International Journal of Engineering & Technology, 7 (3.19) (2018) 4-19

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Generation of an Indoor 2D Map and Track Encryption Based on Mobile Crowdsourcing

Tianyang Cao

[1]*Student Member, IEEE, Tianhao Xue, Lei Hong, Hao Zhou, Yubo Song*
School of Information Science and Engineering, Southeast University
Nanjing 210096, China, Email: {ctymy, 213153679, 213153492, 213151590, songyubo}@seu.edu.cn

Abstract

The widespread application of mobile crowdsourcing modes provides new ideas for generating indoor maps. By collecting and analyzing the trajectory datas of users properly, we can obtain the location information of indoor paths. Unfortunately, currently studies usually rely heavily on a satellite location, which restricts their indoor application. In this paper, a simple and practical method of generating indoor maps on Andriod platform is presented, and this method is able to correct deviation duly. User's datas collected by several bulit-in sensors are preprocessed utilizing Gaussian filter, after which we adopt feature recognition to confirming one's walking track based on multiple experiment datas. In order to integrate tracks generated by different persons, we then propose a new data structure based on a transition probability that can be updated online to store track information. In addition, we minimize possible deviations by testing the signal power launched by four Bluetooth base stations. Discrete tracks are finally integrated into a complete indoor map using a graph_based model. We then propose a novel encryption scheme exploiting chaos in a nonlinear digital filter, where secure key generation methods are discussed in detail. The secure key scheme includes: 1)channel measurement 2)a decorrelation transform 3)multibit adaptive quantization and encoding. Experiments are conducted in rectangle fields of 8m*8m, 44m*44m, respectively, and the results show our method can attain a maximum error of 5.94%.

Keywords: mobile crowdsourcing, feature recognition, track integration, unlinear digital filter, multibit adaptive quantization

1. Introduction

With the rapid development of information technology, navigation technology has received increasingly attention. It plays an essential role in urban life, making daily travelling smarter and faster. For instance, Kaixu Liu, et.al provides an overview on navigation services for indoor and outdoor users [1]. In particular, navigation services based on global satellite systems are widely studied and several designs and performance evaluations are presented in [2], [3] and [4]. In addition, energy-efficient strategies in water navigation and localization are described in [5].
. In the framework of navigation technology, map generation occupies significant stature, without which users have no possibility of obraining access to localization information. Satellite signals are utilized to recognize the features of construction and roads and to measure the distance between two objects in three-dimensional space [6], [7] and [8]. Meanwhile, users are also required to carry terminal devices to receive GPS signals. Recently, wireless local area networks (WLAN) are commonly available for most terminal devices. As the main technology standard of WLAN, WIFI communication provides a flexible way to generate maps and obtain localization [9], [10].
. While location_based map services have been well established outdoor, they are still not commonly available in indoor environments. In addition, traditional ways of generating maps (GPS, WLAN, etc) usually depend too much on specially-designed devices, communication networks and surveying and mapping personnel, which makes it expensive and inconvenient when the construction floor area increases. The mobile crowdsourcing mode is a new concept which has been proposed and discussed recently. It enables people to collect and share a large volume of data through their mobile devices, [11] [12] [13]. However, to our knowledge, few studies have concentrated on its application of indoor map generation, which motivates our paper's work. In this paper, track data generated by a single user is transmitted to cloud server after encrypting, where these datas are analyzed and intergrated into a complete map. Track datas stored on cloud server is updated in real-time, which makes mapping more accurate.
The remainder of this paper is organized as follows: In section II, we present the flow diagram of our whole scheme, while we also give the design method of a pedometer based on self-adaption step recognition, which is utilized to produce single user's track information. Meanwhile, we discussed the method of correct deviation according to signal power launched by bluetooth devices. In subsection III.A, the new data structure we propose is described in details, and track integration algorithm based on maximum probability track is used for mapping. In subsection.III.B, we describe the basic principle of proposed chaos DSP encryption, followed by a discussion of the stability condition of the encryption system and the shared key parameter generation scheme. The experimental setup, results and analysis are presented in section IV. Finally, we provide the conclusion in section V
Notation: The mathematical symbols are defined as follows. We use $\mathrm{Tr}(\mathbf{A})$, $\det(\mathbf{A})$, $\mathrm{vec}(\mathbf{A})$, $\mathbf{A}^T$ to denote the trace, determinant, matrix vector operator and transpose, respectively. A unit vector and identity matrix is denoted by $\mathbf{1}_M$, $\mathbf{I}_M$. Besides, $\mathbf{A}_F$, $\mathbf{A}_1$, $\mathbf{A}_\infty$ are used to represent the Frobenius- norm, 1-norm and infinity-norm, respectively.

2. The Framework of Our Design

Generally, Our proposed scheme can be divided into three parts: single user's path tracking, segmentation and processing of track information, generation of map structure. The flow diagram of our design is shown in Fig.1.

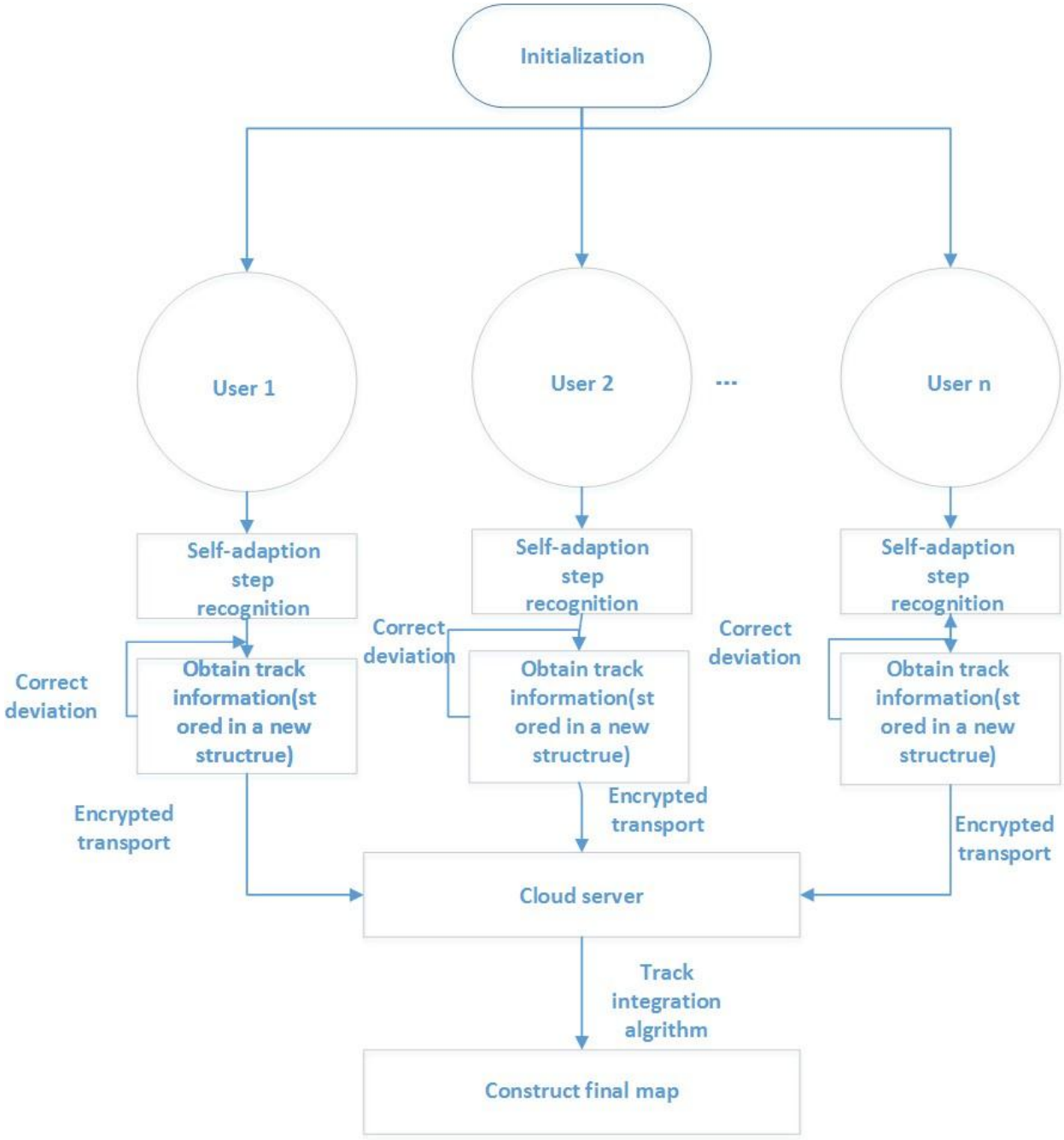


Fig. 1. Flow diagram of generating maps

A. Obtain Single User's Path Tracking

As is discussed in section II, no special survey instrument should be used, so all the datas are collected from built-in sensor devices. The bulit-in sensors of an Android mobile phone mainly includes a gravity sensor, an acceleration sensor, magnetic sensor, etc. In several related research [14], [15] integrated acceleration is utilized to calculate dispacement. Although integrated acceleration is easy to compute, experiments show deviation is quite obvious while integration time becomes longer. Therefore, we propose to adopt the scheme of step counting, i.e., vertical acceleration is collected to deduce dispacement and horizontal acceleration is collected to deduce direction. Let $[a_z1 , a_z2 , a_z3 , a_zN]$ be a uniform sampling of vertical acceleration. As is presented in Fig.2, when a user walks at a low speed, $[a_z1 , a_z2 , a_z3 , a_zN]$ appears to be a periodic sequence approximately: vertical acceleration increases when his center of gravity rises and decreases when his center of gravity drops.

As an example, the image sampling datas is shown in Fig.3. What's more, as one's walking frequence usually ranges from 1HZ-3HZ, gaussian lowpass filtering is utilized to restrain random interference:

$$\begin{cases} h_G(k) = \frac{1}{2\pi\rho^2} e^{-\frac{k^2}{2\rho^2}} \\ h(k) = \frac{h_G(k)}{\sum_k h_G(k)} \end{cases} \quad (1)$$

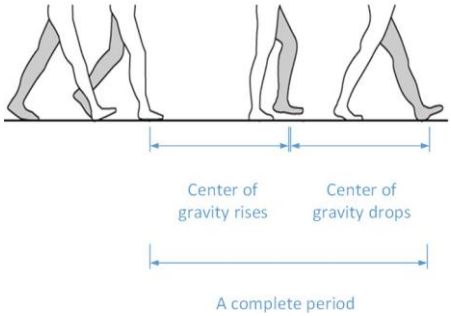


Fig. 2. Feature of a user's walk process

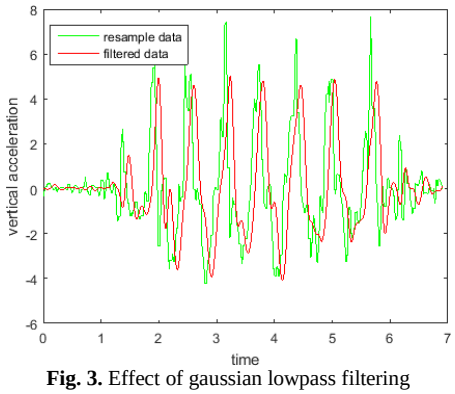


Fig. 3. Effect of gaussian lowpass filtering

As is clearly presented in Fig.3, the number of steps can be approximately reflected by the number of peaks, thus we consider using the following algorithm to identity the number of peaks:
In algorithm 1 $[x]$ denotes Gaussian rounding function.
On the other hand, walking direction of users can be easily identified by the variety of datas collected by acceleration sensors in X and Y direction. Let a_x, a_y denote the accel- eration in X and Y direction, respectively. The change of a_x, a_y when the user swerves is shown in table.1. As the built- in sensors are proved to be extremely sensitive, while most paths indoor actually have only several certain direction, we considering restricting walking direction in 8 special angles in Fig.4. What's more, criterion to identify walking direction is also presented in Fig.4.

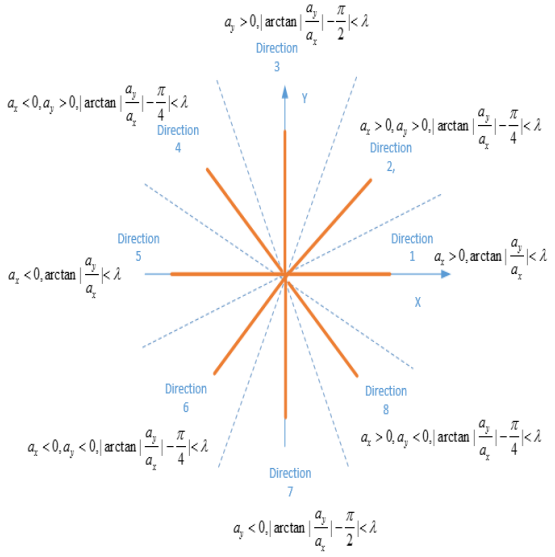


Fig. 4. 8 special angles and the criterion to identify walking direction

Combine the number of steps and waking direction, the trace of one single user can be generated in client. An eaxmple is given in Fig.5, where the length of step is set as 0.5m.

