マンハッタン距離を用いた 幾何学的グループ鍵共有法に関する検討

西上 弘毅[†1]　岩村 惠市[†1]

概要：近年、モノ同士で相互に通信を行う IoT ネットワークが普及している。IoT ネットワークにおいて暗号通信を行う場合、あるグループに属す全ノードが同じ暗号化鍵を所有しその鍵を用いて暗号化を行うことが最も効率的である。しかしその場合、一つのノードが乗っ取られ鍵が漏洩すると全ての情報が漏洩する為、IoT ノードの排除を含むグループ構成の変更が用意なグループ鍵共有法が必要である。それに対して、各ノードが固有の鍵を持ち、その鍵から座標点を生成して全ノードから等距離な点を幾何学的に求め、その距離をグループ鍵とする方式が濱崎・金子らによって提案された。この方式は最小の通信量で容易にそのグループ構成を変えることが出来る。しかし、濱崎・金子らの方式はユークリッド距離を用いている為、二乗計算を行う必要があり計算量に問題があった。本稿では、ユークリッド距離よりも計算量が少ないマンハッタン距離を用いた場合について検討する。

キーワード：グループ鍵共有、共通鍵暗号、IoT ネットワーク、マンハッタン距離

1. はじめに

近年、モノ同士で相互に通信を行う IoT ネットワークが普及している。IoT ネットワークは、ベースステーション（以下 BS）とノードの二種類の端末から構成されるネットワークである。BS は各ネットワークにつき一つだけ存在し、ネットワークの心臓部的な役割を持つ。対してノードは一つのネットワークに大量に存在し、各種の情報の収集、送信を行う。BS はネットワークの中心に当たる為、これが正しく機能しなくなるとネットワーク自体が崩壊する。よって、BS は耐タンパ性を持ち、電源容量も豊富であるという前提を置く。ノードはネットワークの性質上、物理的に安全でない場所に設置される事や、莫大な数の端末が存在する為に一つ一つのコストは低い事が望ましい為、耐タンパ性、CPU の性能は低く、電源容量も小さいとする。

以上の IoT ネットワークにおいて安全な通信を行う為には、全ての通信を暗号化する必要がある。しかし、前述したノードの性能の低さにより、IoT ネットワークにおける暗号通信においては、以下の５点が重要になる。特に、IoT における暗号通信においては暗号通信だけでなく、その前段階の鍵共有の効率化も重要である。

- 通信量の少なさ
  通信はノードにおける電力消費の負担が最も大きい為、ノードの電源容量を考えると少なくすることが望まれる。
- 計算量の少なさ
  ノードが持つ CPU は性能が低い為、公開鍵暗号のような複雑な計算は膨大な処理量になり、電力消費の観点から望ましくない。
- 記憶容量の少なさ
  鍵共有を行う際に、ノード毎の固有鍵でなく、多くの要素鍵を記憶し、その要素鍵の組み合わせによって鍵共有することもできるが、ノードの記憶容量も小さいものが多い為、ノードが記憶する情報は少ない方が望ましい。
- 鍵更新
  IoT ネットワークでは、通信を行うグループ毎に鍵を共有して暗号通信を行うことが効率的である。しかし、そのグループ鍵を固定すると、一つの鍵が漏洩すると全ての情報が漏洩する為、容易に鍵更新できることが望ましい。また、鍵を漏洩させたノード等を排除できる仕組みも必要である。
- 安全性
  用いる状況に応じた要件を満たした上で、安全な暗号通信を実現できる必要がある。

各ノードが固有の鍵を持ち、その鍵から座標点を生成し全ノードの座標点から等距離な中心点を求め、その中心点の座標をブロードキャストして、その点との距離を共通の鍵とする方式が濱崎・金子らによって提案された。この方式を前述した５種類の基準によって評価すると、計算量が多いという問題以外良い結果を示した。計算量が多い理由は、ノード数が多い場合、多次元空間上の座標点を計算する必要があり、かつ中心点と座標点との距離を計算する時ユークリッド距離を用いている為に２乗の計算が必要な為である。この問題を解決する為に、マンハッタン距離を用いることを検討する。マンハッタン距離は、座標の要素同士の差の絶対値の総和によって求められる為、２乗の計算を行う必要がない。ただし、ユークリッド距離では n 個の点に対する中心点の求め方は知られているが、マンハッタン距離において n 個の点に対する中心点の求め方は、著者などが調べた限り容易に求められる手法は知られていない。

