A Close Contact Tracing Method Based on Bluetooth Signals Applicable to Ship Environments

Qianfeng Lin[1], and Jooyoung Son[2*]
[1]Department of Computer Engineering, Korea Maritime and Ocean University
Busan, Republic of Korea
[e-mail: linqianfeng@g.kmou.ac.kr]
[2]Division of Marine IT Engineering, Korea Maritime and Ocean University
Busan, Republic of Korea
[e-mail: mmlab@kmou.ac.kr]
*Corresponding author: Jooyoung Son

Received August 17, 2022; revised December 9, 2022; accepted February 20, 2023; published February 28, 2023

Abstract

There are still outbreaks of COVID-19 across the world. Ships increase the risk of worldwide transmission of the virus. Close contact tracing remains as an effective method of reducing the risk of virus transmission. Therefore, close contact tracing in ship environments becomes a research topic. Exposure Notifications API (Application Programming Interface) can be used to determine the encountered location points of close contacts on ships. Location points of close contact are estimated by the encountered location points. Risky areas in ships can be calculated based on the encountered location points. The tracking of close contacts is possible with Bluetooth technology without the Internet. The Bluetooth signal can be used to judge the proximity among detecting devices by using the feature that Bluetooth has a strong signal at close range. This Bluetooth feature makes it possible to trace close contacts in ship environments. In this paper, we propose a method for close contact tracing and showing the risky area in a ship environment by combining beacon and Exposure Notification API using Bluetooth technology. This method does not require an Internet connection for tracing close contacts and can protect the personal information of close contacts.

***Keywords*:** Covid-19, close contacts, Bluetooth technology, ship environments, Exposure Notification API

A preliminary version of this paper was presented at APIC-IST 2022, and was selected as an outstanding paper.

http://doi.org/10.3837/tiis.2023.02.020 **ISSN : 1976-7277**

1. Introduction

Globally, cruise market operations have been adversely affected by the COVID-19 pandemic. Cruise company market operations are gradually recovering. It is the responsibility of cruise operators to ensure that ship passengers should not be exposed to health risks. Ships with shut-off spatial environments facilitate the spread of COVID-19 in a global pandemic. Therefore, close contact tracing is an important research topic. Close contact is defined as a person less than 6 feet away from an infected person for 15 minutes [1]. Due to the fact that the Bluetooth signal has a strong signal at a distance of 6 feet, the Bluetooth signal can be used to judge the proximity between devices. Close contact tracing in ship environments is thus possible.

Table 1. Abbreviations

Nomenclature			
API	Application Programming Interface	BLE	Bluetooth Low Energy
RSSI	Received Signal Strength Indicator	TCTL	Too Close for Too Long
IoT	Internet of Things	PAN	Personal Area Network
TxPower	Transmission Power	GPS	Global Positioning System
WiFi	Wireless Fidelity	RFID	Radio Frequency Identification
GAEN	Google/Apple Exposure Notification	RPs	Reference Points
KDE	Kernel Density Estimate	CCIA	Close Contact Identification Algorithm
UTC	Coordinated Universal Time	ROC	Receiver Operating Characteristic
TPR	True Positive Rate	FPR	False Positive Rate
TP	True Positive	FN	False Negative
FP	False Positive	TN	True Negative
AUC	Area Under Curve	ID	Identification Code

Cruise ships have been linked to the spread of COVID-19 worldwide [2]. The Japanese government ordered a two-week quarantine for Diamond Princess passengers and crew after a ship passenger tested positive for COVID-19 [3]. Because of the narrow ship environment, ship passengers are most susceptible to outbreaks of this virus [4]. The transmission of viruses onboard is facilitated by contact with land-based populations. When there is limited air exchange in enclosed spaces, viruses are more likely to spread [5,6].

A ship's risk management must focus on the health of ship passengers. It is essential to manage risks in order to ensure proper operation. An analysis of occupational accidents on merchant ships is presented in [7]. According to [8], people are the most critical element for preventing accidents on ships. Therefore, gathering and recording threat information in a structured manner is critical for improving the health of ship passengers through risk control measures [9, 10]. The availability of real-time information regarding ship passengers is essential to risk management. Identifying COVID-19-related hazards is the first step for ship passengers' health [11]. A corporation should evaluate potential health hazards associated with the COVID-19 pandemic, crews, and ship passengers. It is important to minimize the risk as much as possible by implementing adequate protection measures. It is crucial to trace close contacts [12]. The method proposed in this paper can be used to trace close contacts in ship environments.

An application of this type uses Bluetooth Low Energy (BLE), which has a short range and can be used to determine whether a person is near an infected person [13]. The Received Signal Strength Indicator (RSSI) is one of the most popular methods for detecting Too Close for Too

Long (TCTL) between two smartphones [14]. The main device in this paper is a smartphone. In order to estimate location points for a smartphone, the RSSI value from beacons is used also.

2. Previous Works

BLE beacons are Internet of Things (IoT) devices that can communicate over short distances. A number of advantages of the device include its low power consumption, miniaturization, and low cost. When used on ships, such as cruise ships, BLE beacons perform optimally when it comes to indoor positioning accuracy [15]. [16] describes the use of machine learning models to evaluate different Bluetooth signal features for proximity detection. Using a risk-aware physical distancing system, [17] proposes a system that reduces the risk of being infected by COVID-19 and similar diseases. A Bluetooth-enabled Personal Area Network (PAN) is analyzed in order to develop a physical distancing problem [18]. Based on Bluetooth technology, [19] examines the effectiveness of close contact tracing smartphone applications for COVID-19. The social distancing alert system uses Bluetooth proximity detection technology to estimate the real-time social distancing status of Android handsets based on the RSSI and Transmission Power (TxPower) [20].

Automatic detection of possible close contacts with infectious persons can be achieved using the smartphone's BLE signals and machine learning algorithm [21]. [22] has developed a social interaction tracking system based on BLE and Global Positioning System (GPS). In addition, [22] develops an algorithm to predict the likelihood of being infected with COVID-19 based on the BLE data collected. Using the Google/Apple Exposure Notification (GAEN) API, [23] describes a measurement study conducted on a commuter bus in Dublin, Ireland. Apple and Google developed the Exposure Notifications API to enable developers to create Android apps that alert users to potential exposure to COVID-19. By combining the beacon and the Exposure Notification API, this paper can trace close contacts in Internet-free ship environments. [24] proposes and applies a methodology for evaluating the functionality, privacy, and security of Android applications using the GAEN API. So beacon and Exposure Notification API are used to trace close contacts in ship environments using Bluetooth technology without the Internet in our proposed method.

Currently, most ships do not have WiFi coverage, and even when installed, it is often limited to specific areas of the ship. Bluetooth beacons, on the other hand, are a cost-effective alternative for enabling indoor positioning on ships. They are inexpensive, easy to install, and do not require cabling. In contrast, WiFi access points require power, a data connection, and backend infrastructure to function as positional beacons, making them more expensive and time-consuming. In order to achieve an accuracy of 1.91 meters with WiFi, it may be necessary to install additional access points [25], which can further increase the cost of the system. In a library room measuring 39 meters by 20 meters, twelve Bluetooth beacons can provide an indoor positioning accuracy of 0.87 meters [26]. Cost, accuracy, and installation time are often major considerations. Therefore, Bluetooth beacons are the most suitable equipment for ship environments.