To estimate the range of the peak of vertical acceleration, a user is required to take p steps to provide a effective sample, and the peak of vertical acceleration during this process is recorded: $[max_1, max_2, \cdots, max_p]$. The detailed steps are shown in Algorithm 1.

Algorithm 1: Identify the Number of Peaks

Input: N, $[a_{z_1}, a_{z_2}, a_{z_3}, \cdots a_{z_N}]$,$[x_1, x_2, \cdots, x_p]$, $h(k)$, $budder$
Output: $[r_1, r_2, \cdots, r_N]$, where r_i represents if a_{z_i} is a peak.

To estimate the range of the peak of vertical acceleration, a user is required to take p steps to provide a effective sample, and the peak of vertical acceleration during this process is recorded:$[x_1, x_2, \cdots, x_p]$.;
Let $pl = min\,[x_1, x_2, \cdots, x_p]$.;
$\mathbf{Q}_0 = \varnothing$,$c = 0$,;
for $i = 1, i \leq N$ **do**
  $a_{z_i} \leftarrow a_{z_i} h(i)$, where $h(i)$ is defined in (1);
end
for $i = 1, i \leq N$ **do**
  if $x_{z_i} < pl, x_{z_{i-1}} > pl$ **then**
    $c \leftarrow c + 1$;
  end
  if $x_{z_i} > pl, x_{z_{i-1}} > pl$ **then**
    $\mathbf{Q}_c \leftarrow \mathbf{Q}_c \bigcup \{i\}$;
  end
end
$cl = c$,;
for $j = 1; j \leq cl$ **do**
  if $max\{\mathbf{Q}_j\} - min\{\mathbf{Q}_j\} < budder$ **then**
    $r_{min\{\mathbf{Q}_j\}} = r_{min\{\mathbf{Q}_j\}+1} = \cdots = r_{max\{\mathbf{Q}_j\}} = 0$;
  end
  else
    $h_1 = min\{\mathbf{Q}_j\}$;
    $h_2 = max\{\mathbf{Q}_j\}$;
    while $h_2 - h_1 \geq 2$ **do**
      if $x_{h_1+[\frac{2}{3}(h_2-h_1)]+1} < x_{h_1+[\frac{2}{3}(h_2-h_1)]} < x_{h_1+[\frac{2}{3}(h_2-h_1)]-1}$ **then**
        Update h_2 by;
        $h_2 \leftarrow h_1 + [\frac{2}{3}(h_2 - h_1)]$;
      end
      elsif $x_{h_1+[\frac{1}{3}(h_2-h_1)]+1} > x_{h_1+[\frac{1}{3}(h_2-h_1)]} > x_{h_1+[\frac{1}{3}(h_2-h_1)]-1}$ **then**
        Update h_1 by;
        $h_1 \leftarrow h_1 + [\frac{1}{3}(h_2 - h_1)]$;
      end
    end
  end
  $r_{h_2} = 1$, $r_{min\{\mathbf{Q}_j\}} = r_{min\{\mathbf{Q}_j\}+1} = \cdots = r_{h_1} = r_{h_2+1} = \cdots = r_{max\{\mathbf{Q}_j\}} = 0$;
end

B Correct Deviation Utilizing Ibeacon Devices

To avoid accumulated deviation in data processing, bluetooth communication devices with a maximum transmission distance of \$100m\$ are used in this scheme. Bluetooth protocols under the Android System are adopted to guarantee signal exchange. The principal of correcting deviation utilizing is presented in Fig.5 and the signal intensity received by the client can be expressed as $P_r = P_0 L^{\alpha}$, where L denotes the transmission distance.

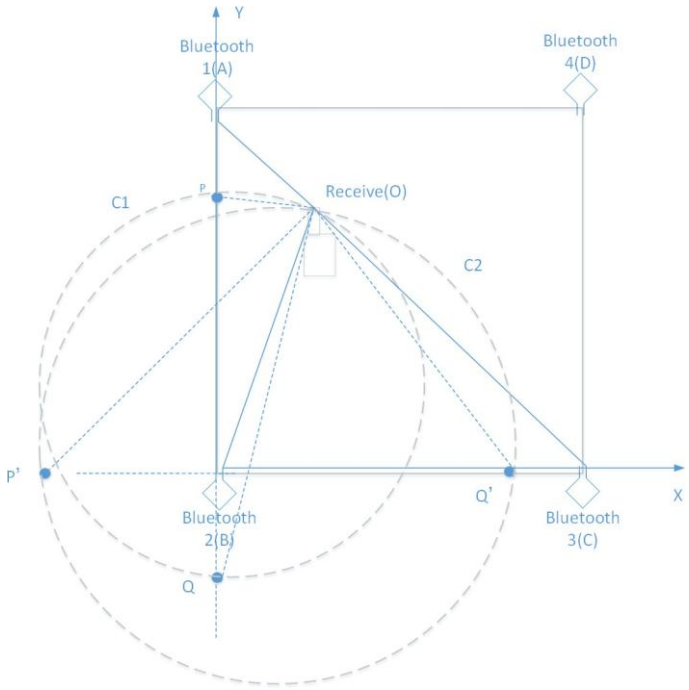


Fig. 5. The principal of correcting deviation by utilizing bluetooth base stations

As is shown in Fig.5, we have $(\frac{P_{r1}}{P_0})^{1/\alpha} = |OA|$, $(\frac{P_{r2}}{P_0})^{1/\alpha} = |OB|$, then it can be deduced $(\frac{P_{r1}}{P_{r2}})^{1/\alpha} = |\frac{OA}{OB}|$,where P_{r1}, P_{r2} can be measured by a receiving device. Sucn expression implies that O locates at an Apollonius Circle C_1 whose diameter is PQ, which satisfies:

$$\begin{cases} |BP| = \frac{|AB| * (\frac{P_{r2}}{P_{r1}})^{1/\alpha}}{(\frac{P_{r2}}{P_{r1}})^{1/\alpha}+1} \\ |BQ| = \frac{|AB|}{(\frac{P_{r2}}{P_{r1}})^{1/\alpha}+1} \end{cases} \quad (2)$$

From (2), the equation of C_1 can be calculated as:

$$x^2 + (y - \frac{|BP|-|BQ|}{2})^2 = (\frac{|BP|+|BQ|}{2})^2 \quad (3)$$

For bluetooth 2,3, the equation of corresponding Apollonius Circle can be similarly computed as is in (3), which implies the coordinate of user O can be uniquely confirmed by measuring the signal intensity coming from bluetooth (1),(2),(3) , i.e., $O_{c1} = C_1 \cap C_2$. Let $O_{c1}(x_1, y_1), O_{c2}(x_2, y_2), O_{c3}(x_3, y_3), O_{c4}(x_4, y_4)$ denote the intersection of Apollonius Circle utilizing the data from (1)(2)(3), (1)(2)(4), (1)(3)(4), (2)(3)(4), respectively. To minimize the accumulated deviation, we then propose to use the gravity center of quadrangle $O_{c1}O_{c2}O_{c3}O_{c4}$ to replace the coordinate generated by subsection (A) per 10 second. The gravity center of $O_{c1}O_{c2}O_{c3}O_{c4}$ is given by:

$$x_{cg} = \frac{\sum_{i=1}^{4}(x_i^2 y_{i+1} - x_{i+1}^2 y_i + x_i x_{i+1} y_{i+1} - x_i y_{i+1} y_i)}{3\sum_{i=1}^{4}(x_i y_{i+1} - x_{i+1} y_i)} \quad (4)$$

$$y_{cg} = \frac{\sum_{i=1}^{4}(x_i y_{i+1}^2 - x_{i+1} y_i^2 + x_i y_i y_{i+1} - x_i x_{i+1} y_i)}{3\sum_{i=1}^{4}(x_i y_{i+1} - x_{i+1} y_i)} \quad (5)$$

3.Map Intergration and Encrypted Transmission

Paths generated in mobile crowdsourcing based on a 2D map are designed to be maximum probablity roads in this scheme, i.e., the information collected through a single user's walking track is stored in special data structure and integrated to be the most authoritative path. This actually implies indoor map generated by statistical feature is infinitely close to real world path as long as the number of samples is large enough.

A. Methods to Integrate Individual Tracks

Firstly, we give a description of the data structure we propose to store track data. For a certain field of size $L \times L$, it is firstly divided into a grid pattern of $M \times M$, where $M \in \mathbb{N}$. Thus a graph G with M^2 elements is constrcted, where any element $G_{m,n}$ includes the following data:

$G_{m,n} \to nu$: For any $m, n \in [1, M]$, $G_{m,n} \to nu$ denotes the number of individual tracks which passes element (m, n). In section , the walking trace generated by single user has been obtained and we might as well set there are S walking tracks: $[C_1, C_2, \cdots, C_S]$. C_i is determined to pass (m, n) if and only if there exists a sampling point $R(x_0, y_0) \in C_j$ which satisfies $(m-1)\frac{L}{M} \le x_0 \le m\frac{L}{M}$, $(n-1)\frac{L}{M} \le y_0 \le n\frac{L}{M}$.

- $G_{m,n} \to [p_e, p_{ne}, p_n, p_{nw}, p_w, p_{sw}, p_s, p_{se}]$: In this array, transition probability from (m, n) to $(m + 1.n)$ is defined as $p_e = \frac{N_{m,n\to m+1,n}}{G_{mn}\to nu}$, $N_{m,n\to m+1,n}$ denotes the number of track C_i whose adjacent two sampling point passes (m, n) and $(m + 1, n)$, respectively. Similarly we have $p_{ne} = \frac{N_{m,n\to m+1,n+1}}{G_{mn}\to nu}, p_n = \frac{N_{m,n\to m,n+1}}{G_{mn}\to nu}, p_{nw} = \frac{N_{m,n\to m-1,n+1}}{G_{mn}\to nu}, p_w = \frac{N_{m,n\to m-1,n}}{G_{mn}\to nu}, p_{sw} = \frac{N_{m,n\to m-1,n-1}}{G_{mn}\to nu}, p_s = \frac{N_{m,n\to m,n-1}}{G_{mn}\to nu}, p_{se} = \frac{N_{m,n\to m+1,n-1}}{G_{mn}\to nu}$.

With the graph constructed above, we propose the following track integration algorithm based on dynamic programming (DP):

***Step*1**:

As is clearly described in Fig.6, a commom indoor map can be approximately divided into two parts–rooms (R) and paths (P). For grids in field P, the distribution of transition probablity in 8 directions of two adjacent grid along the path tends to have perfect

correlation, i.e., let $\mathbf{p}_{m,n} = G_{m,n} \to [p_e, p_{ne}, p_n, p_{nw}, p_w, p_{sw}, p_s, p_{se}]$, then there exsits $i, j \in \{-1,0,1\}, i^2 + j^2 \neq 0$, s.t.:

$$\max_{0\leq k\leq 7} R_{\mathbf{p}_{m,n},\mathbf{p}_{m+i,n+j}}(k) \gg \max_{0\leq k\leq 7} R_{\mathbf{p}_{m,n},\mathbf{p}_{m+i',n+j'}}(k) \quad (6)$$

where $(i', j') \neq (i, j)$ and the cross correlation function between sequence $\mathbf{p}_{m,n}$ and $\mathbf{p}_{m+i,n+j}$ is defined as:

$$R_{\mathbf{p}_{m,n},\mathbf{p}_{m+i,n+j}}(k) = \sum_{i=0}^{7} \mathbf{p}_{m,n}(i)\mathbf{p}_{m+i,n+j}(i+k)^* \quad (7)$$

From (6), field P can be roughly confirmed. In step 2, we will further find each possible path btween two nodes.