†1 東京理科大学, 住所, 〒125-8585 東京都葛飾区 新宿 6-3-1.
Tokyo University of Science, 6-3-1, niijyuku katsushikaku, Tokyo 125-8585, Japan

よって、本稿では、マンハッタン距離での中心点の求め方について検討した結果を示す。

本論文の構成はまず２章において従来方式である幾何学的グループ鍵共有法の詳細について述べる。次に、３章にてマンハッタン距離を用いる方法についての検討を行う。４章では今回の提案方式を説明する。最後に前述した５種類について評価を行う。

2. 従来方式

従来方式として濱崎・金子らによって提案された幾何学的グループ鍵共有法について説明する。幾何学的グループ鍵共有法とは、各ノードが持つ固有の鍵から毎回新たな座標点を生成し、全ての点から等距離な中心点をブロードキャストすることで、その中心点との距離を共通の鍵とする方式である。

2.1 原理

従来方式では、以下の図１のような円を利用している。

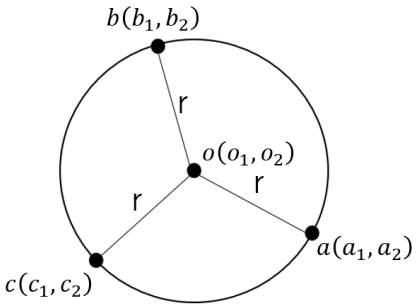

図 1　幾何学的性質

従来方式では円が持つ性質の内、以下の２つの性質を利用する。

① 円周上に存在する全ての点と中心点 O までのユークリッド距離は等しい。

② 同じ中心点を持つ円は無限に存在する。

従来方式では、円周上の点の座標を秘密に生成し、それらの中心点の座標のみを送信する。秘密に生成された円周上の点の座標を知っていれば、性質①より中心点とのユークリッド距離を計算すれば他のノードと等しい値を得る事が出来る。また、攻撃者が通信を盗聴して、中心点の座標を得たとしても性質②より、同じ中心点を持つ円は無限に存在する為、鍵の情報は漏洩しない。

図１の座標点 a、b、c と円の中心点 o のユークリッド距離は、以下の式で求められる。

$$\begin{aligned} r &= \sqrt{(o_1 - a_1)^2 + (o_2 - a_2)^2} \\ &= \sqrt{(o_1 - b_1)^2 + (o_2 - b_2)^2} \\ &= \sqrt{(o_1 - c_1)^2 + (o_2 - c_2)^2} \end{aligned} \quad (1)$$

ただし、2.2 で述べる通り、単に同じ値を共有することだけが目的の場合は、平方根の計算は行わない。

2.2 従来方式のアルゴリズム[1]

まず、BS での処理について説明する。前提として BS は全てのノードの固有鍵を知っているものとし、グループに参加するノードの総数は n 個とする。BS は以下の手順に従って鍵を共有する。

① 乱数 r を生成し、ノード数 m と一緒にブロードキャストする。

② 乱数 r と各ノードの固有鍵を疑似乱数生成器に入力し、n-1 次元のノード座標$a_i(a_{i1}, a_{i2}, \cdots a_{i(n-1)})$ $(i = 1,2,3, \cdots n)$を計算する。

③ この時、全てのノード座標が n-2 次元平面に含まれていた場合、①に戻り乱数 r を新たに生成して同じ手順を繰り返す。

④ ②で得られた全てのノード座標$a_i(a_{i1}, a_{i2}, \cdots a_{i(n-1)})$ $(i = 1,2,3, \cdots n)$を通る円を考え、その中心点をブロードキャストする。

次に、各ノードでの処理について説明する。一例として、i 番目のノードについて説明する。ノードは以下の手順に従って計算を行う。

① BS から受け取った乱数 r と自分の固有鍵から自分のノード座標$a_i(a_{i1}, a_{i2}, \cdots a_{i(n-1)})$を求める。

② BS から受け取った中心点の座標と自分のノード座標 a_iとのユークリッド距離を計算する。また、今回は全てのノードが同じ値を共有することが目的である為、平方根の計算を行う必要はない。よって鍵 S は以下の式(2)によって計算する。

$$S = (o_1 - a_{i1})^2 + (o_2 - a_{i2})^2 + \cdots + \left(o_{n-1} - a_{i(n-1)}\right)^2 \quad (2)$$