3. Program Design

Our proposed method does not require an Internet connection to let the smartphone user know if they have been in contact with close contact and thus determine if they are at risk. The Internet-free close contact mobile application is ideal for ship environments. The Exposure

Notifications API, jointly released by Google and Apple, enables determining whether the ship passenger has made close contact with infectious persons based on Bluetooth technology. However, Exposure Notifications API cannot provide the exact location information of the close contact but can only give the result of whether or not the close contact has been near. In fact, the location point can be calculated by the Bluetooth signal. Therefore, we propose the idea of designing a close contact mobile application based on the beacon and Exposure Notification API. Our proposed method enables the identification and tracing of close contact without the Internet. In addition, based on location points and Exposure Notification API, a series of health and safety applications for ship passengers can be implemented. For example, close contact trajectory replay, virus transmission risky area identification, etc.

Fig. 1 shows the interaction process for close contact tracing in a ship environment. The beacon continuously sends Bluetooth signals to the surrounding area for a fixed period. The smartphone can calculate its location points on the ship by receiving Bluetooth signals from the beacon. Exposure Notification API encrypts Bluetooth signals to the surrounding area. The non-close contact's smartphone can identify the Bluetooth signal from the close contact's smartphone. Exposure Notification API can determine whether it has been in close contact based on the RSSI. Exposure Notification API can calculate the result of close or far based on the RSSI value. At this point, non-close contacts can check whether they have been in contact with close contacts through the RSSI.

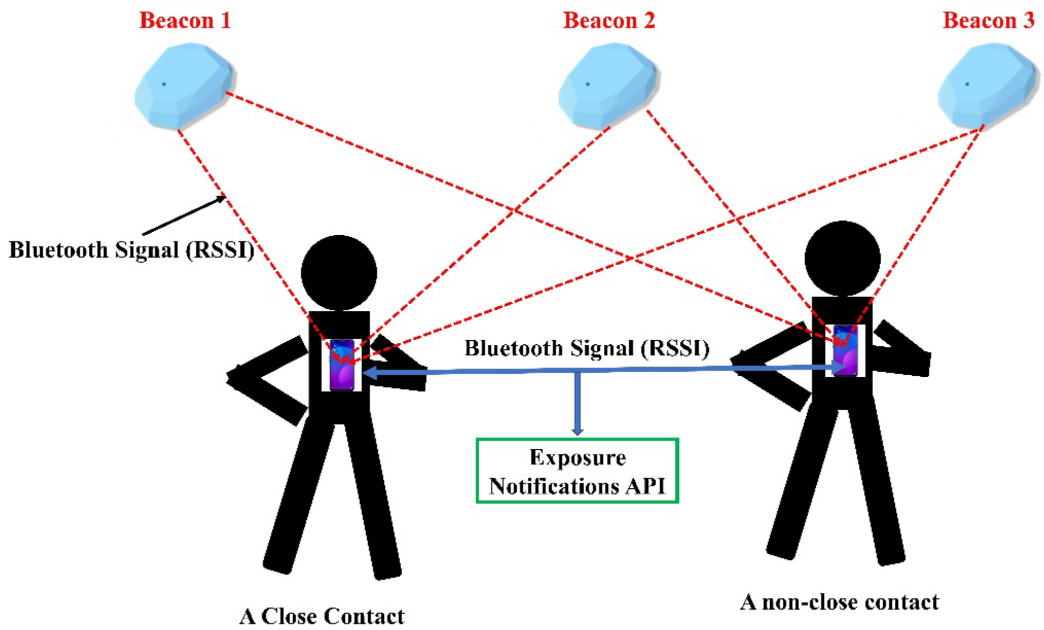


Fig. 1. Close contact tracing diagram

Exposure Notification API can give the encountered time with close contact but cannot provide the location point. The location point of non-close contact can be estimated by RSSI values from a beacon. Our proposed method matches the encountered time with the time information of historical location points of non-close contacts. Thus, close contacts can be tracked in an Internet-free environment. Our proposed method won't expose the personal information and actual location points of close contacts. So our proposed method is suitable for ship environments.

A ship environment-aware indoor positioning algorithm is used in this paper to determine the ship passenger's location points [27]. The algorithm is described in detail in the followings: The relationship between beacons and Reference Points (RPs) must be computed in steps 2 and 3. The coordinates of beacons and reference points can be stored in arrays C and R. Because of the limited coverage of beacons, significant errors often occur near the boundary of the coverage area. In order to solve significant errors, a ship environment-aware indoor positioning algorithm is developed. To solve significant localization errors at the boundary of the coverage area, step 4 is available. List B should be used for storing beacon IDs of boundaries in step 4. As a result of step 5, dictionary D represents the correspondence between each RP and each beacon. In dictionary D, the beacon ID is the key. RP ID represents the value of D in step 5. In step 6, the RSSI value is obtained from the user's device. Step 7 involves selecting peak RSSI values from beacons. In step 8, beacon IDs are sorted in descending order based on peak RSSI values. As a result of step 9, b1 is the top-ranked beacon ID. After that, b2 is the second-ranked beacon ID. The existence of b1 in list B is checked in step 10. The process proceeds to step 11 if b1 exists in list B. The process continues with step 10 if the previous step does not occur. Based on the order of beacon ID, step 11 finds the nearest rp1 in dictionary D. According to the rp1, the user's location can be estimated. The rp2 closest to b2 is selected as a result of step 12. The rp1 and rp2 are calculated as their centroid as P1 in step 13. Step 14 involves identifying the coordinates of b1 in list C as P2. The centroid of P1 and P2 is estimated as the location of the user in step 15. In step 16, return estimated location points. A flow diagram of the ship environment-aware indoor positioning algorithm is shown in **Fig. 2**.

Fig. 3 illustrates the case of close contact 1. Ship passenger 1 and close contact 1 move throughout the ship. As long as the smartphones are within range of a Bluetooth signal, their smartphones can send their respective Bluetooth packets to each smartphone via the Exposure Notifications API. Bluetooth communication must be close enough for the two devices to communicate successfully. Therefore, Bluetooth signals can be used to trace close contacts. Meanwhile, it is possible to estimate the location point of close contact from a known location point.

The definition of encountered location point is a location point at which it is judged to be in contact with close contacts. This study uses encountered location points to estimate the location points of close contacts. **Fig. 3** illustrates how to estimate the location point of a close contact by encountered location points. To establish a Bluetooth connection, other devices must be within the coverage range. If the close contact completes the identity setting on the application, the necessary identification information can be sent via Bluetooth packets as soon as the setting has been completed.

After determining encountered location points, the receiver stores the time information of corresponding encountered location points in its local database. This time information is referred to as the encountered time. In this paper, encountered location points are considered the location point of close contacts.

