and SHA3 are commonly used families of hash functions (where SHA stands for Secure Hashing Algorithms [9, 29]). SHA3 is an NIST standard hash algorithm ([9]).

Even though hash functions are public and usually do not take any secret input, one can always keep a part of the input as a secret key. Such hash functions are called keyed hash functions. Keyed hash functions achieve additional security such as unpredictability and pseudorandomness. Keyed hash functions are sometimes also called Message Authentication Codes (MACs).

Message Authentication Code Message Authentication Codes (MACs) are cryptographic constructions used for authenticating messages. These codes take a message and a secret key as their input and give a tag as output (sometimes along with a ciphertext for the message, in which case it is called an authenticated encryption or AE). Like symmetric key encryption, the secret key is shared by two or more authorized users. In addition to authenticating the message (this property of a MAC is also called integrity), a MAC also ensures authenticity of the source. A rightly verified tag also ensures that the message has been sent by one of the authorized users (those users who share the common secret key). MAC constructions may (optionally) use various cryptographic objects such as hash functions, block ciphers (fixed size symmetric encryption), etc. Commonly known as MACs include the Wegman-Carter (WC) MAC [53], CBCMAC [8] and PMAC [10].

1.5.3 Digital Signature

A digital signature is a public key analog of MAC. Messages sent over certain channels may face a risk of being distorted due to reasons like a bad network or a malicious entity. A digital signature is a value computed for each message input using a signing key (which is secret and specific to users). This value is then used to authenticate the sender and check the correctness (or integrity) of the message. Unlike in the case of MACs, verification can be done by anyone who has the verification key corresponding to the signing key. As the public key in symmetric encryption, the verification key is kept public.
Some digital signatures may also store other information about the message such as the origin, time of sending, etc. Digital signatures can also be applied for non-repudiation, by which, a sender who digitally signs a message cannot deny having signed it later.

Digital signatures are usually pre-processed by a secure hash function and implemented using the RSA [43] or other public key cryptosystems. It is easier to implement digital signatures for maintaining the authenticity and integrity of messages since these are very short values as compared to the far longer corresponding messages.

1.5.4 TRNG/PRBG/PRF

TRNG or True Random Number Generator Most cryptographic constructions require the generation of random numbers or pseudorandom numbers for purposes of key generation, nonce generation, salt generation, etc. A true random number generator (TRNG) or randomness extractor works by taking some random sources as inputs and generates truly random outputs of some target size. A random source can be the current time or some other values which keeps on changing. A good cryptographic TRNG must be efficiently computable and statistically close to a truly random distribution.

PRBG or Pseudorandom Number Bit Generator A cryptographic pseudorandom bit generator (PRBG) is a deterministic algorithm, quite similar to a TRNG. However, the input of PRBG is assumed to be true random and short. Moreover, it outputs a (comparatively far longer) string of pseudorandom bits. This output of a PRBG is called a pseudorandom bit sequence. A combination of TRNG and PRBG can produce arbitrary length pseudorandom or random-looking bit string. RC4 [44], SNOW4G [35] (used in 4G mobile encryption), ZUC [33] (Chinese standard) are some examples of PRBGs.

PRF or Pseudorandom Function Usually, pseudorandom bit strings are generated in a sequential manner. Even though these algorithms are fast, we cannot randomly jump to the bits that are located far away. Pseudorandom function (PRF) are some examples of PRBG which can jump to a random position almost at no cost. Moreover, the bit strings can be generated in parallel which makes this primitive very useful in practice. In addition to the secret key, it also takes a position number as an input. It outputs a short random-looking bit sequence corresponding to the position. It is desirable that there is no correlation among the outputs computed for different positions. Pseudorandom function is also used to design key derivation function. Keyed hash functions and CMAC (NIST recommended pseudorandom function) are some examples of PRFs.
1.6 Contact Tracing Protocols

In this section, we present brief descriptions of some of the recent proposals and implemented solutions of digital contact tracing systems. We detail out these proposals and solutions in later chapters along with the analysis from a cryptographic point of view.

1.6.1 Entities and Modules

We now list down certain entities that are used in digital contact tracing systems. For a given system, not all entities may be applicable. Also, there could be certain overlap of features among the entities in a system and functionalities of multiple entities could be integrated into a single entity as well.