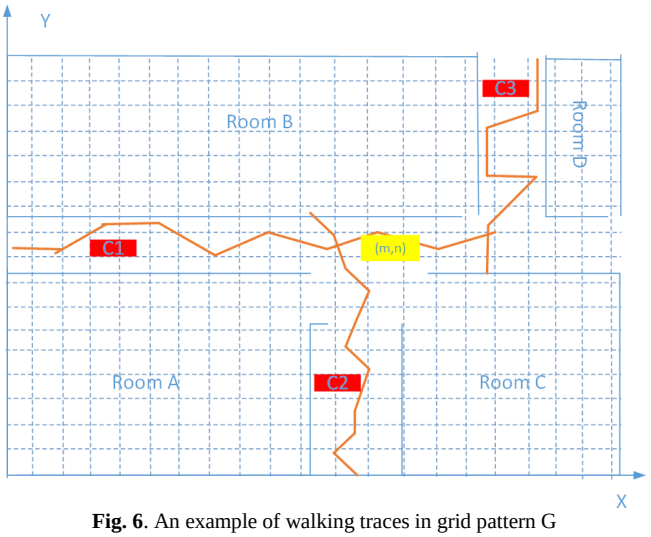


Fig. 6. An example of walking traces in grid pattern G

***Step*2:**
In step 2, we consider finding a maximun probability path between any two nodes $G_{i_0,j_0} \in P$ and $G_{i,j} \in P$, where $i_0 < i, j_0 < j$. This optimization problem can be expressed as:

$$\begin{aligned} w(i_0, j_0, i, j) &= \max \prod_{k=0}^{N} P\left(G_{i_k,j_k} \to G_{i_{k+1},j_{k+1}}\right) \\ s.t.: i_0 &\leq i_1 \leq i_2 \leq \cdots \leq i_N \leq i \\ j_0 &\leq j_1 \leq j_2 \leq \cdots \leq j_N \leq j \end{aligned} \quad (8)$$

To minimize the time complexity of track integration algorithm, bidirectional DP in field P is designed to process problem (8). Let $s(i_0, j_0, i, j) = \log_\lambda w(i_0, j_0, i, j)$, $\lambda \in (0,1)$, then it is easy to verify that problem (8) is equivalent to:

$$\begin{aligned} s(i_o, j_0, i, j) = \max_{\substack{\Delta i_0, \Delta j_0 \in \{0,1\} \\ \Delta i_0^2 + \Delta j_0^2 \neq 0 \\ \Delta i, \Delta j \in \{0,1\} \\ \Delta i^2 + \Delta j^2 \neq 0}} & s(i_0 + \Delta i_0, j_0 + \Delta j_0, i - \Delta i, j - \\ & \Delta j) + \log_\lambda P\left(G_{i_0,j_0} \to G_{i_0+\Delta i_0, j_0+\Delta j_0}\right) + \log_\lambda P\left(G_{i-\Delta i, j-\Delta j} \to G_{ij}\right) \end{aligned} \quad (9)$$

While $i - i_0 > 1$ **or** $j - j_0 > 1$

$$s(i_o, j_0, i, j) = \log_\lambda P\left(G_{i_0,j_0} \to G_{ij}\right) \quad (10)$$

While $i - i_0 \leq 1$ **or** $j - j_0 \leq 1$

With the result of problem (9) calculated, the optimal path between (i, j) and (i_0, j_0) can be recorded by algorithm 2:
For the situation when $i > i_0, j < j_0$, maximun probability path between G_{ij} and $G_{i_0 j_0}$ can be searched using the same principle.
Remark: Estimation of time complexity for the DP problem proposed in (9):
Lemma 1: For two-dimensional linear recurrence sequence:

$$s(i,j) = \alpha s(i-k,j) + \beta s(i,j-k) + \gamma s(i-k,j-k) \quad (11)$$

where the initial statue is set as $s(0,0) = \lambda$. If i, j, k satisfies $k|i, k|j$, then the upper bound of $s(i,j)$ can be approximated by:

$$\begin{aligned} s(i,j) \quad &\leq C_1\left(m + \frac{1}{2} + \frac{\sqrt{4m+1}}{2}\right)^{\frac{i+j}{2k}} \\ &+ C_2\left(m + \frac{1}{2} - \frac{\sqrt{4m+1}}{2}\right)^{\frac{i+j}{2k}} \end{aligned} \quad (12)$$

where $m = (1 + \frac{\alpha}{\beta})^{-2}$.
Proof: See appendix A.

Algorithm 2: Record the path between node $G_{i,j}$ and $G(i_0, j_0)$, $i_0 < i, j_0 < j$

Input: The value of $s(a,b,c,d)$, where $i_0 \leqslant a, c \leqslant i, j_0 \leqslant b, d \leqslant j$ and $(a,b) \in P, (c,d) \in P$, stack $A = \varnothing, B = \varnothing$, $count = 0$, $s(i_0, j_0, i, j)$, γ_{th} (which is the threshold value of $s(i_0, j_0, i, j)$)
Output: Maximun probability path $(i_1, j_1), (i_2, j_2), \cdots, (i_N, j_N)$.

```
if s(i_0, j_0, i, j) < γ_th then
    The path finded by DP problem is judged to be invalid;
end
else
    while i − i_0 > 1 or j − j_0 > 1 do
        (Δi_0, Δj_0, Δi, Δj) = arg max_{Δi_0, Δj_0 ∈ {0,1}, Δi_0^2 + Δj_0^2 ≠ 0, i, j ∈ {0,1}, i^2 + j^2 ≠ 0} s(i_0 + Δi_0, j_0 + Δj_0, i − Δi, j − Δj);
        i_0 ← i_0 + Δi_0, j_0 ← j_0 + Δj_0;
        i ← i − Δi, j ← j − Δj;
        Push (i_0, j_0) into stack A;
        Push (i, j) into stack B;
    end
    Take elements off the top of stack A in proper order: (i_m, j_m), (i_{m−1}, j_{m−1}), ···, (i_1, j_1);
    Take elements off the top of stack B in proper order: (i_N, j_N), (i_{N−1}, j_{N−1}), ···, (i_{m+1}, j_{m+1});
    Then the maximun probability path is finally obtained.
end
```

With lemma 1, as the DP problem (9) adopts the strategy of bidirectional search, the average time complexity can be approximately given by:

$$O\left(\tfrac{1}{2} s(i - i_0, j - j_0)|_{\alpha=\beta=\gamma=k=1}\right) \quad (13)$$

***Step*3:**
In step 3, we try to find key nodes in graph G, after which the maximun probability between any two nodes is connected to form the final indoor map. We propose to use half-line scanning to search key points in this scheme, as is presented in algorithm 3 and Fig.7, where $L(T_{PK_i})$ denotes the length of track T_{PK_i}, i.e., the

number of nodes T_{PK_i} passes and $|A|$ is defined as the number of elements in set A. The method of constructing the final map and amalgamating surplus tracks is also shown in algorithm 4.

Note that in algorithm 3, the angle interval $\Delta\theta$ should be properly decided, with a too large $\Delta\theta$ the precision of searching for key nodes can't be guaranteed, while a too small angle interval leads to enormous increase of computation complexity. In addition, line 22~31 is designed to delete redundancy nodes and edges in graph M, as it is easy to verify that actually only those nodes located in the junction of two paths are necessary for generating the map.

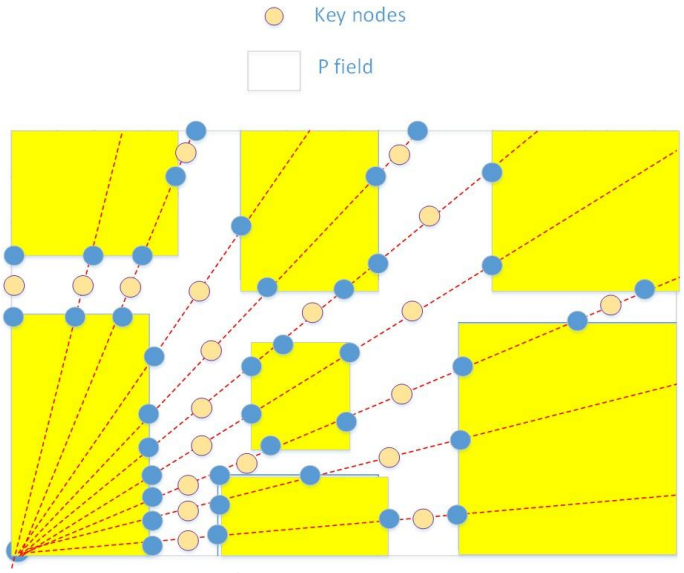


Fig. 7. Selection of key points

Algorithm 3: Search all key nodes in graph G

Input: Graph G with $M \times M$ nodes, whose field P and field R has been obtained roughly by step 1; node set $C = \varnothing$; $v_1 = 0$; node N; $co = 1$

Output: Key nodes in graph G: $K_1, K_2, \cdots$ and a new graph M which stores data information of the generated map

1 For a certain angle interval $\Delta\theta$, a set of half-line is constructed as $\left\{y = tan(k\Delta\theta)x | 0 \leqslant k \leqslant \frac{\pi}{2\Delta\theta}, k \in \mathbb{Z}\right\}$;

2 Calculate end_k by

$$end_k = \begin{cases} M & k\Delta\theta \leqslant \frac{\pi}{4} \\ \frac{M}{tan(k\Delta\theta)} & k\Delta\theta \geqslant \frac{\pi}{4} \end{cases} \tag{14}$$

for $k = 0, k \leqslant \frac{\pi}{2\Delta\theta}$ **do**

3 | **for** $m = 1, m \leqslant end_k$ **do**

4 | | Solve the following inequality with variable n;

5 | |

$$\frac{n-1}{m} \leqslant tan(k\Delta\theta) \leqslant \frac{m}{n-1} \tag{15}$$

| | $C = C \bigcap (m, n)$ according to ascending order of n.

6 | **end**

7 | **for** $i = 1, i \leqslant |C|$ **do**

8 | | **if** $C\{i\} \in R$ *while* $C\{i+1\} \in P$ **then**

9 | | | Update N by $N \leftarrow C\{i\}$;

10 | | **end**

11 | | **if** $C\{i\} \in P$ *while* $C\{i+1\} \in R$ **then**

12 | | | One key node K_{co} is decided by the middle point of node N and node $C\{i\}$;

13 | | | $co \leftarrow co + 1$;

14 | | **end**

15 | **end**

16 **end**

17 Create an empty graph M of the same size as G;

18 **for** $i = 1, i \leqslant |K|$ **do**

19 | **for** $j \geqslant i, j \leqslant |K|$ **do**

20 | | Add a new path between node K_i and K_j;

21 | **end**

22 **end**

23 **for** $i = 1, i \leqslant |K| - 1$ **do**

24 | **for** $j > i, j \leqslant |K| - 1$ **do**

25 | | Let $L_{min} = \min\left\{L(T_{PK_i}), L(T_{PK_j})\right\}$;

26 | | Take first L_{min} nodes in track T_{PK_i}, T_{PK_j}, respectively;

27 | | $G_{x_1y_1}^{T_{PK_i}}, G_{x_2y_2}^{T_{PK_i}}, \cdots, G_{x_{L_{min}}y_{L_{min}}}^{T_{PK_i}}$ and $G_{x_1y_1}^{T_{PK_j}}, G_{x_2y_2}^{T_{PK_j}}, \cdots, G_{x_{L_{min}}y_{L_{min}}}^{T_{PK_j}}$.

28 | | **if** $\sum_{p=1}^{L_{min}} |x_p^{T_{PK_i}} - x_p^{T_{PK_j}}| + |y_p^{T_{PK_i}} - y_p^{T_{PK_j}}| \leqslant 2L_{min}$ **then**

29 | | | Track T_{PK_i} and T_{PK_j} can be amalgamated, i.e., if $L(T_{PK_i}) < L(T_{PK_j})$, we delete node K_i and edge T_{PK_i} from graph M and v.v..

30 | | **end**

31 | **end**

32 **end**

Algorithm 4: Integrate $|K|(|K|+1)/2$ tracks obtained in algorithm 3 and generate final map

Input: $|K|(|K|+1)/2$ tracks connecting any two nodes K_i and K_j.
Output: Final map M.