Algorithm 1 Ship Environment-Aware Indoor Positioning

Input The location list of beacons is called C, the location list of RPs is called R

Output Estimated location points

1: **Begin**
2: The location list of beacons is called C.
3: The location list of RPs is called R.
4: Store the serial numbers of beacons at endpoints in list B.
5: Calculate the Euclidean distance between R and C to find the nearest RP for each beacon and store it as a dictionary D.
6: Obtain RSSI data from beacons on user device.
7: Select the peak value of RSSI data in beacons.
8: Sort the beacons in descending order according to the peak value, and call the sequence as S.
9: Take the first beacon number of the sequence S as b1, take the second beacon number of the sequence S as b2.
10: Determine whether b1 exists in the list B, if it exists, then execute the step 11. If not, execute step 10.
11: Find the nearest rp1 of b1 and the nearest rp2 of b2 in dictionary D.
12: Find the nearest rp1 of b1 and nearest rp2 of b2 in dictionary D.
13: Calculate the centroid of rp1 and rp2 as P1.
14: Get the coordinate information of b1 in list C called P2.
15: Calculate the centroid of P1 and P2 as the estimated location point of the user.
16: Return estimated location points
17: **End**

Fig. 2. A flow diagram of the ship environment-aware indoor positioning [28]

Ship passenger 1 is illustrated in **Fig. 4** with different close contact encountered location points identified. For example, ship passenger 1 wanders around the ship and passes through various areas. During this period, the smartphone of this ship passenger receives Bluetooth signals from multiple devices. Nevertheless, three smartphones belonging to close contacts are eventually identified with the associated identification information. These three smartphones are located in different areas, as shown in **Fig. 4**, but ship passenger 1 is unaware of the exact locations of these three devices. Ship passenger 1 walks on the ship, the Bluetooth signals from their smartphones are received, and the encountered time is stored in the smartphone's local database. Encountered time can be used to calculate how long they were in contact. In this manner, the objective of close contacts tracing is achieved. The smartphones of ship passengers do not need to be connected to the Internet during this process. And the actual location point of close contacts need not be known by other ship passengers.

The Kernel Density Estimate (KDE) is defined as the sum of the kernel functions at each data point [29]. Non-parametric estimation of location points is possible with KDE. In terms of our research topics, this approach is well suited. To calculate the density of location points, this paper uses the KDE method.

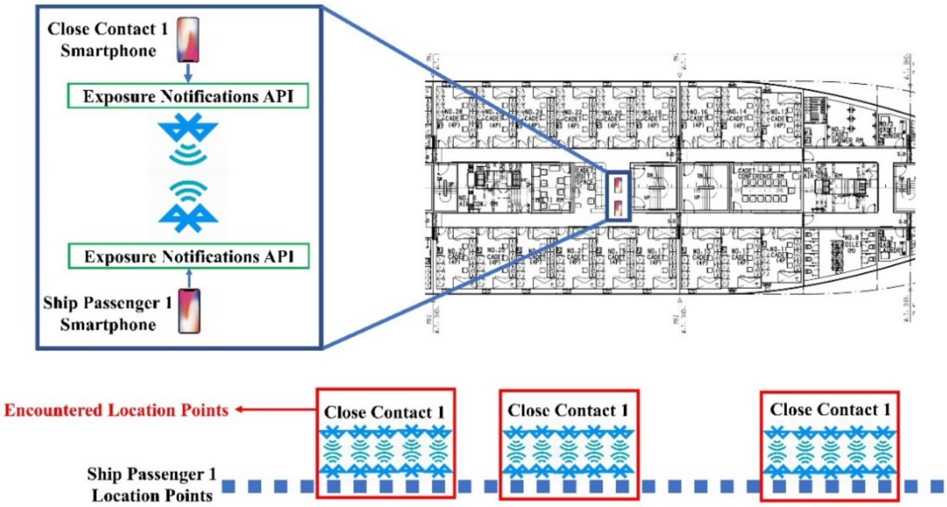


Fig. 3. Identification of encountered location points of close contact 1

The use of α shapes is initially developed in order to generalize a concave hull containing sets of points. Known as the α parameter, this value is such that an edge of a disk of radius 1/α may be drawn between any two edge members of a set of points while still containing all points. Concave hulls, which resemble a rubber band wrapped around pegs at all the data points, correspond to the α shape in which the α parameter is not equal to 0 [30].

The location point of close contacts is needed to cluster for calculating risky areas. This also means that their location points can be clustered by a clustering algorithm. The Close Contact Identification Algorithm (CCIA) is suitable for clustering the location points of close contacts in ship environments. More details about CCIA can be found in [31].

Fig. 5 illustrates risky area identification examples. For the calculation of the risky area, encountered location points are identified. The location points marked in red in **Fig. 5** are the encountered location points. The encountered location points can be fed into CCIA for clustering. The output of CCIA is input to the risky area identification algorithm, namely, the KDE-Alpha algorithm, thereby obtaining risky areas identification results.

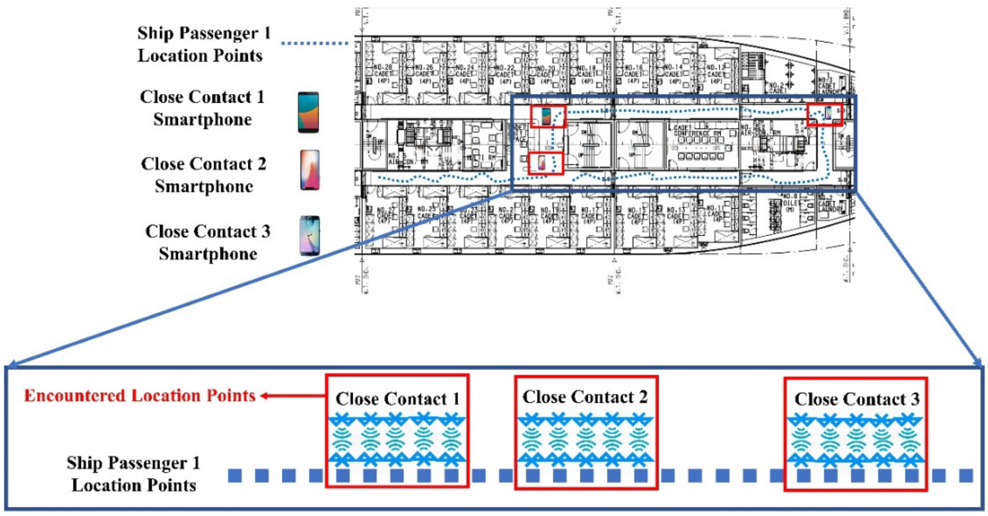


Fig. 4. Identification of encountered location points of different close contacts

Fig. 6 shows the steps of the risky area identification algorithm by calculating the KDE of location points to determine the parameters of α. In the second step, the location data obtained from the user device is converted into location points as S. In the third step, S is input to the KDE algorithm. In the fourth step, the KDE value obtained from the KDE algorithm is used as the α value. In the fifth step, input S and α values to the α shape algorithm. In the sixth step, get the α shape point set from the α shape algorithm. In the seventh step, the area is mapped to each α shape point set. In the eighth step, return the α shape point set.