1. **User.** Any individual willing to participate in an automated contact tracing process having a device enabled with the interface required by the automated system (e.g., BLE, Bluetooth, WiFi, GPS, etc.) and its associated protocol stacks for communication can become a user of the system. To use the system, a user must install the app (or application), register, and start using the app in the recommended way as governed by digital contact tracing system.

2. **User-App.** This is the app that acts as the user interface to the digital contact tracing system. Depending on the design of the DCTS, it may perform different functionalities—including, but not limited to, the user registration, generation/downloading of proximity identifiers or downloading of keys from which such identifiers can be generated, sending and receiving packets to/from nearby devices running the same app, uploading of identifiers to some central or distributed server(s) and identification and intimation of the user’s risk level (or risk-score) of getting infected with the help of other server(s). The terms like user and user-app (or app) may be used interchangeably.

3. **Central Server.** As described in Sect. 1.4.3, the central server (C) acts as the intermediate server between the devices A and B running the DCTS app and coming in proximity to each other. At the minimum, it is responsible for handling the registration, authentication, and deregistration of users. This is expected to be the case for a fully decentralized system. In a fully centralized DCTS, it may perform the functionalities of generating the pseudonymous identifiers to be used by the devices, hold the list of identifiers (sent/received) corresponding to the users who have been diagnosed positive and identify the risk levels (or risk-scores) of every user by referring to their (sent/received) proximity identifiers and matching those with the (sent/received) identifiers uploaded by the infected users’ apps. At the implementation level, it may consist of multiple cloud-based servers and the administrative jurisdiction of these servers can be dependent upon their physical locations, associated authorities, and the laws of the land. There may also be a group of federated central servers that would manage the automated
contact tracing process in different countries or states or geographical regions while inter-operating among themselves to ensure a seamless movement of users across different regions.

4. **Distributed Servers.** Decentralized systems can use one or more distributed servers as intermediate server(s) between the two devices (A and B). The implementation of such Distributed Servers may be accomplished in the form of a public Bulletin Board Server, Distributed Hash Tables, Blockchain-Based Servers, etc. There should be a mechanism by which distributed servers and the central server (responsible for the user registration) communicate with each other to ensure the authentication of users at the time of accessing the Distributed Servers. The presence of Distributed Servers ensures privacy and security of sensitive proximity data and information shared by the users who have been diagnosed positive, in a way that no party (including the central administrative authority) may derive any useful information other than a user who needs to know his/her risk level (or risk-score) of being infected.

5. **Central Administrative Authority.** The main task of the Central Administrative Authority in a contact tracing system is the administration and management of the entire process, including the delegation of any administrative task to other authorities under its supervision. It is expected to have legal and administrative jurisdiction of central server and any other server used in the system. It may also fulfill other purposes such as estimating the spread of the disease, deciding upon the plan of actions in various scenarios of the spread, etc., or delegate such types of tasks to separate Epidemiological Authority.

6. **Health Authority.** A trusted Health Authority (HA), with its associated server and database, is required by the Central Administrative Authority as well as the apps to determine which users are infected. The HA should therefore ideally involve medical personnel and other health department officials. The responsibility of testing individuals would be the duty of such an authority. Depending upon the design of the system, either HA or the infected user (based on HA’s validation) may upload the sent/received proximity identifiers to the Central/Distributed Server(s). The system should have checks and balances to prevent anyone from bypassing (or forging) the validation step that is mandatorily required to be done with the help of HA.

From the perspective of functionalities or phases, a digital contact tracing system may be divided into the following modules.

1. **Initial setup.** The roll-out of any automated contact tracing system starts with the initial setup phase. In this phase, the non-user entities like the central administrative authority, the health authority, the central server (including a system of federated central servers), the distributed servers (if present), etc., should establish the required symmetric and/or asymmetric keys for the purposes of exchanging information among themselves in an authenticated, private, and secured manner. A part of setup may also take place at the time of app installation by the users. Algorithms like hash functions, PRFs, PRGs, etc., that are used in various phases of the contact tracing process are also initialized in the setup phase.


2. **Registration and initialization.** An individual interested in joining the system as a user must first install the app (or application) on his/her device and create a valid login-id with the registration phase with which he/she would access the app in future. Once the installation and registration steps are completed, the app would initialize itself by contacting the backend server(s). These might include setting up certain authentication keys and/or encryption/decryption keys, downloading proximity identifiers from the backend server(s) or keys to generate such identifiers, setting up a local database at the device end, gathering some minimal information from the user (like the first part of his/her postal code), user’s preferences (e.g., the look and feel theme of the app), etc., before the app can be effectively used for the purpose of automated contact tracing.