Pretreatment: We divide graph G of $M \times M$ nodes into $D \times D$ blocks, where $D|M$. Then we mark these $(M/D) \times (M/D)$ blocks as B_{ij}, $1 \leqslant i, j \leqslant M/D$. ;
for $i = 1$, $1 \leqslant i \leqslant M/D$ **do**
  for $j = 1$, $1 \leqslant j \leqslant M/D$ **do**
    Since block B_{ij} contains some keynodes and tracks (two vertices of each track belongs to the key node set obtained in algorithm 3), firstly we delete those tracks which satisfy: One vertex of the track doesn't belong to B_{ij}. Note those remaining tracks and key nodes can be actually regarded as an undirected graph BT_{ij}. Let V_{ij} and E_{ij} denotes the vertex set and edge set of B_{ij}, respectively.;
    Obtain the minimalspanning tree of BT_{ij} utilizing prim algorithm:;
    Initialize: Node set $V_{new} = N_0$, where N_0 can be selected randomly from V_{ij}, edge set $E_{ij} = \varnothing$.;
    repeat
    Choose nodes u, v which satisfy:;

$$(u, v) = \arg \min_{\substack{u \in V_{new} \\ v \in V_{ij} - V_{new}}} f(L(u, v)) \tag{16}$$

    where the weight of edge $< u, v >$ is set as $f(L(u, v))$, $L(u, v)$ is the length of maximun probability path and $f(x)$ can be any monotonic increasing function.
    Add u into V_{new}, add $< u, v >$ into E_{new}.;
    until $V_{new} = V_{ij}$;
    The minimalspanning tree BT_{ij} contains E_{new} and V_{ij} (all tracks which don't belong to E_{new} are deleted), then the gravity center of BT_{ij} can be worked out by tree dynamic programming (tree DP);
    Initialize: Choose a vertex r from V_{ij} as the root node of BT_{ij}, where r satisfies there exists one and only one vertex $s \in V_{ij}, r \neq s$, s.t. $< r, s > \in E_{new}$.;
    Let $s(i)$ represent the number of nodes of the subtree whose root node is i, and $s(i)$ is given by the following recurrence relation:

$$s(i) = 1 + \sum_{j \in \mathcal{A}_i} s(j) \tag{17}$$

    Where $\mathcal{A}_i$ refers to the set of the subnodes of i. With $s(i)$ computed in (17), the gravity center of BT_{ij} can be expressed as:

$$g_{ij} = \arg \min_{g_{ij} \in V_{ij}} \max_{t \in \mathcal{A}_g} \{|V_{ij}| - s(g_{ij}), s(t)\} \tag{18}$$

  end
end
For any two adjacent block B_{ij} and $B_{i'j'}$, find the maximun probability path between g_{ij} and $g_{i'j'}$, add $< g_{ij}, g_{i'j'} >$ into graph M. ▷ For some reason we need to break here!

C A Novel Scheme for Encrypted Transmission

1) Basic Description of Proposed Encryption Schem

Security and privacy of the data exchange between users and the cloud server should be guaranteed by exploiting special encryption algorithm. Some effective ones includes DES, 3DES, ADS and chaotic methods, etc. In [16], it is mentioned that chaotic systems show splendid performance once the key generator is properly chosen. In [17], chaos in the linear digital filter (FIR) is designed to function as an encryptor

while its inverse instruction is used as the decryptor. In this paper, we furtherly propose an encryptor based on chaos in nonlinear digital filter, since it can (1) make the system more sensitive to coefficients (2) provide convenience for hardware implementation when digital signal processor (DSP) is combined with digital transmission. The formula of a nonlinear digital decryption system can be described as (19) with the help of [16]:

$$\begin{cases} x_1(n+1) = s(n) - g_1(x_1(n)) + \theta_c + \alpha x_2(n-1) \\ x_2(n+1) = g_2(x_1(n)) - x_2(n) + \theta_M \\ x_3(n+1) = -\beta x_3(n) - x_3(n-1) + x_2(n) \end{cases} \tag{19}$$

where $s(n)$ denotes the input sequence, i.e., the track data of one single user, while $x_3(n)$ functions as the encrypted sequence, α, β are the coefficients of the digital filter and θ_c, θ_M are the threshold value. The unlinear function $g_1(x)$ and $g_2(x)$ are given by:

$$g_1(x) = \begin{cases} \kappa \cdot x + \sigma & x \leq -\sigma/\kappa \\ 0 & \\ \kappa \cdot x - \sigma & x \geq \sigma/\kappa \end{cases} \tag{20}$$

and

$$g_2(x) = \begin{cases} -\kappa \cdot x - \sigma & x \leq -\sigma/\kappa \\ 0 & \\ -\kappa \cdot x + \sigma & x \geq \sigma/\kappa \end{cases} \tag{21}$$

correspondingly, the demodulation scheme can be obtained by calcualting the dual formula of (19), as is depicted in (22):

$$\begin{cases} x_5(n+1) = x_5(n) - g_2(x_4(n)) - \theta_M \\ x_6(n+1) = x_6(n-1) + \beta x_6(n) \\ r(n+1) = x_4(n+1) + g_1(x_4(n)) - \alpha x_5(n-1) - \theta_c \end{cases} \quad (22)$$

Data communication between the modulation and demodualtion module can be expressed as:

$$x_2(n+1) = x_5(n+1) \quad (23)$$

The DSP structure aiming at implementing the encryption and decryption is shown in Fig.8, it is easy to show that even for the attackers, to capture the transmitted signal $x_2(n+1)$, have no access to initial data sequences without the privacy key $\alpha, \beta, \theta_C, \theta_M$, while the cloud server can decrypt the encrypted sequences by utilizing the demodulation circuit, as is described in the diagram.

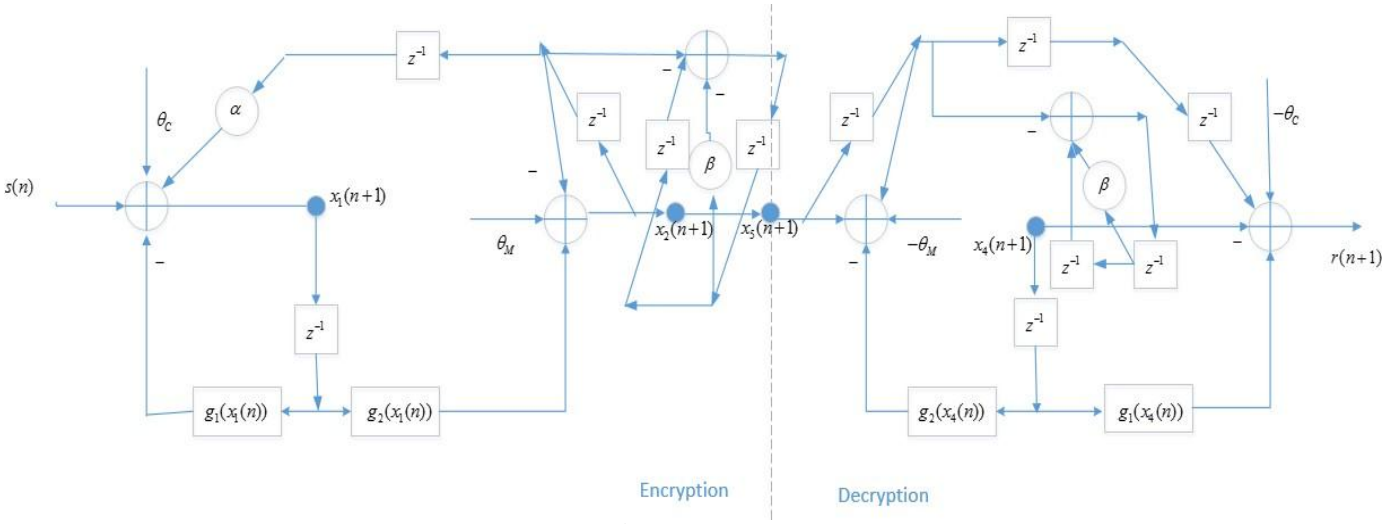


Fig. 8. The DSP Structure for Chaos Encryption and Decryption

2) *Discussion of Stability Condition:*

To make certain the modulation system is unstable to generate the chaos, it is resonable to discuss the possible value of parameter $\alpha, \beta, \theta_C, \theta_M$. Actually, it is equivalent to calculate the stable condition of systems as the unstable set is the complementary set of stable set.

Firstly, the nonlinear function $g_1(x)$ and $g_2(x)$ can be written as:

$$g_1(x) = (\kappa \cdot x - \sigma)u(x - \sigma/\kappa) + (\kappa \cdot x + \sigma)u(-x + \sigma/\kappa) \quad (24)$$

and

$$g_2(x) = (-\kappa \cdot x + \sigma)u(x - \sigma/\kappa) + (-\kappa \cdot x - \sigma)u(-x + \sigma/\kappa (25)$$

where $u(\cdot)$ is the unit step function. By using the power series of $u(x)$: $u(x) = 1 + \sum_{k=1}^{+\infty} \delta^k(0) \frac{x^k}{k!}$, where $\delta^k(\cdot)$ denotes the k-th order impluse doublet function, $g_1(x)$ can be furtherly written as :

$$\begin{aligned} & g_1(x) \\ & = 2\kappa x + \kappa \sum_{k=1}^{+\infty} \delta^k(0) \frac{(x-\sigma/\kappa)^{k+1}}{k!} \\ & + (\kappa x + \sigma) \sum_{k=1}^{+\infty} \delta^k(0) \frac{(-x+\sigma/\kappa)^k}{k!} \end{aligned} \quad (26)$$

which is similar for $g_2(x)$. Define $\mathbf{x}_n \triangleq [x_1(n), x_2(n), x_3(n)]^T$ and combine Eq. (19) and (26), (19) can be expressed as

$$\begin{aligned} & \mathbf{x}_{n+1} \\ & = \mathbf{A}\mathbf{x}_n + \mathbf{B}\mathbf{x}_{n-1} + \sum_{k=2}^{+\infty} \mathbf{Q}_i [x_1(n)^{2k}, x_2(n)^{2k}, x_3(n)^{2k}]^T \\ & + [s(n), 0, 0]^T \end{aligned} \quad (27)$$

With the conclusion of [17], the n-order discrete system stays stable if and only if the multidimensional z-transform:

$$\begin{aligned} & ||H(z_1, z_2, \cdots, z_n)||_{+\infty} \\ & \triangleq ||\sum_{i_1=0}^{+\infty} \sum_{i_2=0}^{+\infty} \cdots \sum_{i_n=0}^{+\infty} h(i_1, i_2, \cdots, i_n) z^{i_1} z^{i_2} \cdots z^{i_n}||_{+\infty} < +\infty \end{aligned} \quad (28)$$

where $h(i_1, i_2, \cdots, i_n)$ is the impulse response. Then it is easy to verify:

$$\begin{aligned} & \sum_{i_1=0, i_2=0, \cdots, i_n=0}^{+\infty} h^2(i_1, i_2, \cdots, i_n) \\ & = \frac{1}{2\pi} \int_0^{2\pi} |H(e^{jt_1}, e^{jt_2}, \cdots, e^{jt_n})|^2 dt_1 dt_2 \cdots dt_n < +\infty \end{aligned} \quad (29)$$

from (29), it's easy to show $|H(z_1.z_2, \cdots, z_n)|_{max} < +\infty$ is equivalent to $||h(i_1, i_2, \cdots, i_n)||_F < +\infty$, where $h(i_1, i_2, \cdots, i_n)$ can be determined by letting $s(k) = \delta(k)$ in (27). We employ the well-known inequality $||\mathbf{Ax}||_F \leq ||\mathbf{A}||_F ||\mathbf{x}||_F$ and $||\mathbf{A} + \mathbf{B}||_F \leq ||\mathbf{A}||_F + ||\mathbf{B}||_F$ to deduce:

$$||\mathbf{x}||_{n+1} \leq \det(\mathbf{A})||\mathbf{x}_n||_F + \det(\mathbf{B})||\mathbf{x}_{n-1}||_F + \delta(n) \quad (30)$$

(30) implicts $||\mathbf{x}_n||_F$ is convergent when characteristic equation $\lambda^2 - \det(\mathbf{B})\lambda - \det(\mathbf{C}) = 0$ has two complex roots within the unit circle in z-plane, i.e., $|\lambda_i| < 1$, $i \in \{1,2\}$. We substitute λ with $\lambda = \frac{t-1}{t+1}$, which yields:

$$\begin{aligned} & (1 - \det(\mathbf{B}) - \det(\mathbf{C}))t^2 - (2 + 2\det(\mathbf{C}))t + (1 + \det(\mathbf{B}) - \\ & \det(\mathbf{C})) = 0 \end{aligned} \quad (31)$$

where we obtain $|\lambda_i| < 1 \Leftrightarrow t_i < 0$, i.e, $t_1 + t_2 < 0$ and $t_1 t_2 > 0$.

3) *Shared Key Generation from Channel Measurement:*

In our proposed scheme, different bits of secret key (α, β, θ_C, θ_M) can be generated based on wireless channel measurement ([18]), whose main process can be decomposed into three steps (see Fig.9): channel response sampling, decorrelation transform and vector quantization.