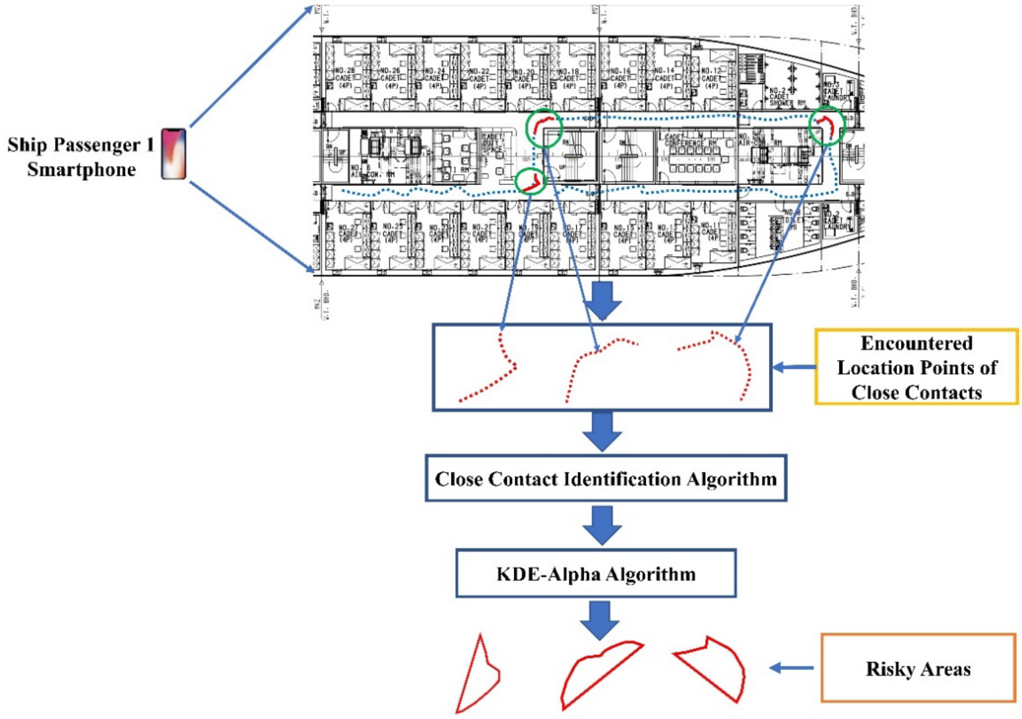


Fig. 5. Risky areas identification

Algorithm 2 KDE-Alpha

Input Estimated location points from each user device
Output The α shape point set

Begin
Estimated location points from each user device is called S.
Input S to KDE algorithm to calculate a KDE value.
The KDE value is set to the α value.
Input S and α values to α shape algorithm.
The α shape points are obtained from the α shape algorithm.
Areas are mapping to each α shape point set.
Return the α shape point set
End

Fig. 6. The workflow of the KDE-Alpha algorithm

Fig. 7 shows the workflow of the Exposure Notification API. Key data is used to generate the Bluetooth broadcast, which maintains a record of the interaction between the two devices. If the ship passenger is a COVID-19 patient or a close contact, they can choose to broadcast Key data to surrounding devices and upload it to the server. The parameter of ReportType is the key parameter responsible for uploading the key to the server. The parameter of TransmissionRiskLevel is the risk level of exposure. This parameter can be set to high or low. The above is the initialization part of the Exposure Notification API. The API function of Day is the date of the outbreak notification, expressed in Coordinated Universal Time (UTC). The API function of DailySummariesConfig provides the risk weights and thresholds, which can be further used for virus propagation risk score calculation. The API function of TotalRiskScore represents the total risk of virus propagation. The API function of ExposureSummaryData means some summary information of exposure notifications. The API function of ScanInstances represents the chronological order of exposure notification scan windows. The API function of Infectiousness represents the infectiousness of the exposure.

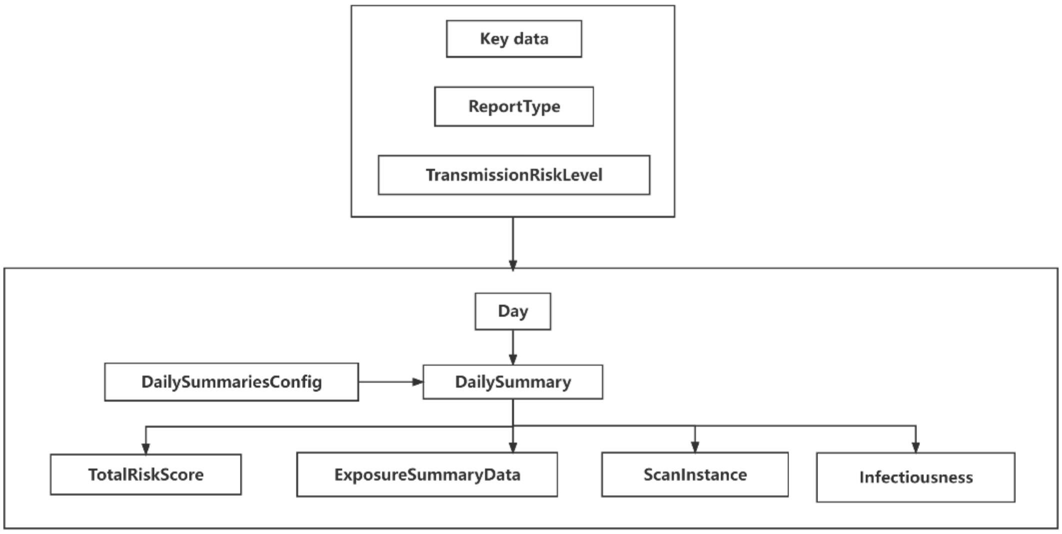


Fig. 7. Exposure Notification API workflow [23]

4. Application Cases

Fig. 8 shows an example of the application of our idea in a ship environment. The red location points represent a close contact, and the green location points represent a non-close contact. The orange boundary represents the area where they met and came into contact. The close contact can know their location points through the beacon, and the non-close contact can also know their location points through the beacon. The Exposure Notification API determines the contact based on the Bluetooth RSSI signal from the close contact's smartphone. The contact result is notified to the non-close contact smartphone. Based on the encountered time and the encountered location points, the location point of close contact can be inferred. This approach does not obtain the identity information of close contacts and can serve to protect personal privacy. At the same time, it can provide valid data for further calculation of virus transmission risk.

Fig. 9 is an example of a close contact tracing app, where close contacts can set up detailed symptom information on the app. Relevant symptom information can be further provided for calculating risk indicators of virus transmission.

When passengers' smartphones can connect to the server. By setting the TotalRiskScore of the Exposure Notification API, the result information can be updated on each smartphone terminal.

Fig. 10 illustrates how a ship passenger can obtain their location points through a Bluetooth signal sent by beacons. A current user location point shown in **Fig. 10** is calculated from the ship environment-aware indoor positioning of **Fig. 2**. Based on the peak value of each RSSI data, the user device identifies which beacon is the closest to the user device. The nearest RP and beacon can be used to estimate user location points.