3. **Contact-Broadcast.** Once the registration and initialization phase is done, the user-app starts sending (in most of the cases broadcasting) proximity identifiers over a certain range so that the nearby devices running the app may receive these packets. This range is limited by the strength of the medium of broadcast (which makes BLE signals a very good option for such a medium). Every user-app is also designed to receive such proximity identifiers on a continuous basis if shared by other devices that are nearby. The physical interface through which such communication happens (like Bluetooth signal or Bluetooth scanning mode) must be kept **ON** for the app to run successfully.

4. **Reporting Infected Status to Server(s).** On being tested positive by the Health Authority (HA), a user would receive authentication to upload the relevant information to the server(s). Depending upon the legislative requirement of the concerned country or state, the user may be mandated or have an independent choice to upload the proximity identifiers sent/received in previous CTDays, where the parameter CTDays indicate the number of days for which the user is considered infectious (as per the guideline of the corresponding Health Authority).

5. **Risk-level (or risk-score) computation.** A user may receive exposure details from a server (Central or Distributed) or the user-app may download a list (or a subset) of proximity identifiers of infected users on the device and determine the exposure by matching the downloaded identifiers with those stored locally on the device. The form of identifiers downloaded from the server need not be the same as those stored on the device. Once the exposure details are known, the user-app may compute the risk level (or risk-score) of the user as per some risk-computation algorithm using other associated meta-data (like duration of exposures, distance between the devices as estimated from the signal strengths, etc.). The final result may be displayed to the user as the risk level (low, medium, high, etc.) or a risk-score (e.g., the probability of being infected). In some systems, the entire computation may take place at the backend and the user-app receives the final result and displays the same to the user. For the user, the next step could be getting tested or consulting a health professional or self-isolating for a certain number of days or do nothing depending upon the guideline from the local Health Authority. From the DCTS perspective also, there could be various options like preventing any further access of the user to the system unless he/she uploads a
validated test result or allowing a restricted access to the system or no change of access (again depending upon the action the user is advised to take based on his/her risk level/risk-score).

1.6.2 **Centralized Versus Decentralized**

In this section, we investigate the definitions of the two categories, referred to as centralized and decentralized systems (or protocols).

A natural definition of centralized could be those protocols in which the central server is present. The central server (C) can directly communicate to all the users (or user-apps), whereas two apps (A and B) can communicate with each other only when they are in close contact. All implemented and proposed solutions use a central server and hence by this definition, all existing systems can be categorized as centralized.

**Is it possible to design a system that does not have an intermediate entity like the central server?**

This can be an interesting research question. In fact, Vaudenay has hinted at the possibility of using blockchain-based central database in his analysis [51] of DP-3T (Decentralized Privacy-Preserving Proximity Tracing) [24], a decentralized digital contact tracing protocol. Whenever there is a third entity, there would be a question of trust, security and privacy considerations for the new entity (in addition to the need of trust, security, and privacy for the existing entities coming in close proximity to each other). We would discuss in later chapters that there exists an inherent conflict among many of these considerations.

Here, we consider two different approaches that are used to define the centralized and decentralized protocols. In one approach, we look at the role that the central server plays in the contact broadcast phase of the protocol, while in the other, the categorization is based upon the core challenges that the protocol aims to solve.

1.6.2.1 **Centralized or Decentralized: Nature of Communications of a Protocol**

Currently, the cryptographic community defines the centralized and decentralized categories based on how the proximity identifiers are derived or generated.

1. If the proximity identifiers are known to (or could be derived from a key supplied by) the central server then we call the protocol centralized. The prominent examples are BlueTrace, NTK, and ROBERT.
2. On the other hand, if the proximity identifiers are not known to the central server, then we call the protocol decentralized. This means, for decentralized protocols, the proximity identifiers are generated only by the user-app (or the client-side application run at the user’s device end) and only in the event a user is tested positive, some of these identifiers may become known to the central server (how-
ever, in some protocols even that may not happen). So a decentralized protocol can run the contact broadcast phase without any help from the central server. The examples of decentralized protocols are DP-3T, Apple-Google, PACT (East and West), TCN, Pronto-C2, DESIRE, Epione, protocol proposed by the IST group, etc.

1.6.2.2 Centralized or Decentralized: Nature of Goal of a Protocol

We can also categorize the protocols by looking at the problems that the protocol aims to solve.