A: Channel Response Sampling

As is depicted in Fig.9, we use Alice, Bob and Eve to represent the users, cloud server and malicious attackers, $\mathbf{H}_{AB}$ and $\mathbf{H}_{BA}$ to denote the measured channel matrix at Bob's node and Alice's node (which satisfies $\mathbf{H}_{AB} = \mathbf{H}_{BA}^T$), respectively. Thus all users carrying mobile devices can be modeled as moving antennas, the sampling interval time T_s should satisfy

$$1/T_s \geq 2 \max_i \frac{|\overrightarrow{v_{user_i}}|}{\lambda} \cos\theta_i \quad (32)$$

when Dolpper Shift is considered. We record the measured channel matrix obtained by uniform sampling as $\mathbf{H}_{AB}(\tau_i)$, where

$\tau_i = iT_s$. We then approximate the channel matrix at time : $i\tau \le t \le (i+1)\tau$ by :

$$\mathbf{H}_{AB}(t) = \mathbf{H}_{AB}(\tau_i) + (t-\tau_i)\frac{\mathbf{H}_{AB}(\tau_{i+1}) - \mathbf{H}_{AB}(\tau_i)}{\tau} \quad (33)$$

where $\mathbf{H}_{AB}(t)$ can be furtherly written as

$$\mathbf{H}_{AB}(t) = [\mathbf{h}_1^T, \mathbf{h}_2^T, \cdots, \mathbf{h}_N^T]^T \quad (34)$$

in the form of colmun vector. To transform the direct component in $\mathbf{h}_i$ to $\mathbf{0}$, we substract the mean value for each $\mathbf{h}_i$, i.e., $\mathbf{h}_i{}' \triangleq \mathbf{h}_i - \mu_i \mathbf{1}_N$, where μ_i is the mean value of $\mathbf{h}_i$.

B: **Decorrelation Transform**

To illustrate the necessity of decorrelation Transform, i.e, to make $\mathbb{E}\{\mathbf{h}'_i \mathbf{h}'^H_i\} = \mathbf{\Lambda}$ (where $\mathbf{\Lambda}$ is a diagonal matrix), we introduce a lemma:

Lemma 2: The bits available for key generation achieves maximal value if and only if $\mathbf{R} \triangleq \mathbb{E}\{\mathbf{h}'_i \mathbf{h}'^H_i\} = \mathbf{\Lambda}$.

Proof: See appendix B

To complete decorrelation transforming, we firstly rewrite $\mathbf{h}_i{}'$ by separating the real part and imaginary part of $\mathbf{h}_i{}'$:

$$\mathbf{y}_i \triangleq [(\Re(\mathbf{h}_i'))^T \quad (\Im(\mathbf{h}_i'))^T]^T \quad (35)$$

perform singular value decomposition (SVD) to $\mathbf{R}_a = \mathbf{h}'_i \mathbf{h}'^H_i$, i.e., $\mathbf{h}'_i \mathbf{h}'^H_i = \mathbf{U\Sigma V}^H$(36)

where $\mathbf{\Sigma} = \mathrm{diag}[\sigma_1, \sigma_2, \cdots, \sigma_N]$, $\mathbf{U}$ and $\mathbf{V}$ is the matrix comprised of eigenvectors of $\mathbf{R}_a\mathbf{R}_a^H$ and $\mathbf{R}_a^H\mathbf{R}_a$, respectively. Utilizing a linear transform $\mathbf{y}' = \mathbf{U}^H\mathbf{y}$, it can be verified:

$$\begin{aligned} \mathbb{E}\{\mathbf{y}'\mathbf{y}'^H\} &= \mathbf{U}^H\mathbb{E}\{\mathbf{y}\mathbf{y}^H\}\mathbf{U} \\ &= \mathbb{E}\{\mathbf{U}^H\mathbf{U\Sigma V U}^H\} \\ &= \mathbb{E}\{\mathbf{U}^H\mathbf{U\Sigma V V}^H\} \\ &= \mathbb{E}\{\mathbf{\Sigma}\} \end{aligned} \quad (37)$$

where the third equation holds for $\mathbf{R}_a$ is Hermite-positive definite, which implicts $\mathbf{R}_a\mathbf{R}_a^H = \mathbf{R}_a^H\mathbf{R}_a$.

C: **Vector Quantization**

For vector quantization, some related work has investigated a quantization scheme based on threshold value, i.e.

$$\mathbf{key}_i[n] = \begin{cases} 1 & \mathbf{y}'_i[n] > \gamma \\ 0 & \mathbf{y}'_i[n] < -\gamma \end{cases} \quad (38)$$

where γ is the therehold value, note that even if $\mathbf{y}'_i$, is captured by attackers, won't reveal any information as attackers don't know if $\mathbf{y}'_i[n] > \gamma$ or $\mathbf{y}'_i[n] < -\gamma$. However, two obvious drawbacks has been explained in [19]: 1)some bits are lost when the measured value locates in $[-\gamma, \gamma]$, 2)it's not possible to generate more than one bit according to one single measured value.

We propose a multibit adaptive quantization (MAQ) scheme based on discrete walsh transform (DWT) ([19]). Define $\mathcal{V}_n$: $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{2^n}\}$, where $\mathbf{v}_1 = \{0,0,\cdots,0\}$, $\mathbf{v}_2 = \{1,0,\cdots,0\}$, $\mathbf{v}_{2^n} = \{1,1,\cdots,1\}$, i.e., $\mathcal{V}_n$ is the set of n-dimensional 0-1 vectors. Then DWT can be expressed as:

$$W_f[\mathbf{y}] = \sum_{\mathbf{x}\in\mathcal{V}_n}(-1)^{f(\mathbf{x})+\mathbf{x}\cdot\mathbf{y}} \quad (39)$$

where $\mathbf{y} \in \mathcal{V}_n$ and $f(\mathbf{x})$ is an n-dimensional logical function. Assume n bits are required for key generation, for any given $\mathbf{y}'_i[k]$, we first normalize it by $\mathbf{y}''_i[k] = \mathbf{y}'_i[k] / \sum_{k=1}^{N} \mathbf{y}'_i[k]$, then the binary expression of $\mathbf{y}''_i[k]$ are given by: $\mathbf{y}''_i[k] = 0.\overline{a_1 a_2 \cdots a_n}$, $a_i \in \{0,1\}$. Let $\mathbf{c}_{i,k} = \{a_1, a_2, \cdots, a_n\}$ and $\{t_1, t_2, \cdots, t_n\}$ be a permutation of $\{1,2,\cdots,2^n\}$, which satisfies

$$W_f[\mathbf{y}_{t_n}] > W_f[\mathbf{y}_{t_{n-1}}] > \cdots > W_f[\mathbf{y}_1] \quad (40)$$

where $\mathbf{y}_0 = \{0,0,\cdots,0,0\}$, $\mathbf{y}_1 = \{0,0,\cdots,0,1\}$, $\mathbf{y}_{2^n} = \{0,0,\cdots,1,0\}$. Define

$$k(i,k) = \min_r: \{W_f(\mathbf{y}_{t_r}) > W_f(\mathbf{c}_{i,k}), \ 1 \le r \le 2^n\} \quad (41)$$

and

$$e(k) = \begin{cases} 1 & if\ k \bmod 4 = 2,3 \\ 0 & if\ k \bmod 4 = 0,1 \end{cases} \quad (42)$$

In this scheme, Bob (cloud server) transmits $[e(k(i,1)), e(k(i,2)), \cdots, e(k(i,2N))]$ to Alice (users). For $1 \le k \le 2N$, we quantify $\mathbf{y}''_i[k]$ by

$$z_i[k] = \overline{d_{e(k(i,k))}(k(i,k))} + j\overline{d_{e(k(i,k+N))}(k(i,k+N))} \quad (43)$$

where: $j = \sqrt{-1}$, $d_0(k)$ is the $\left[\frac{k-1}{4}\right]$ 2-bit Gray code and $d_1(k)$ is the $\left[\frac{k+1}{4}\right]$ 2-bit Gray code. Then $[z_i(1), z_i(2), \cdots, z_i(N)]$ is proposed to be the key parameter for channel vector $\mathbf{h}_i$.

4. Experiments Results and Simulations

A Experiments on map generation

To verify the accuracy of proposed algorithm proposed in part ,III, we conduct the experiment in A) a part of the ground floor of Liwenzheng library ($44\text{m} \times 44\text{m}$), Southeast University B)a typical classroom of size $8\text{m} \times 8\text{m}$. The 2D plan of those two scenes are plotted in Fig.10 and Fig.11 , respectively, where the necessary size are also labelled. We implement the algorithm by programing on Android Studio based on 20 single user's track information and bluetooth device of version 4.0 is also equipped in four corners. The indoor map after integration are depicted in Fig.12 (for scene A) and Fig.13 (for scene B), where:

- Blue solid lines represent integrated track.
- Red dots represent locating points corrected by bluetooth devices.

We evaluate the mean error of generated map by comparing the integrated track (blue solid line in Fig12, Fig .13) with standard track (which is shown via red dotted line in Fig.12, Fig.13) and the error with respect to sampling points are given in Fig.14.