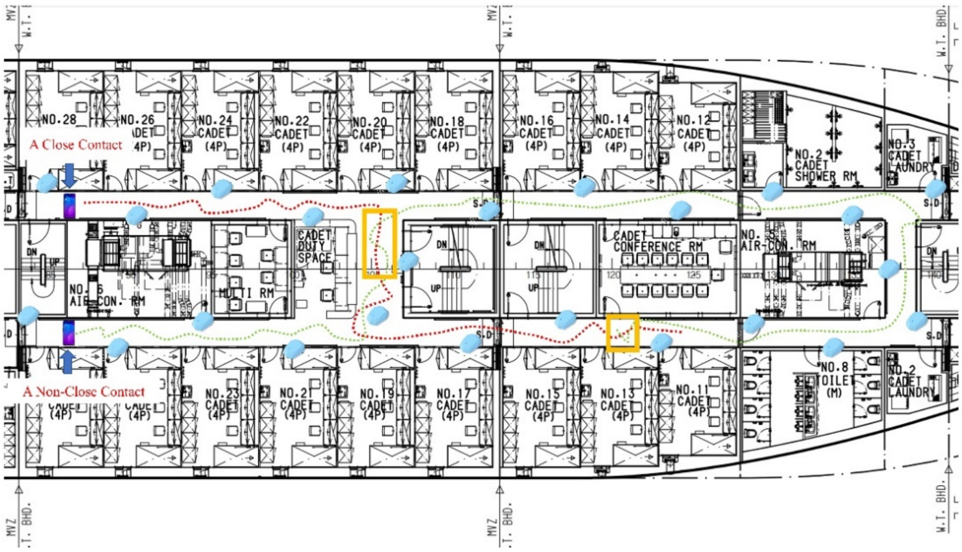


Fig. 8. Example of close contact tracing in a ship environment

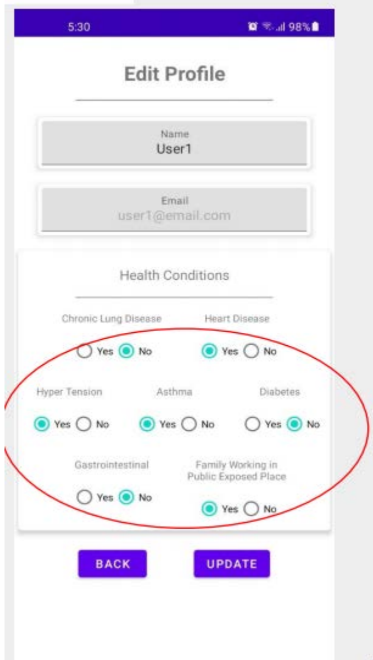


Fig. 9. Close contact tracing APP diagram [32]

Fig. 11 shows the results of the location points marked by the ship's passengers and close contacts. As the ship passenger moves, the Bluetooth device of the smartphone remains on. The Bluetooth device continuously transmits signals to the surrounding area. When the two devices meet the Bluetooth proximity requirement, Bluetooth communication becomes possible. When two devices are closed, the encountered time is recorded in the local database of the respective smartphone.

In **Fig. 11**, several close contacts are visible during the movement of ship passenger. Location points of close contact are highlighted in red. This is because Bluetooth is a proximity communication technology. Two Bluetooth devices can only communicate if they are close to one another. Based on this proximity property, the encountered location point is considered the location point of close contact in this paper. As shown in **Fig. 11**, these encounter location points are dispersed throughout the ship. As the number of ship passengers increases and the range of activities expands, the close contacts can be continuously tracked.

The results of the risky area calculation are shown in **Fig. 12**. Identifying the encounter location points is the basis for calculating the risky area. By inputting the encounter location points into the KDE-Alpha algorithm, the results of the risky area calculation can be obtained. As shown in **Fig. 12**, seven risky areas are calculated. The results of the risky area calculation can be viewed on the smartphones of the ship passenger. Therefore, the proposed app design can be used to trace close contacts without an Internet connection. It is possible to identify risky areas without knowing the actual location point of the close contacts.

To be able to show the results for risky areas in greater detail. In this paper, 18 ship passengers' location points are used as examples to illustrate the algorithm's results. Two ship passengers are present in some areas, while three ship passengers are present in another. In this paper, location points of close contacts are found based on encountered location points. There are 18 passengers on the ship who are either COVID-19 patients or close contacts. Seven activity areas for each of these ship passengers are created.

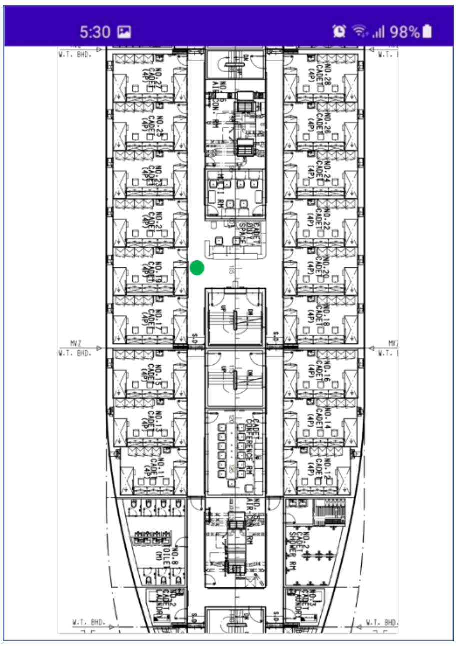


Fig. 10. A current user location point

The red boundary indicates that the ship passengers are close contacts and COVID-19 patients, respectively, and they belong to the same cluster. A red boundary has a length of at least two meters and a width of about one meter. The assumed number of ship passengers is 18. There are 7 risky areas. There is a distribution of ship passengers in different areas (**Fig. 13**).

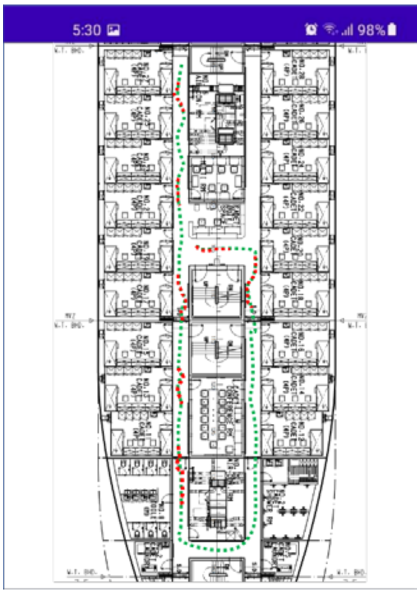


Fig. 11. Encountered user location points example

Close contact is defined as spending 15 minutes with a COVID-19 patient. It is necessary to obtain location points for these 18 ship passengers for a period of 15 minutes (900 seconds). There should be 900 location points per ship passenger [31].

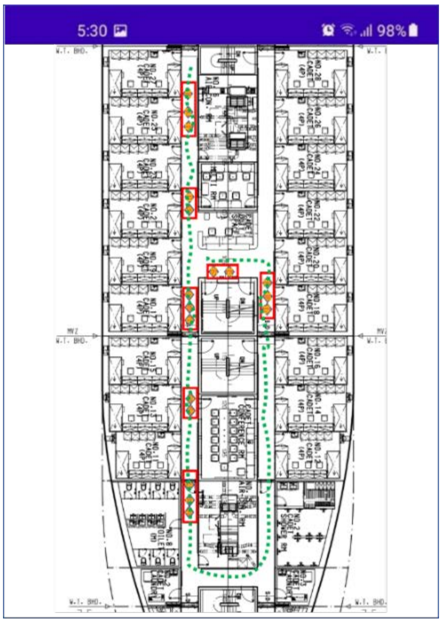


Fig. 12. Risky areas calculation results

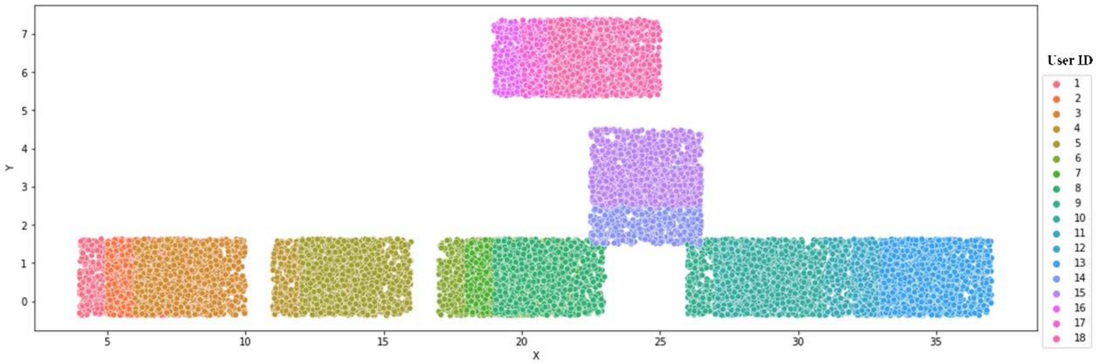


Fig. 13. Distribution of location points of COVID-19 patients and close contacts [31]

The Receiver Operating Characteristic (ROC) curve can be used to measure the performance of classification algorithms. The ROC curve is a graphic plot by True Positive Rate (TPR) and False Positive Rate (FPR). TPR and FPR are calculated by Eq. 1 and Eq. 2, respectfully [32].

$$TPR = \frac{TP}{TP+FN} \tag{1}$$

$$FPR = \frac{FP}{FP+TN} \tag{2}$$

TP (True Positive) is the test result of correctly determining the presence of a condition. *FN (False Negative)* is the test result of wrongly determining the absence of a condition. *FP (False Positive)* is the test result of wrongly determining the presence of a condition. *TN (True Negative)* is the test result of correctly determining the absence of a condition [32].