Here, we identify two types of goals or problems-

1. In one of the types, a centralized protocol is one in which the central server can trace all the contacts of a positively-diagnosed user as soon as he/she reports the status (of being infected) to the system. The protocols for which this would not be feasible would be categorized as decentralized. In this approach, BlueTrace, Aarogya Setu would be classified as centralized and almost all other systems would fall under the category of decentralized.

2. In another approach, we consider a protocol decentralized if the central server does not have any information about the “close contact” events. Therefore in such systems, the risk level (or risk-score) of a user can not be computed by the central server. However, the central server may have some information (e.g., the proximity identifiers sent in the last CT days) about the users who have been diagnosed positive. Hence, DP-3T, Apple-Google, PACT (East coast and West coast) can be categorized as decentralized protocols whereas BlueTrace, NTK, ROBERT, etc. would fall under the bucket of centralized protocols.

From these discussions, it would be apparent that there could be several other possible types of categorizations as well for defining the centralized and decentralized protocols. One may also create a hybrid of some of the above approaches and define that as a new categorization. For example, we may define a system as decentralized only when the central server does not play any role in generating the proximity identifiers, has no role to identify the “close contact” events, does not compute the risk level (or risk-score) of a user and has no direct knowledge of infected users. In this approach, DESIRE will be classified as a decentralized protocol but DP-3T, Apple-Google, etc., would be viewed as centralized protocols. It must be pointed out that a protocol that is considered decentralized as per its role (at the time of contact broadcast event) can potentially be converted to a decentralized one according to the type of problem it solves, however, a protocol where the proximity identifiers are generated by the central server cannot be converted so easily to the decentralized version.
1.6.3 Security and Privacy Assumptions

Certain reasonable assumptions must be made about the security and integrity of these entities against attacks from a malicious entity or other adversaries. This not only requires notions of trust but also permissions for communication to be built for these entities.

1. The health authority may only provide information about positive users to the server. The health authority is assumed to be a fully trusted entity, as otherwise, the absence of reliable testing and infection data would render the entire exercise pointless.
2. The central server may not necessarily be trusted, and it could prove useful to consider some attacks from a malicious server.
3. Any two user-apps may communicate with each other only if in close contact. Finally, although users cannot be trusted at all, user apps may or may not be trusted.
4. All communications from a user to the server and from the server to the health authority are carried out over a secure channel (e.g., by using TLS protocol). On the other hand, communication between users cannot be considered to take place over a secure channel, which renders it vulnerable to tapping, but it is not tamperable.

**Semi-Honest and Active Model of Adversary.** These assumptions about the powers an adversary may possess over each of these entities leads to the following two distinctions: In a *semi-honest model*, an adversary will follow the protocol honestly, but gain information from this honest transcript, whereas in an *active model*, an adversary has the power to gain extra information through dishonest means. If the semi-honest model is a preference, the contact tracing app must be installed in a TPM (Trusted Platform Module); in case the enforcement of a TPM is not possible, the active model must be allowed. We may have a hybrid of the above two models where some simple computation can be assumed to be done honestly and the rest can be done arbitrarily by an adversary. This could be a reasonable assumption as the implementation of TPM can be costly and the cost could depend on the computation which would be done in TPM.

1.7 Some Examples of Contact Tracing Protocols

In this section, we provide brief descriptions of some digital contact tracing protocols/systems that have already been launched or have been proposed. The detailed descriptions and analyses of the protocols and systems are presented in Chaps. 2 and 3 under the categorization of centralized and decentralized systems respectively. The intent here is to provide the reader with a glimpse of the variety of approaches and not an exhaustive listing of the launched or proposed systems.
1.7 Some Examples of Contact Tracing Protocols

1.7.1 The TraceTogether System (BlueTrace Protocol)

TraceTogether is the first digital contact tracing system (DCTS) that was deployed at a country level for automated contact tracing of SARS-CoV-2 infection. It was developed by Singapore’s Government Technology Agency and was launched on 20 March 2020 to supplement the centralized contact tracing effort of the Ministry of Health. The source code of the app was later released to the open-source community as OpenTrace along with the design specification of the protocol (called BlueTrace) on top of which the app had been developed. The detailed documentation (including a white paper) about the BlueTrace protocol is available in its official website [7, 13].