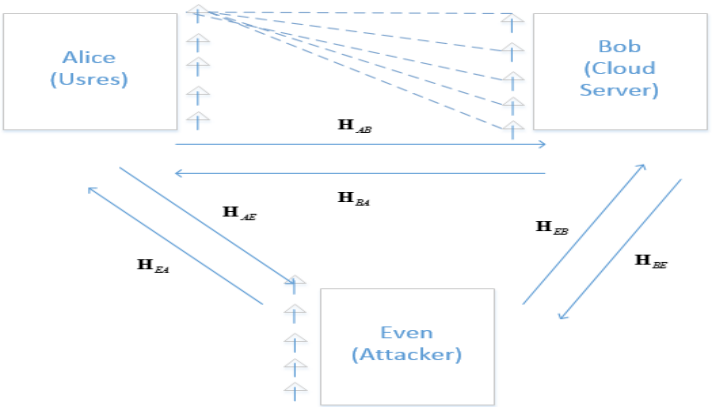


Fig. 9. Sketch of wireless channel measurement

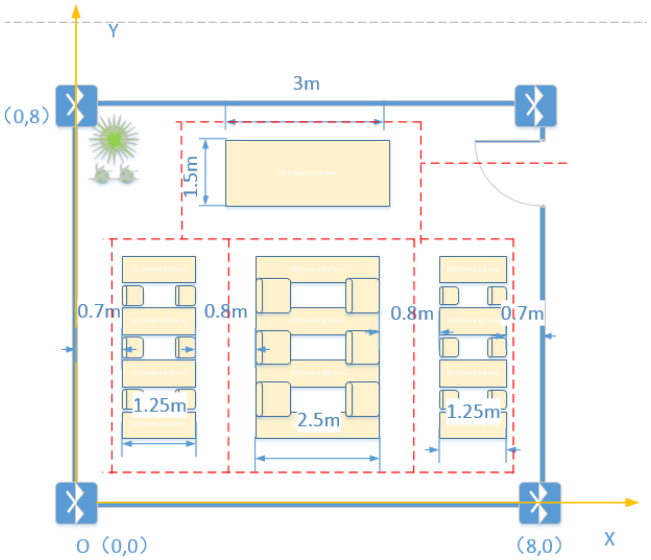


Fig. 10. A: a typical classroom

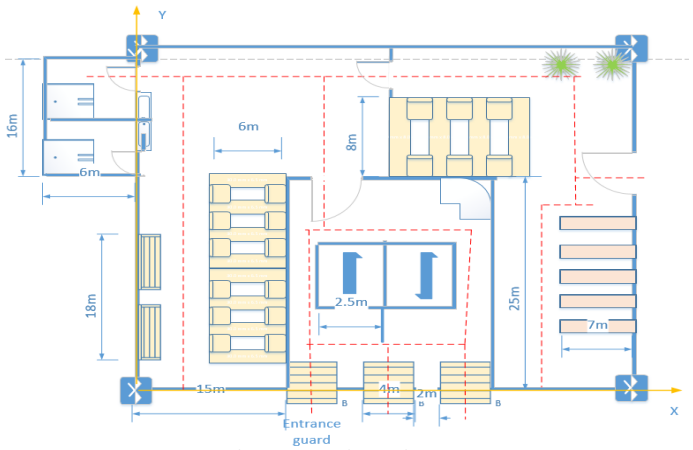


Fig. 11. B: a part of the ground floor of Liwenzheng library

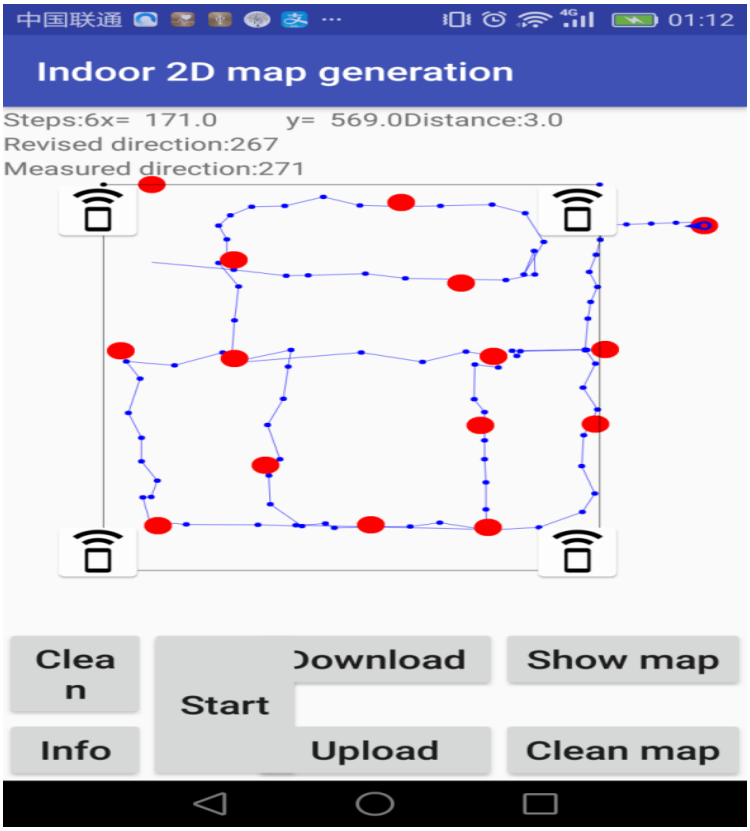


Fig. 12. Generated 2D map for scene A

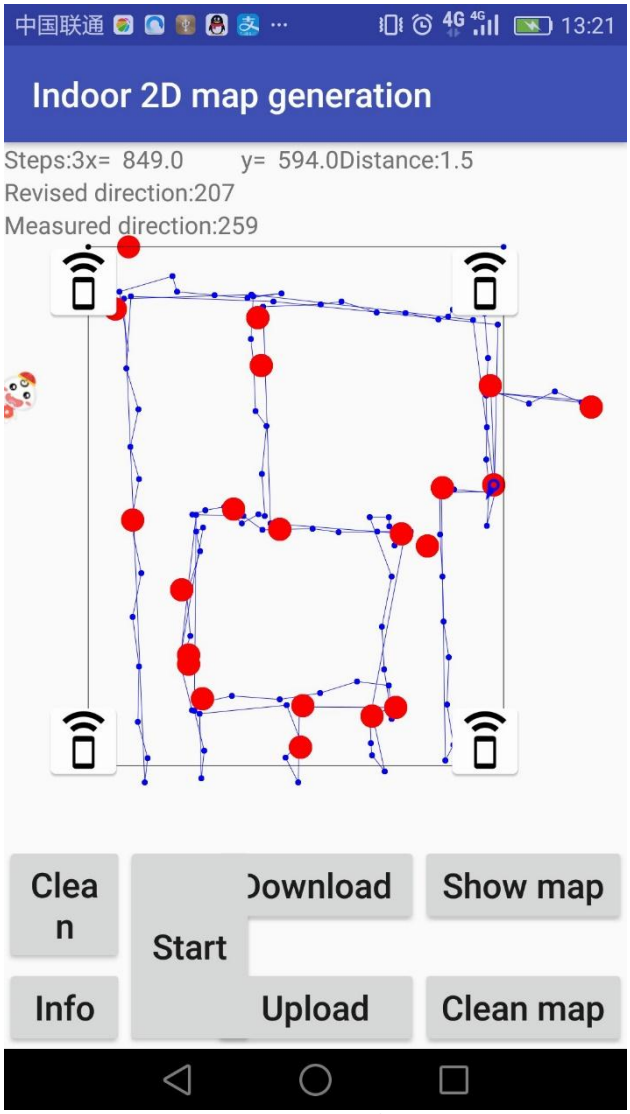


Fig. 13. Generated 2D map for scene B

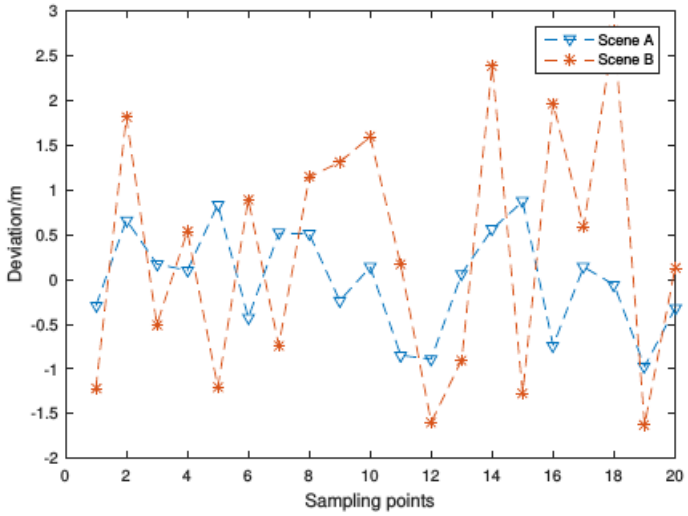


Fig. 14. Deviation with respect to sampling points

B Experiments on track data encryption

In [18][20], channel measurement is implemented by using Crossbow TelosB wireless sensors. In our paper, we adopt Android based cell phones with built-in WIFI (following IEEE 802.11 protocol) instead, for which no extra hardware devices are needed. Our experiment parameters are listed in Table.1, which includes 4 different users, 1 fixed cloud server and 1 fixed attacker. For a classic Bluetooth device, the working frequency ranges from 2.4MHZ to 2.4835MHZ, we then assume $T_s = \frac{1}{16}s$, which satisfies (32). We program using Android Studio to accomplish WIFI measurement supported by WIFI Manager and the app interface is given in Fig.15, where MAC address is also detected. The measured RSS values (dBm) detected by the cloud server and the attacker are shown in Fig.16 and Fig.17 ,respectively. In addition, the RSS value after decorrelation transformation is also

shown in Fig.18 .

From Fig.16 and Fig.17, it is easy to show the obvious difference between the channel measured by the cloud and eavesdropper due to the uncertainty of user's track, which means the eavesdropper is unable to obtain effective channel statue information (CSI) for key generation.

In our proposed MAQ scheme, the n-dimensional logical function mentioned in (39) is chosen to be:

$$f(\mathbf{x}) = (\sum_{k=1}^{N} k\,\mathbf{x}[k])\text{mod}2 \quad (44)$$

and 4 binary bits are intercepted after normalization. Table 2 represents the quantization results, which concludes $k(i,k)$, $e(k(i,k))$ and possible codeword d_0, d_1 for each given range of $\mathbf{y}''_i[k]$. The transmitted information $e(k(i,k))$ won't leak the actual codeword used to determine key parameters, as there exsits four equally likely codewords for both $e(k(i,k)) = 1$ and $e(k(i,k)) = 0$.

To represent the encrypted input sequence, we intercept the 4th and 64th time point in Fig.18, which results: $(\alpha, \beta, \theta_C, \theta_M) = (0,0.5,0.75,0.5)$ and $(\alpha, \beta, \theta_C, \theta_M) = (0,0.75,0.5,0.5)$. We randomly choose the transition probability of one grid as the input of unlinear system designed in Fig.8, Fig.19 and Fig.20 shows the output sequence $x_1(n), x_2(n), x_3(n)$. In both 4th time point and 64 time point, the transmitted over the wireless channel shows obviuos difference compared with input sequence $s(k)$, which guarantee the security of communication between users and cloud servers.

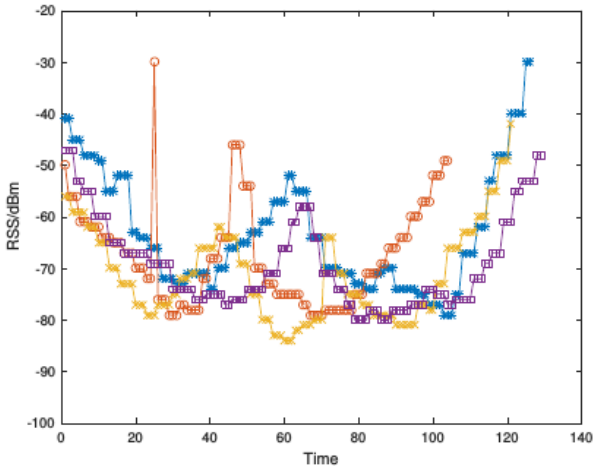


Fig. 16. RSS measurement value detected in the cloud server, *- for user1,+- for user2, Q- for user3, o- for user4.

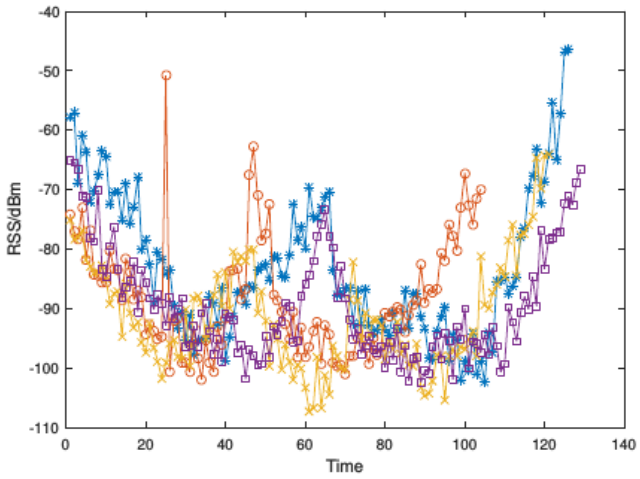


Fig. 17. RSS measurement value detected in the attacker, *- for user1, +- for user2, Q- for user3, o- for user4

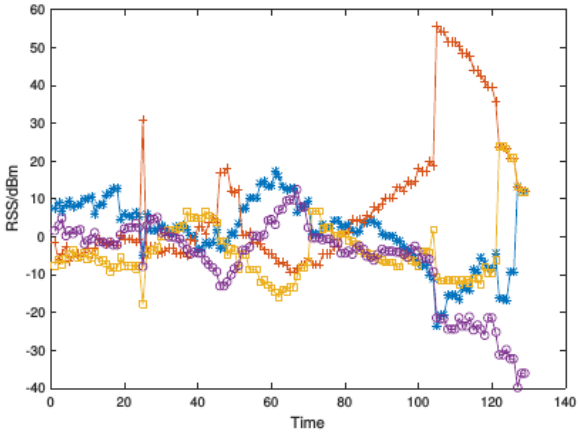
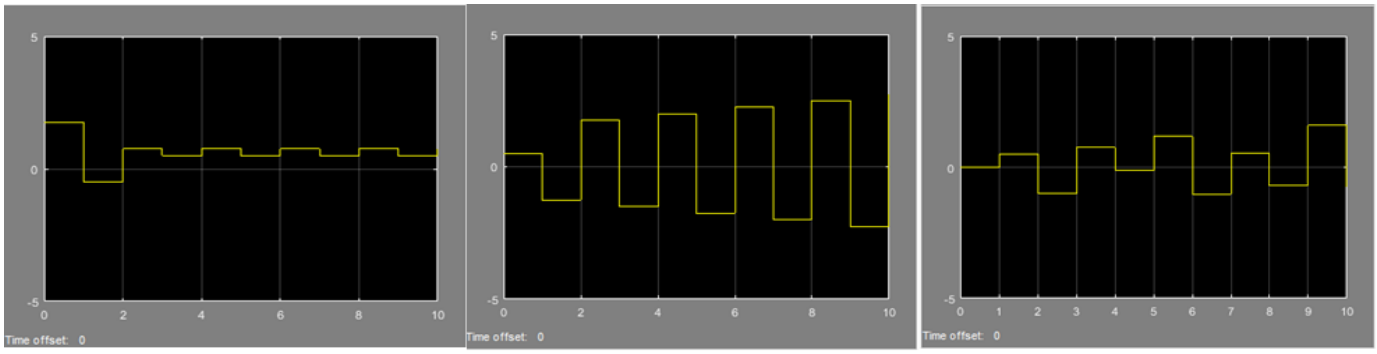
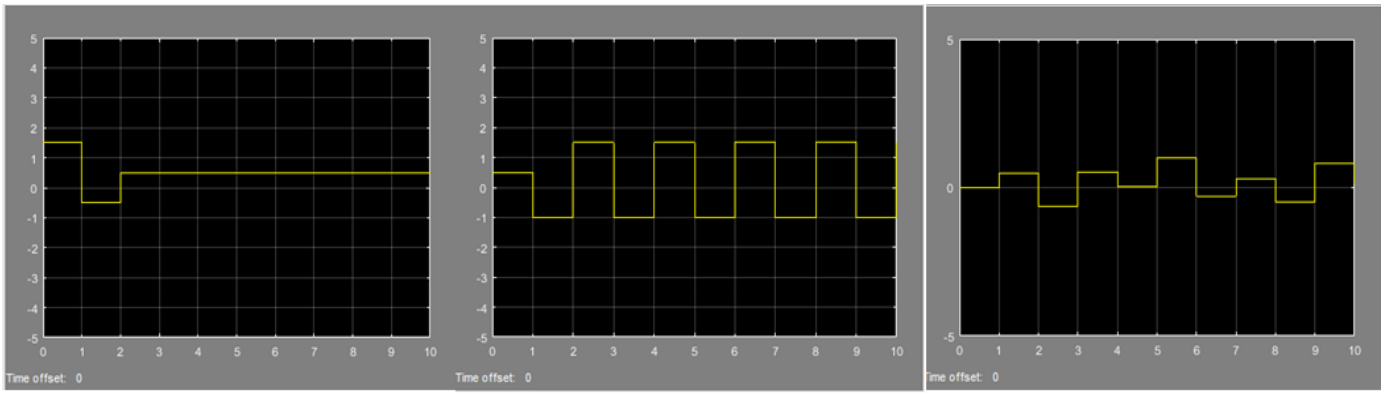