3. 基礎理論及び種々の検討

3.1 マンハッタン距離

マンハッタン距離とは、距離構造の一種であり、「２点の各要素の差の絶対値の総和」と定義されている。つまり、n 次元上の２点A$(a_1, a_2, \cdots, a_n)$、B$(b_1, b_2, \cdots, b_n)$のマンハッタン距離 L は以下の式(3)で表される。

$$L = |a_1 - b_1| + |a_2 - b_2| + \cdots + |a_n - b_n| \quad (3)$$

式(2)と比べると２乗の計算がなくなっていることが分かる。この性質によってノードの計算量を削減する事を目的とする。

3.1.1 マンハッタン距離における等距離な点の軌跡

図１のようにユークリッド距離では１つの点から等距離な点の軌跡は円になった。

それに対し、マンハッタン距離における等距離な点の軌跡は図２のように x,y 軸に対して 45 度傾いた正方形となる。

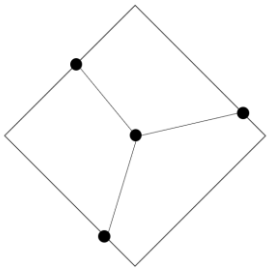

図 2　マンハッタン距離における等距離な点の軌跡

図２では、２次元の場合を例としたが、３次元では各軸に対して 45 度傾いた正八面体となり、それ以降は次元数に応じた多面体となっていく。

3.2 検討１

まず、マンハッタン距離における中心点を 45 度傾いた正方形の定義式から求める方法を考える。一般的に n 次元の正方形は以下の式で表される。

$$|x_1 - o_1| + |x_2 - o_2| + \cdots + |x_n - o_n| = S \qquad (4)$$

式(4)におけるo_1～o_nが中心点の座標、S が中心点と各座標x_1～x_nとのマンハッタン距離である。一般的に絶対値を含む計算を行うときには場合分けが必要である。例えば、2 次元において 3 点のノードを対象とした場合、以下の連立方程式を解く必要がある。

$$\begin{cases} |a_1 - o_1| + |a_2 - o_2| = S \\ |b_1 - o_1| + |b_2 - o_2| = S \\ |c_1 - o_1| + |c_2 - o_2| = S \end{cases}$$

上式では、中心点と各座標点の大小関係によって場合分けが必要がある。よって、この場合絶対値が 6 つ存在するので$2^6 = 64$通りに場合分けをする必要がある。

これを n 次元において考えると式(4)は、n 個の絶対値を持つ方程式となり、それを n+1 個連立させることになる。この為、場合分けの数は、$2^{n(n+1)}$通りになる。この n+1 はグループに参加するノード数に相当する為、例えば n=1000 とした場合は、$2^{1000\times1001}$となり、BS が十分な計算量を持つと仮定したとしても計算は困難であると考えられる。一般的に全数探索の限界は2^{80}通り程度だといわれている為、絶対値の場合分けに対しても、n=8 程度、即ち$2^{8\times9}$程度が限界と考えられ、それ以上大きな n に対しての計算は困難であると考えられる。

3.3 検討２

検討１より、絶対値を含む式を解く場合、n が大きくなると計算量が膨大になるという問題があった。そこで、絶対値による場合分けを必要としない方法を考える。この方法では、ノード座標によって分割される空間それぞれに中心点がある場合を計算する。例えば、2 次元での 3 点のノード A,B,C に対しては、図３のように空間は 16 分割される。