In this protocol, the proximity identifiers (called the temporary IDs) are generated by the backend server (or the central server). A central database, hosted in the cloud-based secured backend server and administered by the Ministry of Health, keeps track of the permanent ID of every device that is generated at the time of registration against a phone number. When a user of the app (say B) is found to be infected, a contact tracer will reach out and ask him to securely upload the temporary IDs received by his device (say B) in the last 14 days to the central server. In the backend, the received temporary IDs would be analyzed to identify the permanent IDs of the users corresponding to those temporary IDs. The Ministry of Health personnel would next contact the individuals who came in close contact with B, by referring to the phone numbers stored in the central database against their permanent IDs. The usual process that is followed in case of manual contact tracing would then be followed for these individuals.

Based on our discussion from the previous section on the classification of automated contact tracing systems, we can categorize the TraceTogether (as well as OpenTrace and BlueTrace) under the centralized category.

Australian Government has recently launched the COVIDSafe app that has been built on top of the BlueTrace protocol [19]. At the implementation level, there are certain differences between TraceTogether and COVIDSafe. In both the systems, the apps can run seamlessly in the background mode on Android phones. However, the apps can be used in Apple phones successfully only when running in the foreground mode which is not a convenient option for the users. In case of COVIDSafe, any uploading of proximity data to the central database can be done when the user who has been diagnosed positive gives consent to the health authority.

1.7.2 Aarogya Setu

The Government of India has launched Aarogya Setu [1, 2] as “…a digital service, primarily a mobile application, developed by the Government of India and is aimed at protecting the citizens during SARS-CoV-2. It is designed to augment the initiatives of the Government of India by informing the people of their potential risk of SARS-
CoV-2 infection and the best practices to be followed to stay healthy, as well as providing them relevant and curated medical advisories, as per MoHFW and ICMR guidelines, pertaining to the SARS-CoV-2 pandemic.” It has been developed by the National Informatics Center (NIC)

As per the FAQ [2], the salient features of this system are:

- Automatic contact tracing using Bluetooth,
- Self-assessment test based on ICMR guidelines,
- Risk status of user,
- Updates, advisory, and best practices related to SARS-CoV-2,
- Geo location-based SARS-CoV-2 statistics,
- Nationwide SARS-CoV-2 statistics,
- Emergency SARS-CoV-2 helpline contacts,
- List of ICMR approved labs with SARS-CoV-2 testing facilities,
- e-Pass integration, and
- Support for 12 languages.

The risk status of a user is indicated by four different colors where “Green” signifies Low Risk, “Yellow” means Moderate Risk, “Orange” indicates High Risk, while “Red” implies SARS-CoV-2-positive status. The digital signature of every interaction is created by recording the time, the duration, the proximity, and the location. The contact data is stored for 14 days in the device and the risk computation is done at the app end by referring to the list of identifiers sent by the infected users in the last 14 days and cross-checking the local database at the device end for any possible matches.

The app is available for iOS, Android, KaiOS operating systems and it requires Bluetooth as well as GPS protocols to be enabled for its effective functioning. Identifiers get shared through Bluetooth messages between two devices that come in close proximity to each other. The system tries to check whether the user has come in close contact with SARS-CoV-2-infected person by cross-referencing the proximity data against a centralized database of known cases.

### 1.7.3 ROBERT and DESIRE

Researchers from two institutions, namely France’s Inria and Germany’s Fraunhofer, have come up with a digital contact tracing protocol which they have named as ROBust and privacy-presERving proximity tracing protocol (ROBERT). These two institutes are also members of Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) project.

A detailed set of documents including a high-level summary illustrated example and a detailed specification have been shared by the team in Github repository [45]. In a way, ROBERT lies between the centralized and decentralized systems as there are some aspects in which it is similar to a centralized protocol (like BlueTrace), while in some other features, it behaves more like a decentralized protocol. The proximity
identifiers (called EphIDs) in ROBERT are generated by the central server and shared with the devices that eventually use the same in the contact broadcast phase.

When a user is tested positive, with his/her permission and with the authorization of the Health Authority, the entire list of EphIDs that it received from other devices that came in close proximity in the last 14 days—gets uploaded to the central server (in this way, it is similar to BlueTrace protocol). The detection of possible exposure events and the computation of risk scores are done in the central server.

We position this protocol in between centralized and decentralized categories because the main function of identifying potential exposure events is divided into two parts. On one hand, the detection of proximity of an infected user (B) with another (who is not yet known to be infected) user (A) is done at the central server end, while on the other, the discovery of the possible exposure is done by A (the mobile device of A) when it explicitly requests the server for such information.