Fig.18 RSS measurement value after decorrelation transformation, *- for user1, +- for user2, Q- for user3, o- for user4.



$x_1(n), n \geq 0$ $x_2(n), n \geq 0$ $x_3(n), n \geq 0$

Fig. 19. Output sequence after the procession of dsp structure in Fig.8, where $\alpha = 0, \beta = 0.5, \theta_C = 0.75, \theta_M = 0.5$



$x_1(n), n \geq 0$ $x_2(n), n \geq 0$ $x_3(n), n \geq 0$

Fig. 20. Output sequence after the procession of dsp structure in Fig.8, where $\alpha = 0, \beta = 0.75, \theta_C = 0.5, \theta_M = 0.5$

Performance Analysis: Bit Error Probability

We evaluate the performance of proposed MAQ scheme by Bit Error Probability (P_E), that is, the conditional probability of node a and node b encode a bit differently. Note that when $\mathbf{y}''_{a,i}[k]$ is given, $k_a(i,k)$ can be calculated uniquely, then node b obtains the same codeword as a if and only if $k_a(i,k) = k_b(i,k)$, with the definition explained in (41), $P_{E,\mathbf{y}''_{a,i}[k]}$ can be expressed as:

$$P_{E,\mathbf{y}''_{a,i}[k]} = 1 - P(\text{Codeword}_a = \text{Codeword}_b | \mathbf{y}_{a,i}[k]) \quad (45)$$

where $\mathbf{d}_{i,k} = \{b_1, b_2, \cdots, b_n\}$ is the n-dimensional vector determined by $\mathbf{y}''_{b,i}[k] = 0.\overline{b_1 b_2 \cdots b_n}$. Taking the case when $\mathbf{y}''_{a,i}[k] \in (0.5, 0.5625)$ as an example, then we have $k_a(i,k) = 7$ and $\text{Codeword}_a = 11$ according to Table.2. Thus (45) can be rewritten as:

$$\begin{aligned} P_{E,\mathbf{y}''_{a,i}[k]} &= 1 - P(\mathbf{y}''_{b,i}[k] \in (0.125, 0.1875)) \\ &\quad - P(\mathbf{y}''_{b,i}[k] \in (0.1875, 0.25)) \\ &\quad - P(\mathbf{y}''_{b,i}[k] \in (0.5, 0.5625)) \\ &\quad - P(\mathbf{y}''_{b,i}[k] \in (0.5625, 0.625)) \\ &\quad - P(\mathbf{y}''_{b,i}[k] \in (0.875, 0.9375)) \end{aligned} \quad (46)$$

For the case when the channel variable $(\mathbf{y}''_{a,i}[k], \mathbf{y}''_{b,i}[k])$ obeys jointly Gussian distribution, i.e., $\mathbf{y}''_{b,i}[k] \sim \mathcal{CN}(0, \sigma^2)$, the cumulative distribution function (CDF) can be expressed as $\Phi(x) \triangleq \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{\frac{x^2}{2\sigma^2}} dx$, which results:

$$\begin{aligned} P_{E,\mathbf{y}''_{a,i}[k]} &= 1 - \Phi(0.25) - \Phi(0.625) - \Phi(0.9375) \\ &\quad + \Phi(0.125) + \Phi(0.5) + \Phi(0.875) \end{aligned} \quad (47)$$

With total probability formula, the total error probability for n-bit MAQ scheme can be deduced by:

$$P_E = \sum_{\mathbf{y}''_{a,i}[k] \in (j/2^n, (j+1)/2^n)} \left(\Phi\left(\frac{j+1}{2^n}\right) - \Phi\left(\frac{j}{2^n}\right) \right) P_{E,\mathbf{y}''_{a,i}[k]} \quad (48)$$

Table.3 shows the relationship between P_E and quantization bits n, where one can see the significant negative correlation between P_E and n, however, low P_E is also at the cost of high time complexity.

Table I.Experimental Scene Setting

Individual	Moving Speed	Initial Location
Cloud Server	0	(0, 0)
Attacker	0	(0, 15)
User1	0.5m/s	(6, 0)
User2	0.5m/s	(6, 0)
User3	1m/s	(6, 0)
User4	1m/s	(6, 0)

table ii.multibit adaptive quantization for n = 4-bit

Bin $k(i, k)$	Codeword d_0d_1	$e(k(i, k))$	Range of $y_i''[k]$
2	0000	1	(0, 0.0625)
6	0101	1	(0.0625, 0.125)
9	1111	0	(0.125, 0.1875)
12	1110	0	(0.1875, 0.25)
8	0111	0	(0.25, 0.3125)
11	1110	1	(0.3125, 0.375)
5	0101	0	(0.375, 0.4375)
16	1000	0	(0.4375, 0.5)
7	0111	1	(0.5, 0.5625)
10	1111	1	(0.5625, 0.625)
5	0101	1	(0.625, 0.6875)
16	1000	0	(0.6875, 0.75)
5	0101	1	(0.75, 0.8125)
16	1000	0	(0.8125, 0.875)
13	1111	0	(0.875, 0.9375)
16	1000	0	(0.9375, 1)

[1] Notation: For m = 4 bit quantization, we have $t_{16} = 16$, $t_{15} = 8$, $t_{14} = 12$, $t_{13} = 14$, $t_{12} = 7$, $t_{11} = 11$, $t_{10} = 13$, $t_9 = 4$, $t_8 = 6$, $t_7 = 10$, $t_6 = 3$, $t_5 = 14$, $t_4 = 5$, $t_3 = 9$, $t_2 = 2$, $t_1 = 1$.

table iii.bit error probability with respect to quantization bits n ($\sigma = 1$)

Quantization bits n	2	4	6	8
P_E	0.7549	0.3073	0.1002	0.0305
Quantization bits n	10	12	14	16
P_E	0.0091	0.0027	8.0392×10^{-4}	2.3834×10^{-4}

6. Conclusion

This paper proposes and implements a complete scheme for indoor map and track data encryption based on mobile crowdsourcing. For map generation, our main contribution can be summarized as: 1)we establish the maximum probability track generation algorithm by employing feature recognition and dynamic planning, where the upper bound of time complexity is also deduced theoretically. 2)we design a deviation correction method based on bluetooth signal, which is of no dependence on satellite and wifi. To transmit track data safely, we propose a novel encryption scheme by applying chaos in a nonlinear system, where the stability condition are also discussed. The key parameters are designed in a framework of channel measurement and multibit adaptive quantization to defend the attack of eavesdroppers, where we show the proof of an important lemma to ensure maximum secure bits. Experimentally, we program using Android Studio to verify the effectiveness of map generation by testing in several scenes, which is confirmed to have satisfactory accuracy. We report the numerical result of track encryption in detail and furtherly evaluate the performance of MAQ scheme via bit error probability.

Acknowledgement

We would like to express our appreciation for the assistance of associate professor Yubo Song for his suggestions and revison of our paper.
The experiment is also partially supported by Jiapeng Hu and Xi Chen, form the School of Electrical Engineering,

References

[1] K. Liu, G. Motta, and T. M, "Navigation services for indoor and outdoor user mobility: An overview," in *2016 9th International Conference on Service Science (ICSS), Chongqing, 2016, pp. 74-81*, pp. 74–81, 2016.

[2] X. Hai, X. Li, and C. Pan, "Compatibility study between CDR and aeronautical radio navigation service ils/vor," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, 2017.

[3] D. V. G. K. I. Dyuldina, S. I. Snopko, K. J. Shakhgeldyan, and V. V. Kryukov, "Indoor navigation service based on Wi-Fi positioning,," in *2017 Second Russia and Pacific Conference on Computer Technology and Applications (RPC)*, pp. 68–71, 2017.

[4] K. Liu, G. Motta, and T. Ma, "Navigation services for indoor and out- door user mobility: An overview," in *2016 9th International Conference on Service Science (ICSS)*, pp. 74–81, 2016.

[5] M. H. E. W. T. Sadhu, A. B. Albu and B. Wyvill, "Obstacle Detection for Image-Guided Surface Water Navigation," in *2016 13th Conference on Computer and Robot Vision (CRV), Victoria, BC*, pp. 45–52, 2016.

[6] T. A. Teo and C. Yu, "Three-dimensional positioning using ALOS/Prism triple linear-array satellite images," in *2017 IEEE 2nd International Con- ference on Signal and Image Processing (ICSIP), Singapore*, pp. 232– 236, 2017.

[7] M. E. Gorbunov, "Three-dimensional satellite refractive tomography of the atmosphere: Numerical simulation," *Radio Science*, vol. 31, pp. 95– 104, Jan.-Feb 1996.

[8] S. Spira, M. Schneider, T. Welker, J. Mller, and M. A. Hein, "Compact three-dimensional four-way vectorial steering module for ka-band multiple feeds-per-beam satellite payload applications," in *2016 IEEE MTT-S International Microwave Symposium (IMS), San Francisco, CA*, pp. 1–4, 2016.

[9] X. Du, K. Yang, and D. Zhou, "Mapsense: Mitigating Inconsistent WiFi Signals using Signal Patterns and Pathway Map for Indoor Positioning," *IEEE Internet of Things Journal*, vol. 99, no. 99, pp. 1–1, 2018.

[10] K. Nguyen-Huu, K. Lee, and S. W. Lee, "An indoor positioning system using pedestrian dead reckoning with WiFi and map-matching aided," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo*, pp. 1–8, 2017.

[11] S. Bhattacharjee, N. Ghosh, V. K. Shah, and S. K. Das, "QnQ: A reputation model to secure mobile crowdsourcing applications from incentive losses," in *2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV*, pp. 1–9, 2017.