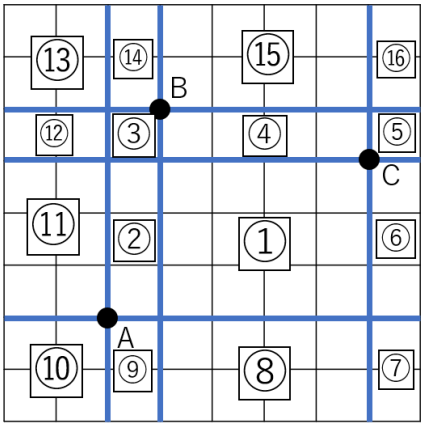

図 3　分割する範囲

図３に示すように p×p の 2 次元空間上にある３点のノード座標を$A(a_x, a_y)$、$B(b_x, b_y)$、$C(c_x, c_y)$とする。この場合、各ノードの座標によって 2 次元空間は x 座標では 0～a_x、a_x～b_x、b_x～c_x、c_x～p に、y 座標では 0～a_y、a_y～c_y、c_y～b_y、b_y～p に分割される。よって、例えば図３の①の範囲に中心点があるとすると、各座標点と中心点$O(o_x, o_y)$との関係は、$a_x < b_x \le o_x \le c_x$、$a_y \le o_y \le c_y < b_y$となる。この関係を用いると、以下のように式(4)の絶対値を用いずに、直接以下のように書くことが出来る。

$$S = o_x - a_x + o_y - a_y$$
$$S = o_x - b_x - o_y + b_y$$
$$S = -o_x + c_x - o_y + c_y$$

この場合、絶対値による場合分けでなく、ノード座標による空間分けとなり、計算量は分割される空間数に依存する。n 次元空間に n+1 点のノード座標があるとすると、それぞれの軸方向に n+2 個の範囲に分割される為、全部で$(n+2)^n$個の空間に分割される。この場合、n=15 としても2^{60}程度の空間分けとなり、計算可能な範囲であるといえる。よってこの手法では、n が 15、16 程度であれば計算可能であり、検討１に比べて効率化を実現できたことが分かる。ただし、n が 20 になると、$20^{21} \approx 2^{90}$となり計算が困難になると考えられる。

3.4 法の数による制限

ユークリッド距離では 2 次元上の 3 点は直線上にない限りその中心点を持ち、n 次元空間においても n+1 個の点が n-1 次元平面上になければ中心点を持つことが言える。3.2、3.3 では、マンハッタン距離による計算の仕方について検討したがそもそもマンハッタン距離において２次元上３点はどのような場合に中心点を持つか定かではない。例えば、図４に示すように 2 点を通る 45 度及び-45 度傾いた直線に囲まれた部分に 3 点目が存在する場合、正方形を描く事が出来ない。

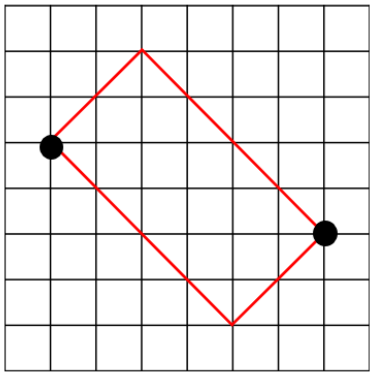

図 4　正方形を描くことが出来ない点配置の例

そこで、ある範囲において 3 点の組み合わせ全てについて正方形を描くことができるかを確認し、成功する確率を求めた。例えば、x,y の範囲を 0～2 とした場合、図 5 のような正方形を描く事が出来る。

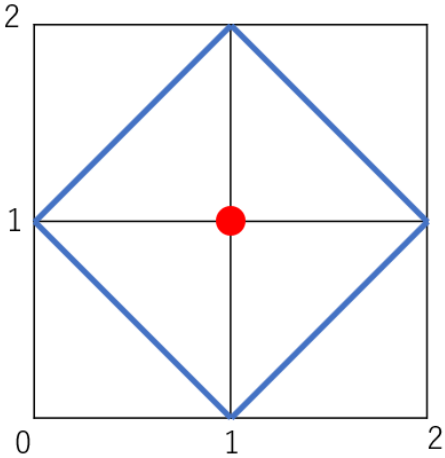

図 5　x,y の範囲が 0～2 の時における正方形の一例

図 5 の 45 度傾いた正方形は、{(0,1),(1,0),(2,1),(1,2)}の点の組合せによって構成されている。また、この他にも{(0,0),(0,2),(2,0),(2,2)}、{(0,2),(1,1),(2,2)}、{(0,2),(1,1),(0,0)}、{(0,0),(1,1),(0,2)}、{(2,2),(1,1),(0,2)}、{(0,2),(1,1),(0,2)}、{(0,0),(1,1),(2,2)}の組合せも 45 度傾いた正方形の点を表すと考える事が出来る。2 次元では 3 点が 45 度傾いた正方形上の点となる組合せの数は、上記の構成する点の中から 3 点選ぶ組合せの総和になるので、