DESIRE protocol has been proposed as an evolution of the ROBERT protocol by the PRIVATICS project team from Inria, France. It is a more decentralized protocol than ROBERT. In DESIRE, the mobile devices generate the EphIDs independently without the help of the server and store those in such a fashion (using a non-interactive key exchange protocol) that the IDs can be transparently matched at the server end to identify the possible close-contact events without any possibility of recovering the original pseudorandom IDs that got exchanged between the devices when they came in close proximity to each other.

### 1.7.4 East Coast PACT

This is the acronym for Private Automated Contact Tracing (PACT) protocol that has been proposed by the joint group of academicians/researchers/professionals in the US from MIT, ACLU, Boston University, American Civil Liberties Union, Brown University, MIT Lincoln Laboratory, Weizmann Institute, and Thinking Cyber-security. The detailed specification is available at [38] and is licensed under the Creative Commons Attribution 4.0 License.

This is also a decentralized protocol with a lot of overlap in basic features with DP-3T. In fact, the specification has highlighted that there exist a number of similarities with some other proposals as well—including, the West Coast PACT [39], the strawman protocol from Covid-Watch project [32, 46] and the Canetti et al. protocol [15].

One may also refer to some other references in this regard at [16, 28, 40, 48].

### 1.7.5 Apple-Google Exposure Notification Framework

Apple and Google have jointly worked and developed a protocol specification and an API framework [4] for an exposure notification system that can be accessed
by an authorized app in Android and iOS platforms. The specification is hugely
influenced by the DP-3T protocol and the TCN protocol by Covid Watch [20]. The
primary differentiator of this protocol and API framework is that it can be used by
apps running in the background mode in both Android and iOS devices which is
not the case for any other launched systems so far. At this point of time, a number
of countries (including the UK) have developed automated contact tracing systems
using the Apple-Google framework.

1.8 Conclusion

In this chapter, we introduce the basic understanding of contact tracing protocols. Not
only the technical points we also try to touch upon the social issues given the unex-
pected SARS-CoV-2 pandemic. It is quite evident that all the problems can never be
solved by digital and automated contact tracing protocols using smartphones. How-
ever, technology can always help in fighting a virus, at least in its limited capacity.
The success of technology is always connected to social and political will. Thus,
while not sufficient, technology is necessary to fight this virus and that is why so
many protocols are arriving in a short time. We must accept that in this effort we could
not touch upon all the proposals, but the important ones are listed in this chapter. We
also provide a brief overview of related cryptologic primitives. With this background,
we continue with more detailed technical aspects in the following chapters. In the
next chapter, we concentrate mostly on the proposals that are centralized in nature.

References

1. Aarogya Setu App. https://www.mygov.in/aarogya-setu-app/.
2. Aarogya Setu FAQs. Accessed 26 May 2020. https://static.mygov.in/rest/s3fs-public/mygov_
   _159056978751307401.pdf.
3. Altuwaiyan, T., Hadian, M., & Liang, X. (2018). EPIC: Efficient privacy-preserving contact
   tracing for infection detection. In 2018 IEEE International Conference on Communications
   (ICC), Kansas City, MO, 2018 (pp. 1–6). https://ieeexplore.ieee.org/document/8422886.
4. Apple and Google. Privacy-preserving contact tracing. https://www.apple.com/covid19/
   contacttracing.
5. Apuzzo, M., Gebrekidan, S., & Kirkpatrick, D. D. (2020). How the world missed Covid-
   19’s silent spread? Retrieved June 28, 2020. https://timesofindia.indiatimes.com/world/europe/
   how-the-world-missed-covid-19s-silent-spread/articleshow/76673535.cms.
6. Ariel Bogle. COVIDSafe’s effectiveness on iPhone in question as Government releases coro-
   navirus contact tracing app. Retrieved April 26, 2020, https://www.abc.net.au/news/2020-04-
   26/coronavirus-tracing-app-covidsafe-apple-iphone-covid-19/12187448.
7. Bay, J., Kek, J., Tan, A., Hau, C. S., Yongquan, L., Tan, J., & Quy, T. A. BlueTrace: A privacy-
   preserving protocol for community-driven contact tracing across borders. https://bluetrace.io/
   static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf.
8. Bellare, M., Kilian, J., & Rogaway, P. (2000). The security of the cipher block chaining message
   authentication code. Journal of Computer and System Sciences, 61(3), 362–399.
References

9. Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2013). Keccak. In T. Johansson & P. Q. Nguyen (Eds.), Advances in Cryptology - EUROCRYPT 2013 (pp. 313–314). Berlin: Springer.
10. Black, J., & Rogaway, P. (2000). Cbc macs for arbitrary-length messages: The three-key constructions. In Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO ’00 (pp. 197–215). Berlin: Springer.
11. Blake, I., Seroussi, G., & Smart, N. (1999). Elliptic Curves in Cryptography. London Mathematical Society Lecture Note Series. Cambridge: Cambridge University Press.
12. BLE: Overview of COVID-19 contact tracing using BLE. https://blog.google/documents/57/Overview_of_COVID-19_Contact_Tracing_Using_BLE.pdf.
13. BlueTrace Protocol: Privacy-preserving cross-border contact tracing. https://bluetrace.io/.
14. Bluetooth Low Energy. Smart or Version 4.0+ of the Bluetooth specification. bluetooth.com. Archived from the original on 10 March 2017 (Accessed via https://web.archive.org/web/20170310111443/, https://www.bluetooth.com/what-is.bluetooth-technology/how-it-works/low-energy).
15. Canetti, R., Trachtenberg, A., & Varia, M. Anonymous collocation discovery: Harnessing privacy to tame the coronavirus. arXiv:2003.13670.
16. Cho, H., Ippolito, D., & Yu, Y. W. Contact tracing mobile Apps for COVID-19 privacy considerations and related trade-offs. arXiv:2003.11511.
17. Contact Tracing – Wikipedia. https://en.wikipedia.org/wiki/Contact_tracing.
18. Contact Tracing for COVID-19. https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/contact-tracing.html.
19. COVIDSafe app. https://www.health.gov.au/resources/apps-and-tools/covidsafe-app.
20. Covid Watch. https://covid-watch.org/.
21. Dierks, T., & Allen, C. (1999). RFC2246: The TLS Protocol Version 1.0.
22. Dierks, T., & Rescorla, E. (2006). The Transport Layer Security (TLS) protocol version 1.1. RFC 4346. Archived from the original on 2017-12-24.
23. Diffie, W., & Hellman, M. (1976). Anonymous collocation discovery: Harnessing privacy to tame the coronavirus. arXiv:2003.13670.
24. Ebola Report: Tracing Contacts. https://www.cdc.gov/about/ebola/tracing-contacts.html.
25. El Gamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. In Proceedings of CRYPTO 84 on Advances in Cryptology (pp. 10–18). Berlin: Springer.
26. El Gamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. In Proceedings of CRYPTO 84 on Advances in Cryptology (pp. 10–18). Berlin: Springer.
27. Farrahi, K., Emonet, R., & Cebrian, M. (2014). Epidemic contact tracing via communication traces. PLOS ONE. Retrieved May 1, 2014, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0095133.
28. Gvili, Y. Security analysis of the COVID-19 contact tracing specifications by Apple Inc. and Google Inc. https://eprint.iacr.org/2020/428.pdf.
29. Handschuh, H. (2011). Sha-0, sha-1, sha-2 (secure hash algorithm). In H. C. A. van Tilborg, & S. Jajodia (Eds.), Encyclopedia of Cryptography and Security, (2nd Ed., pp. 1190–1193). Springer.
30. Hankerson, D., & Menezes, A. (2011). Elliptic curve cryptography (p. 397). Boston: Springer US.
31. Hart, M. B., & Yarwood, T. (2020) Explainer: What is contact tracing and how does it help limit the coronavirus spread? Retrieved March 27, 2020, https://theconversation.com/explainer-what-is-contact-tracing-and-how-does-it-help-limit-the-coronavirus-spread-134228.
32. Ingle, M., White, T., Becker-Mayer, I., Petrie, J., Szabo, Z., Blank, D., Colligan, J., Hittle, M., Fenwick, R., Nash, O., Nguyen, V., Schwaber, J., Veeraghanta, A., Voloshin, M., Von Arx, S., & Xue, H. (2020) Slowing the spread of infectious diseases using crowdsourced data. Retrieved March 20, 2020, https://www.covid-watch.org/, https://www.covid-watch.org/article#contactTracing.
33. Jing, J., Liu, Z., Zhang, L., & Pan, W. (2010). Efficient pipelined stream cipher ZUC algorithm in FPGAC.
34. Khalid, A., & Shendruk, A. (2020). How bluetooth could unleash the world’s largest experiment in digital contact tracing. Retrieved April 16, 2020, https://qz.com/1838625/how-bluetooth-could-power-a-global-experiment-in-contact-tracing/.