[12] Z. Chi, Y. Wang, Y. Huang, and X. Tong, "The Novel Location privacy- preserving CKD for Mobile Crowdsourcing Systems,"

IEEE Access, vol. 6, pp. 5678–5687, 2017
[13] Y. Miao, X. L. J. Ma, Z. L. X. Li, and H. Li, "Practical Attribute-Based Multi-keyword search scheme in Mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.
[14] J. Rivas, R. Wunderlich, and S. J. Heinen, "An integrated acceleration sensor for traffic condition detection," in *Proceedings of 2012 9th IEEE International Conference on Networking, Sensing and Control, Beijing*, pp. 127–132, 2012
[15] L. Shan, C. Yang, W. Xu, and M. Zhang, "Heterogeneous acceleration for CNN training with many integrated core," in *2017 IEEE International Conference on Signal Processing, Communications and Computing(ICSPCC), Xiamen*, pp. 1–6, 2017.
[16] R. Pich, S. Chivapreecha, and J. Prabnasak, "A new key generator for data encryption using chaos in digital filter," in *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam*, pp. 87–92, 2017.
[17] Y. Umezawa, M. Dobashi, H. Kamata, and T. Endo, "Chaos signal generator by iir digital filters including nonlinear functions and its appli- cation," in *Proceedings KES '98. 1998 Second International Conference on, Adelaide, SA*, pp. 169–175, 1998.
[18] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measure- ments," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 17–30, Jan 2017.
[19] S. H. K. H. et.al, "Evaluation of denoising performance indices for noisy partial discharge signal based on DWT technique," in *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, pp. 392– 397, 2017.
[20] A. Limmanee and W. Henkel, "Secure physical-layer key generation protocol and key encoding in wireless communications," in *2010 IEEE Globecom Workshops, Miami, FL*, pp. 94–98, 2010

APPENDIX A
PROOF OF LEMMA 1

To deduce the recurrence formula mentioned in (11), we transform this problem into a combinatorial counting, which is depicted in Fig.10. Assume $i|k, j|k$, $i-j \equiv 1 \pmod 2$, the value of $s(i,j)$ equals to the number of total possible ways from $(0,0)$ to (i,j). With s moves ($s \le \min\{i/k, j/k\}$) from (i',j') to $(i'+k, j'+k)$ and altogether $\frac{i+j}{k}$ moves required, the total number of moves from (i',j') to $(i'+k,j')$ or $(i',j'+k)$ can be expressed as $\frac{i+j}{k}-s$, which yields:

$$\begin{aligned} s(i,j) &= \beta^{\frac{i+j}{k}} \sum_{s=0}^{\min\{\frac{i}{k},\frac{j}{k}\}} \sum_{r=0}^{\frac{i+j}{k}-2s} \binom{s}{\frac{i+j}{k}-s} \binom{r}{\frac{i+j}{k}-2s} \alpha^r \beta^{\frac{i+j}{k}-2s-r} \gamma^s \\ &= \beta^{\frac{i+j}{k}} \sum_{s=0}^{\min\{\frac{i}{k},\frac{j}{k}\}} \binom{s}{\frac{i+j}{k}-s} \sum_{r=0}^{\frac{i+j}{k}-2s} \binom{r}{\frac{i+j}{k}-2s} (\tfrac{\alpha}{\beta})^r \\ &= \beta^{\frac{i+j}{k}} \sum_{s=0}^{\min\{\frac{i}{k},\frac{j}{k}\}} \binom{s}{\frac{i+j}{k}-s} (1+\tfrac{\alpha}{\beta})^{\frac{i+j}{k}-2s} \end{aligned} \quad (49)$$

where combinatorial number $\binom{m}{n} = \frac{n!}{m!(n-m)!}$. Define $\frac{i+j}{k} = 2b+1$, $m = (1+\frac{\alpha}{\beta})^{-2}$, from (49) we can furtherly obtain:

$$s(i,j) \le (\alpha+\beta)^{\frac{i+j}{k}} \underbrace{\sum_{s=0}^{b} \binom{s}{2b+1-s} m^s}_{a_b} \quad (50)$$

$$\begin{aligned} a_b &= 1 + \sum_{s=1}^{b} [\binom{s}{2b-s} + \binom{s-1}{2b-s}] m^s \\ &= \sum_{s=0}^{b} \binom{s}{2b-s} m^s + \sum_{s=1}^{b} \binom{s-1}{2b-s} m^s \\ &= \sum_{s=0}^{b} \binom{s}{2b-s} m^s + m \sum_{s=0}^{b-1} \binom{s}{2(b-1)+1-s} m^s \\ &= \sum_{s=0}^{b} \binom{s}{2b-s} m^s + m a_{b-1} \\ &\quad \underbrace{\quad}_{c_b} \end{aligned} \quad (51)$$

Where the first equation comes from the well-known conclusion $\binom{m}{n+1} = \binom{m}{n} + \binom{m-1}{n}$. On the other side, c_b can be rewritten as:

$$\begin{aligned} & a_b - m a_{b-1} \\ &= c_b \\ &= 1 + m^b + \sum_{s=1}^{b-1} \binom{s}{2b-s} m^s \\ &= \sum_{s=0}^{b-1} \binom{s}{2b-s-1} m^s + \sum_{s=1}^{b} \binom{s-1}{2b-s-1} m^s \\ &= \sum_{s=0}^{b-1} \binom{s}{2b-s-1} m^s + m \sum_{s=0}^{b-1} \binom{s}{2(b-1)-s} m^s \\ &= m c_{b-1} + a_{b-1} \end{aligned} \quad (52)$$

Combine (50) (51) (52), it can be obtained that:

$$\begin{cases} c_b = a_b - m a_{b-1} \\ c_b = m c_{b-1} + a_{b-1} \end{cases} \quad (53)$$

(53) implicts:

$$a_b = (2m+1) a_{b-1} - m^2 a_{b-2} \quad (54)$$

which completes the proof.

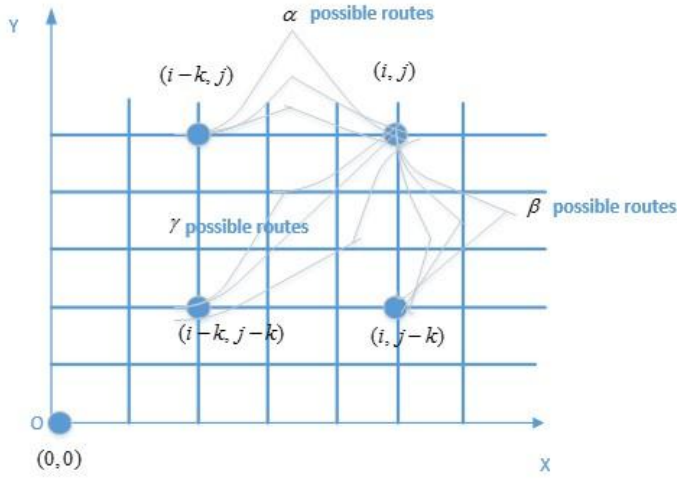


Fig. 21. Sketch of the equivalent combinatorial problem

APPENDIX B
PROOF OF LEMMA 2

We start by using the conclusion deduced in [20], i.e., when channel vectors are assumed to be correlated zero-mean complex Gaussian vector, the bits available for generating secury bits are given by:

$$\begin{aligned} I_K &= \log_2(\pi e)^N \det(\mathbf{R}_a + \sigma_1^2 \mathbf{I}) \\ &\quad + \log_2(\pi e)^N \det(\mathbf{R}_a + \sigma_2^2 \mathbf{I}) \\ &\quad - \log_2(\pi e)^N \det(\mathbf{R}_{AC}) \end{aligned} \quad (55)$$

where:

$$\mathbf{R}_{AC} \triangleq \begin{bmatrix} \mathbf{R}_a + \sigma_1^2 \mathbf{I} & \mathbf{R}_a \\ \mathbf{R}_a & \mathbf{R}_a + \sigma_2^2 \mathbf{I} \end{bmatrix} \quad (56)$$

and σ_1, σ_2 denotes the covariances of noise at Alice node, Bob node, respectively. Simplifying $\mathbf{R}_{AC}$ results in:

$$\det(\mathbf{R}_{AC}) = \det(\mathbf{R}_a + \sigma_2^2 \mathbf{I}) \det(\mathbf{R}_a + \sigma_1^2 \mathbf{I} - \mathbf{R}_a (\mathbf{R}_a + \sigma_2^2 \mathbf{I})^{-1} \mathbf{R}_a) \quad (57)$$

Substitute (57) into (55), which yields:

$$I_K = \log_2(\pi e)^N \det^{-1}(\mathbf{I} - \mathbf{R}_a (\mathbf{R}_a + \sigma_2^2 \mathbf{I})^{-1} \mathbf{R}_a (\mathbf{R}_a + \sigma_1^2 \mathbf{I})) \quad (58)$$

Considering power restriction, the optimization problem can be formulated as:

$$\begin{aligned} &\min_{\mathbf{R}_a} f(\mathbf{R}_a) = \det(\mathbf{I} - \mathbf{R}_a (\mathbf{R}_a + \sigma_2^2 \mathbf{I})^{-1} \mathbf{R}_a (\mathbf{R}_a + \sigma_1^2 \mathbf{I})^{-1}) \\ &s.t. \operatorname{Tr}(\mathbf{R}_a) = c, \mathbf{R}_a \succcurlyeq \mathbf{0} \end{aligned} \quad (59)$$

We simplify the target function by using the first-order approximation $(\mathbf{I}+\mathbf{A})^{-1} \approx \mathbf{I} - \mathbf{A}$, i.e.,

$$\begin{aligned} f(\mathbf{R}_a) &= \det(\mathbf{I} - \mathbf{R}_a ((\sigma_1^2 + \sigma_2^2)\mathbf{I} + \mathbf{R}_a + \sigma_1^2 \sigma_2^2 \mathbf{R}_a^{-1})^{-1}) \\ &\approx \det((1 - \sigma_1^2 \sigma_2^2)\mathbf{I} - (\sigma_1^2 + \sigma_2^2)\mathbf{R}_a - \mathbf{R}_a^2) \end{aligned} \quad (60)$$

With (60), the lagrange function of problem (59) can be expressed as:

$$L(\mathbf{R}_a, \lambda) = f(\mathbf{R}_a) - \lambda(\operatorname{Tr}(\mathbf{R}_a) - c) \quad (61)$$

the gradient of $L(\mathbf{R}_a, \lambda)$ with respect to variable $\mathbf{R}_a$ is calculated by

$$\max_{\mathbf{R}_a} L(\mathbf{R}_a, \lambda) = \operatorname{Tr}(\mathbf{U}^{-1}(-(\sigma_1^2 + \sigma_2^2)\mathbf{I} - 2\mathbf{R}_a)) - \lambda \operatorname{Tr}\mathbf{I} \quad (62)$$

where (58) holds from the fact that $\frac{\partial \det \mathbf{W}}{\partial \mathbf{X}} = \operatorname{Tr}(\mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \mathbf{X}})$ and $\frac{\partial \operatorname{Tr}\mathbf{W}}{\partial \mathbf{W}} = N$. $\mathbf{U} = (1 - \sigma_1^2 \sigma_2^2)\mathbf{I} - (\sigma_1^2 + \sigma_2^2)\mathbf{R}_a - \mathbf{R}_a^2$. By letting $\nabla_{\mathbf{R}_a} L(\mathbf{R}_a, \lambda) = 0$, it is obtained that:

$$\mathbf{R}_a((\sigma_1^2 + \sigma_2^2 - 2)\mathbf{I} + \mathbf{R}_a) = (\lambda(1 - \sigma_1^2 \sigma_2^2) + (\sigma_1^2 + \sigma_2^2))\mathbf{I} \quad (63)$$

Let $\lambda_1, \lambda_2, \cdots, \lambda_N$ denote N eigenvalues of $\mathbf{R}_a$, from (59) it is easy to

verify:

$$\prod_{i=1}^{N}\lambda_i\prod_{i=1}^{N}(1+\frac{\lambda_i}{\sigma_1^2+\sigma_2^2-2})=\left[\frac{\lambda(1-\sigma_1^2\sigma_2^2)+(\sigma_1^2+\sigma_2^2)}{\sigma_1^2+\sigma_2^2-2}\right]^N \tag{64}$$

where $\sum_{i=1}^{N}\lambda_i=c$. By introducing parameter d and employing AM-GM inequality, the left-hand-side (LHS) of (60) can be rewritten as:

$$\begin{aligned}&\frac{1}{d^N}\prod_{i=1}^{N}(d\lambda_i)\prod_{i=1}^{N}(1+\frac{\lambda_i}{\sigma_1^2+\sigma_2^2-2})\\&\leq\frac{1}{d^N}(\frac{\sum_{i=1}^{N}d\lambda_i}{N})^N(\frac{N+\sum_{i=1}^{N}d\lambda_i/(\sigma_1^2+\sigma_2^2-2)}{N})^N\\&=(\frac{c}{N})^N(\frac{N+cd/(\sigma_1^2+\sigma_2^2-2)}{N})^N\end{aligned} \tag{65}$$

where the inequality holds if and only if $d=\frac{N}{c}(1+\frac{c}{N(\sigma_1^2+\sigma_2^2-2)}$, $\lambda_1=\lambda_2=\cdots=\lambda_N=c/N$, λ satisfies $\frac{\lambda(1-\sigma_1^2\sigma_2^2)+(\sigma_1^2+\sigma_2^2)}{\sigma_1^2+\sigma_2^2-2}=\frac{c}{N}\cdot\frac{N+cd/(\sigma_1^2+\sigma_2^2-2)}{N}$.