The result of the clustering algorithm is to assign a label to each user location point. This also means that the clustering algorithm can be tested using the ROC curve. Each user location point has a cluster ID (Identification Code) assigned by the clustering algorithm. The actual cluster ID is known. By comparing the estimated and real cluster IDs, TPR and FPR can be calculated. **Fig. 14** shows the ROC curves of CCIA, DBSCAN, Kmeans, and Hierarchical clustering algorithms. The Area Under Curve (AUC) is the area under the ROC curve. The higher value of AUC indicates that the probability of the classification algorithm ranking a randomly chosen presence of a condition is higher than a randomly chosen absence of a condition. The AUC of DBSCAN is 53%. The AUC of the Kmeans is 68%. The AUC of Hierarchical is 70%. The AUC of CCIA is 93%. The result shows that CCIA is more suitable for ship environments. **Fig. 15** shows the result of CCIA algorithm.

The procedure of determining the risky area is the same with classification algorithms, which classify the category of new observation into a number of classes or groups [33]. Each user location point is labeled with 0 or 1. The user location point of the boundary of the risky area is labeled as 1. The user location point of the inside of the risky area is labeled with 0. The presence of a condition is the user location point, which is labeled with 1. The absence of a condition is the user location point, which is labeled with 0. Alpha, KDE-Alpha, and Convex hull algorithms can make the result of labeling the user location point, which is determined as the presence of a condition or the absence of a condition. Therefore, TPR and FPR can evaluate the performance of Alpha, KDE-Alpha, and Convex hull algorithms.

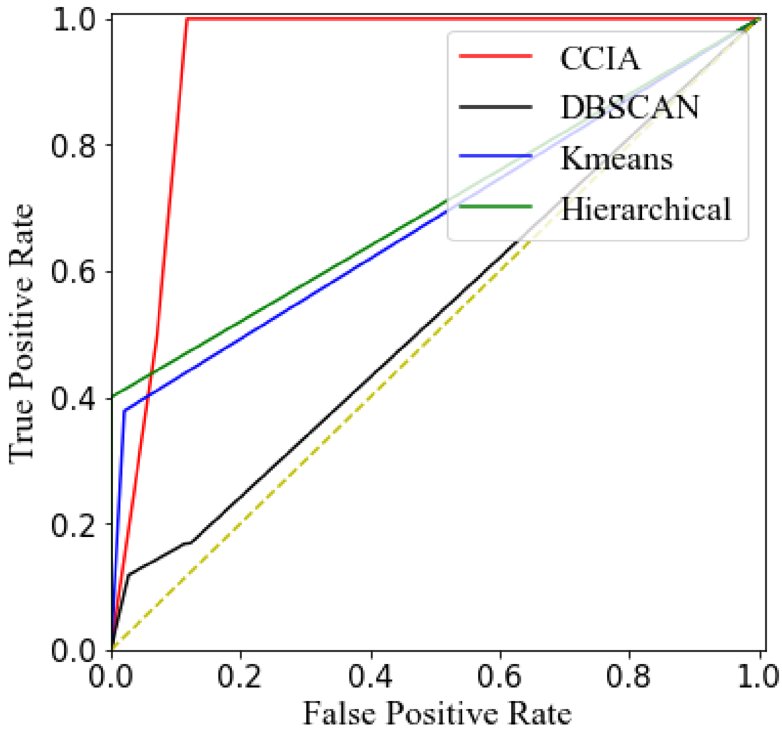


Fig. 14. The ROC curves of CCIA, DBSCAN, Kmeans, and Hierarchical clustering algorithms

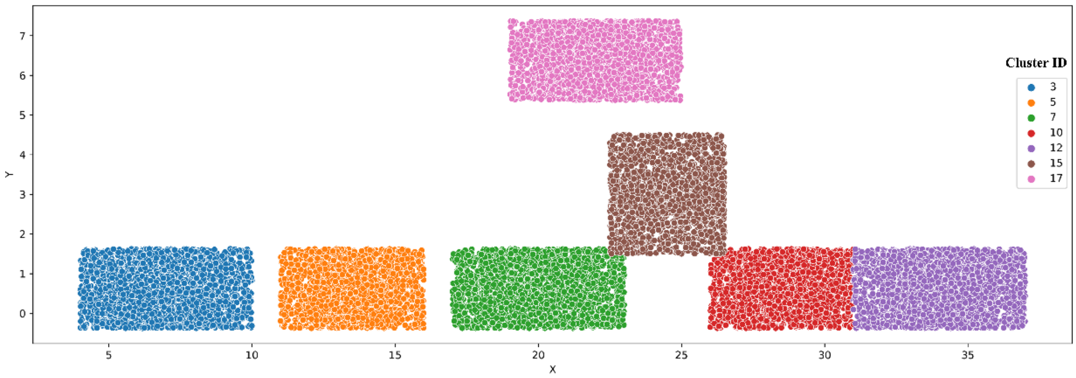


Fig. 15. The result of CCIA [32]

Fig. 16 shows the ROC curves of Alpha, KDE-Alpha, and Convex hull algorithms. The AUC of Alpha algorithm is 61%. The AUC of the Convex algorithm is 72%. The AUC of KDE-Alpha algorithm is 92%. The result indicates that the probability of correctly determining the presence or absence of a condition of KDE-Alpha is higher than Alpha and Convex hull algorithms. **Fig. 17** shows the result of using KDE-Alpha algorithm.

The self diagnosis for close contacts UI can be seen in **Fig. 18**. Selecting a close contact is based on the presence or absence of the corresponding symptom. Close contacts do not need to enter text information. As soon as the symptoms are selected, the app will further analyze them based on the selected symptoms. In general, the more symptoms, the greater the risk of transmission. The longer period of time spent with the close contact, the greater the likelihood of confirming the diagnosis of COVID-19.