35. Madani, M., Benkhadra, I., Tanougast, C., Chitroub, S., & Siéler, L. (2017). Digital implementation of an improved LTE stream cipher snow-3G based on hyperchaotic PRNG. Security and Communication Networks, 2017, 5746976:1–5746976:15.
36. McDonnell, T. (2020). Is it too soon for a “CoronaPass” immunity app? Retrieved April 16, 2020 https://qz.com/1838764/is-it-too-soon-for-immunity-passports/.
37. Muller, J. (2019). Smartphone penetration as share of population in Singapore 2017–2023. Statista. Retrieved Sep 9, 2019, https://www.statista.com/statistics/625441/smartphone-user-penetration-in-singapore/.
38. PACT protocol specification - version 0.1 (4/8/2020). https://pact.mit.edu/wp-content/uploads/2020/04/The-PACT-protocol-specification-ver-0.1.pdf.
39. PACT: Privacy-sensitive protocols and mechanisms for mobile contact tracing. https://arxiv.org/abs/2004.03544.
40. Privacy-preserving contact tracing - Apple and Google. https://www.apple.com/covid19/contacttracing.
41. PTI London. Nearly 200 genetic mutations identified in SARS-CoV2. Retrieved May 06, 2020, https://www.thehindubusinessline.com/news/science/nearly-200-genetic-mutations-identified-in-sars-cov2/article31520645.ece.
42. Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446.
43. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120–126.
44. Rivest, R. L., & Schuldt, J. C. N. (2016). Spritz—a spongy rc4-like stream cipher and hash function. Cryptology ePrint Archive, Report 2016/856. https://eprint.iacr.org/2016/856.
45. ROBERT: ROBust and privacy-presERving proximity Tracing - Inria, France and Fraunhofer AISEC, Germany. https://github.com/ROBERT-proximity-tracing/documents.
46. Specification and reference implementation of the TCN Protocol for decentralized, privacy-preserving contact tracing. https://github.com/TCNCoalition/TCN.
47. Stinson, D. R. (2002). Cryptography - Theory and practice (2nd ed.). Boca Raton: Chapman & Hall/CRC.
48. Tang, Q. Privacy-preserving contact tracing: Current solutions and open questions. arXiv:2004.06818.
49. Townsend, K. Introduction to bluetooth low energy. Adafruit Learning System. https://cdn-learn.adafruit.com/downloads/pdf/introduction-to-bluetooth-low-energy.pdf.
50. Townsend, K., Cufi, C., Akiba, & Davidson, R. (2014). Getting started with bluetooth low energy. Sebastopol: Oreilly. https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch01.html.
51. Vaudenay, S. Analysis of DP-3T between Scylla and Charybdis. https://eprint.iacr.org/2020/399.pdf.
52. Washington, L. C. (2008). Elliptic Curves: Number Theory and Cryptography, Second Edition (2nd ed.). London: Chapman and Hall (CRC).
53. Wegman, M. N., & Carter, L. (1981). New hash functions and their use in authentication and set equality. Journal of Computer and System Sciences, 22, 265–279.
54. Wiki: Compartmental models in epidemiology. https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology.
55. Wolfel, R. et al. (2020). Virological assessment of hospitalized patients with COVID-2019. Nature, 581, 465–469. https://www.nature.com/articles/s41586-020-2196-x.
56. World Health Organization. https://www.who.int/.
57. World Health Organization. https://www.who.int/news-room/q-a-detail/contact-tracing.
58. World Health Organization. https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200402-sitrep-73-covid-19.pdf?sfvrsn=5ae25bc7_2.
59. World Health Organization. How long does the virus survive on surfaces? https://www.who.int/news-room/q-a-detail/q-a-coronaviruses.
60. World Health Organization. Transmission of COVID-19 by asymptomatic cases. Retrieved June 11, 2020, http://www.emro.who.int/health-topics/corona-virus/transmission-of-covid-19-by-asymptomatic-cases.html.
61. Ylonen, T. (1996). SSH - Secure Login Connections over the Internet (pp. 37–42).
62. Zhang, R., Zhang, J., Zhang, Y., Sun, J., & Yan, G. (2013). Privacy-preserving profile matching for proximity-based mobile social networking. *IEEE Journal on Selected Areas in Communications, 31*(9), 656–668. https://ieeexplore.ieee.org/document/6544548.