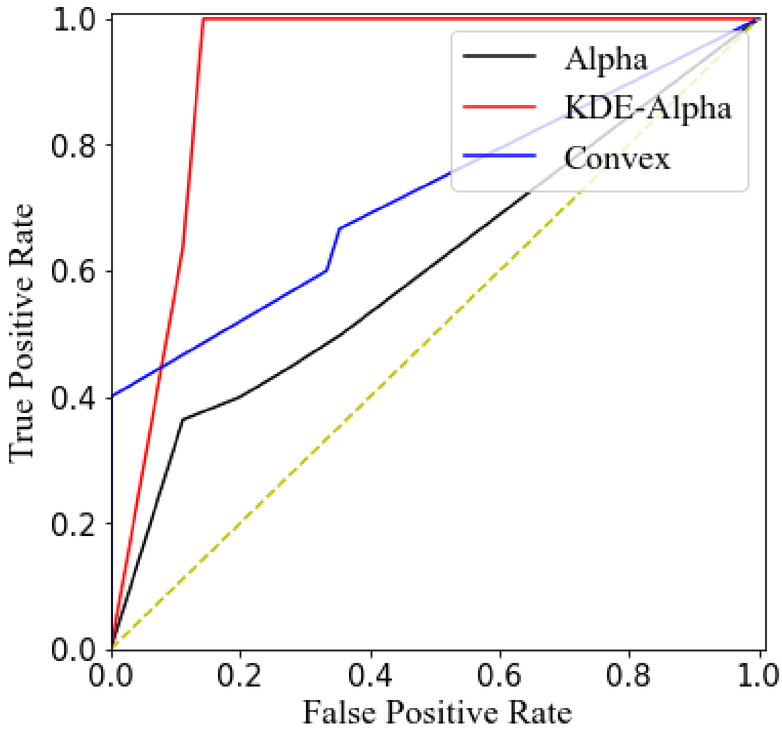


Fig. 16. The ROC curves of Alpha, KDE-Alpha, and Convex hull algorithms

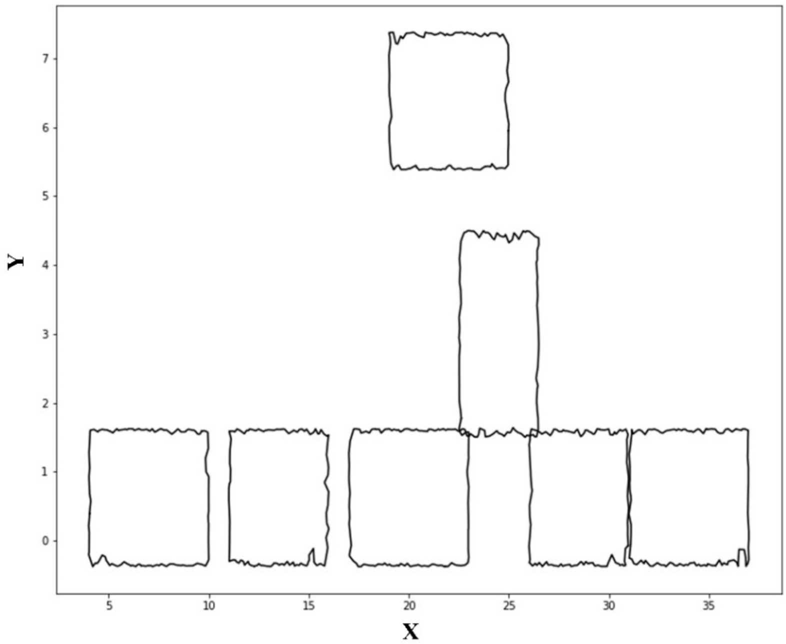


Fig. 17. The result of using KDE-Alpha algorithm

Fig. 19 shows the results of the risky area identification. It can be seen that the areas where COVID-19 patients and close contacts are clustered are accurately depicted. Fig. 19 illustrates the calculation results of the different risk levels for each risky area. Four of these areas are classified as high risk. Three of these areas are classified as low risk. Risk classification is based on the combination of symptom type, the number of symptoms, and the duration of the encounter. The duration of the encounter can be calculated from the symptom type and number of symptoms. As a result, a risk level can be calculated to classify risky areas in more detail. By using the results of the risk calculation, the "transmissionRiskLevel" parameter of the Exposure Notifications API can be updated. When the smartphone is connected to the server, the risk level of the risky area can be updated.

Fig. 18. Self diagnosis for close contacts UI [34]

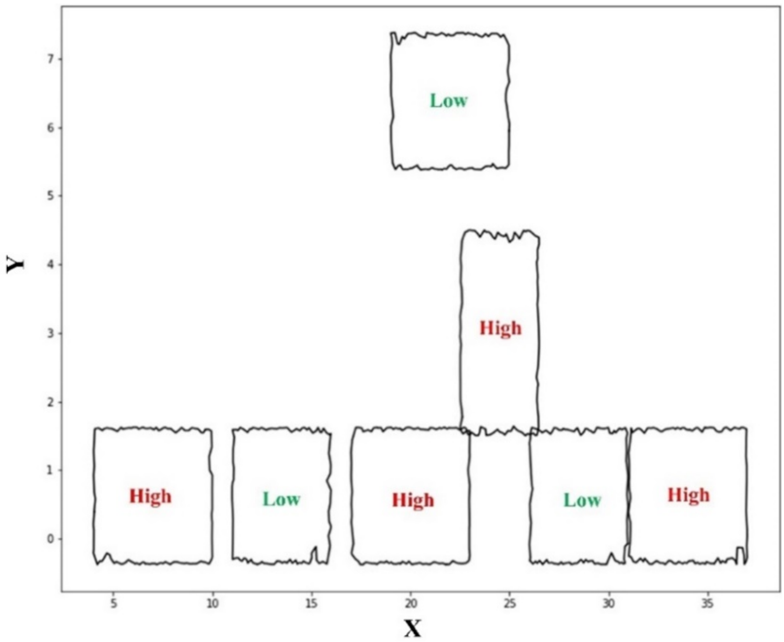


Fig. 19. Example of virus transmission risky area identification

5. Conclusions

This paper combines beacon and Exposure Notification API to propose ideas applicable to the ship environment without the Internet to trace close contacts. Firstly, the beacon can allow smartphones to get to their location point. Secondly, the Exposure Notification API can provide the encountered time with close contacts. Finally, the location point of contact with close contacts can be inferred by comparing the time with the historical location point of non-close contacts. A virus transmission risk indicator can be calculated from the symptoms entered by the close contacts on the app and the suspected location point. The virus transmission risk indicator can be updated to each smartphone terminal via Bluetooth based on the Exposure Notification API. This enables the tracking of close contacts without the Internet. This approach does not expose the personal information of close contacts and fully protects the personal privacy of close contacts. Our proposed method is suitable for closed ship environments.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (grant number: 2021R1I1A3056125).

References

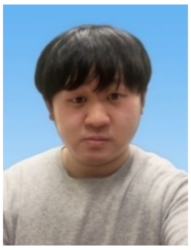
[1] Centers for Disease Control and Prevention, Close Contact, [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/appendix.html#:~:text=Close%20Contact%20through%20proximity%20and,a%20total%20of%2015%20minutes, Accessed on: July. 18, 2022.
[2] Natalie. Klein, "International law perspectives on cruise ships and COVID-19," *J. Int. Humanit. Leg. Stud.*, vol. 11, no. 2, pp. 282–294, Dec. 2020. Article (CrossRefLink)
[3] B. Batista, et al, "Minimizing disease spread on a quarantined cruise ship: A model of COVID-19 with asymptomatic infections," *Math. Biosci*., vol. 329, no. 108442, pp. 1-11, Aug. 2020. Article (CrossRefLink)
[4] Ewout. Fanoy, et al, "A. Outbreak of COVID-19 on an industrial ship," *Int. Marit. Health*., vol. 72, no. 2, pp. 89-92, Jun. 2021. Article (CrossRefLink)
[5] T. von Münster, et al, "When the ship comes in–COVID-19 Ausbrüche an Bord von Frachtschiffen im Hamburger Hafen," *Flugmed. Trop. Reisemedizin-FTR*. vol. 28, no. 01, pp. 13-24, Feb. 2021. Article (CrossRefLink)
[6] G. Battineni, et al, "Assessment of awareness and knowledge on novel coronavirus (COVID-19) pandemic among seafarers," *Healthcare*, vol. 9, no. 2, pp. 1-11, Jan. 2021. Article (CrossRefLink)
[7] E. Çakır, "Fatal and serious injuries on board merchant cargo ships," *Int. Marit. Health.*, vol. 70, no. 2, pp. 113-118, Jun. 2019. Article (CrossRefLink)
[8] H. Karahalios, "The contribution of risk management in ship management: The case of ship collision," *Safety Science*, vol. 63, no. 2014, pp. 104-114, Nov. 2013. Article (CrossRefLink)
[9] L. Hans, et al, "Quantitative risk analysis–Ship security analysis for effective risk control options," *Saf. Sci.*, vol. 58, no. 2013, May. 2013. Article (CrossRefLink)
[10] Svilicic, B., Kamahara, J., Celic, J. et al, "Assessing ship cyber risks: A framework and case study of ECDIS security," *WMU J. Marit. Aff.*, vol. 18, no. 3, pp. 509-520, Oct. 2019. Article (CrossRefLink)
[11] M. M. Plucinski, et al, "Coronavirus disease 2019 (COVID-19) in Americans aboard the Diamond Princess cruise ship," *Clinical Infectious Diseases*, vol. 72, no. 10, pp. 448-457, May. 2021. Article (CrossRefLink)

[12] P. Azimi, et al, "Mechanistic transmission modeling of COVID-19 on the Diamond Princess cruise ship demonstrates the importance of aerosol transmission," *Proceedings of the National Academy of Sciences*, vol. 118, no. 8, pp. 1-8, Feb. 2021. Article (CrossRefLink)
[13] J. Aljohani, et al, "Evaluating the Dynamics of Bluetooth Low Energy Based COVID-19 Risk Estimation for Educational Institutes," *Sensors*, vol. 21, no. 19, pp. 1-14, Oct. 2021. Article (CrossRefLink)
[14] Z. Su, et al, "Performance evaluation of COVID-19 proximity detection using bluetooth LE signal," *IEEE access*, vol. 9, no. 2021. pp. 38891-38906, Mar. 2021. Article (CrossRefLink)
[15] A. Mariscotti, "Experimental determination of the propagation of wireless signals on board a cruise ship," *Measurement*, vol. 44, no. 4, pp. 743–749, May. 2011. Article (CrossRefLink)
[16] P. G. Madoery, et al, "Feature selection for proximity estimation in COVID-19 contact tracing apps based on Bluetooth Low Energy (BLE)," *Pervasive and Mobile Computing*, vol. 77, no. 2021, pp. 1-13, Oct. 2021. Article (CrossRefLink)
[17] M. S. Munir, et al, "When CVaR meets with Bluetooth PAN: A physical distancing system for COVID-19 proactive safety," *IEEE Sensors Journal*, vol. 21, no. 12, Mar. 2021. Article (CrossRefLink)
[18] A. Bellouch, et al, "A Markov Model for Improving the Performance of COVID-19 Contact Tracing App," in *Proc. of International Conference on Innovations in Bio-Inspired Computing and Applications*, vol. 419. no. 2021, pp. 88-97, Feb. 2021. Article (CrossRefLink)
[19] E. Hernández-Orallo, et al, "Evaluating the effectiveness of COVID-19 Bluetooth-Based smartphone contact tracing applications," *Applied Sciences*, vol. 10, no. 10, pp. 1-19, Oct. 2020. Article (CrossRefLink)
[20] S. Kumar, et al, "Social Distancing using Bluetooth Low Energy to Prevent the Spread of COVID-19," in *Proc. of 2021 11th International Conference on Cloud Computing, Data Science & Engineering*. Article (CrossRefLink)
[21] P. C. Ng, et al, "COVID-19 and your smartphone: BLE-based smart contact tracing," *IEEE Systems Journal*, vol. 15, no. 4, pp. 5367-5378, Mar. 2021. Article (CrossRefLink)
[22] C. Sandeepa, et al, Montreal, "Canada. Social interaction tracking and patient prediction system for potential COVID-19 patients," in *Proc. of 2020 IEEE 3rd 5G World Forum (5GWF)*, 2020. Article (CrossRefLink)
[23] D. J. Leith and S. Farrell, "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, Sep. 2020. Article (CrossRefLink)
[24] R. Flood, et al, "Checking Contact Tracing App Implementations," *ICISSP*, vol. 1, no. 1, pp. 133-144, Feb. 2021. Article (CrossRefLink)
[25] M. Ahmed, et al, "Everywhere: A Framework for Ubiquitous Indoor Localization," *IEEE Internet of Things Journal*, Nov. 2022. (Early Access) Article (CrossRefLink)
[26] X. Yang, Z. Wu, and Q. Zhang, "Bluetooth Indoor Localization With Gaussian–Bernoulli Restricted Boltzmann Machine Plus Liquid State Machine," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-8, Jan. 2022. Article (CrossRefLink)
[27] Q. Lin and J. Son, "Sustainable Ship Management Post COVID-19 with In-Ship Positioning Services," *Sustainability*, vol. 14, no. 1, pp. 1-23, Dec. 2021. Article (CrossRefLink)
[28] Q. Lin and J. Son, "An In-ship Localization Algorithm For Close Contact Identification," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 5, pp. 1706-1717, Mar. 2022.
[29] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *The annals of Statistics*, vol. 38, no. 5, pp. 2916-2957, Oct. 2010. Article (CrossRefLink)
[30] S. Asaeedi, F. Didehvar, and A. Mohades, "α-Concave hull, a generalization of convex hull," *Theoretical Computer Science*, vol. 702, no. 2017, pp. 48-59, Nov. 2017. Article (CrossRefLink)
[31] Q. Lin and J. Son, "A Clustering Mechamism to Identify Close Contact for the Ship Passenger Health," in *Proc. of 2022 Prognostics and Health Management Conference (PHM 2022)*. Article (CrossRefLink)

[32] S. H. Park, J. M. Goo, and C. H. Jo, "Receiver operating characteristic (ROC) curve: practical review for radiologists," *Korean journal of radiology*, vol. 5, no. 1, pp. 11-18, Mar. 2004. Article (CrossRefLink)

[33] M. M. Saritas, A. Yasar, "Performance analysis of ANN and Naive Bayes classification algorithm for data classification," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 2, pp. 88-91, June. 2019. Article (CrossRefLink)

[34] Q. Lin, J. Son, and Z. Tao, "A Close Contact Tracing Method Applicable to Ship Environments," in *Proc. of 17th Asia Pacific International Conference on Information Science and Technology (APIC-IST 2022)*, 2022.



QIANFENG LIN is currently pursuing Ph.D. with the Department of computer engineering, Korea Maritime and Ocean University, Busan, South Korea. His research interests include indoor positioning, deep learning, machine learning, data mining, and maritime intelligence.

JOOYOUNG SON received the Bachelor of Science degree from the Seoul National University, Seoul, Korea in 1985, the Master of Engineering degree in computer engineering from the Seoul National University in 1993, and the Ph.D. degree in computer engineering from the Seoul National University in 1997. He was a senior researcher at media communication laboratory, LG electronics during 1985 and 1998. He is currently a professor at division of marine IT engineering, Korea Maritime and Ocean University, Busan, Korea since 1998. His research interests include in-ship localization, telecommunication systems in ships and at sea, and optimization for maritime affairs.