$$ {}_4C_3 \times 2 + {}_3C_3 \times 6 = 14 $$

となる。x,y の範囲が 0～2 の時における全ての点の数は 9 個なので、3 点の組合せの総数は、$ {}_9C_3 = 84$となるので、法の数が 3 の場合の正方形を描く事が出来る確率は、

$$ \frac{14}{84} \approx 0.1667 $$

となる。以上のように x,y の範囲ごとに成功確率を求めた結果を以下の表 1 に示す。

表 1　x,y の範囲と正方形を描く事が出来る確率

x,y の範囲	正方形を描く事が出来る確率[%]
0～2	16.67
0～4	21.39
0～6	21.73
0～10	21.15
0～12	20.83
0～16	20.34

表 1 より、0～6 を超えると x,y の範囲が大きくなるほど確率は低くなっている。つまり、x,y の範囲を増加させると、全ての 3 点の組み合わせの増加に対して、45 度傾いた正方形を構成できる 3 点の組み合わせの増加は小さくなり、大きな範囲を設定する場合、正方形を構成する 3 点の組合せが存在する確率は低下する。

4.　提案方式

以上の事を踏まえ、法の数が小さい範囲で鍵共有を繰り返し、生成した鍵をつなげることで十分な長さにする方式を提案する。ここでは簡単のため 2 次元空間を考える。

4.1　提案方式のアルゴリズム

2.2 と同様に、BS の処理から説明する。また、前提として、各ノードはそれぞれ自分の固有鍵を持っていて、BS は全ノードの固有鍵を知っているとする。また、各ノードには$ID_i(i = 1,2,\cdots n)$が定められており、昇順に並べる事が出来るとする。2.2 と同様にノードの数は n とする。

① ID_iを昇順に並べ、3 つずつグループに分ける。各グループを昇順に$g_j(j = 1,2,\cdots m)$とする。

② 乱数r_jとノードID_iの固有鍵を疑似乱数生成器に入力し、ノード座標$a_{j,i,h}(h = x, y)$を計算する。

③ g_j毎にマンハッタン距離による 3 つのノードの中心点$O_{j,h}(h = x, y)$を求め、その点との距離を計算する。この時の距離を$s_j(j = 1,2,\cdots m)$とする。

④ グループ鍵$S_l(l = 1)$を生成し、$s_j(j = 1,2,\cdots m)$との差分$d_{j,l}$を計算する。

⑤ s_jでグループ番号jとlの和を暗号化し、差分$d_{j,l}$と排他的論理和し、$c_j(i = 1,2,\cdots,m)$とする。

⑥ 以下の情報をブロードキャストする。ただし、gID_jはグループg_jにおける最も小さなIDとする。

$$ (gID_j, r_j, O_{j,x}, O_{j,y}, c_j)(j = 1,2,\cdots,m) $$

⑦ r_jを変更し、①～⑥の操作を$S_l(l = 2,3,\cdots,t)$について繰り返し、$S_l(l = 1,2,,\cdots,t)$を連接したものをグループ鍵 S とする。

次にノードでの処理について説明する。

① BS から受け取ったグループ番号の内、自分の ID がgID_j～gID_{j+1}の間に入っていた場合、自分はグループjだと判断する。

② 受信したr_jから自分のノード座標$a_{j,i,h}(h = x, y)$を生成し、$O_{j,x}, O_{j,y}$とのマンハッタン距離を計算し、s_jを求

める。

③ s_jで$j+l$を暗号化し、それと$c_{j,l}$の EXOR から差分$d_{j,l}$を計算し、s_jと$d_{j,l}$から$S_l(l=1,2,,\cdots,t)$を求め連接してグループ鍵 S とする。

5. 評価

グループ鍵共有法では、通信量、ノードの計算量、安全性、記憶容量、鍵更新の５種類にて評価する。また、評価の方法として従来方式と比較することとする。前提として、グループに参加するノードの数は n とした場合について考える。

5.1 通信量

まず、通信量について比較する。鍵共有では１回に必要な通信量は少ない方が望ましい。通信量が増えると、その分だけ電力消費が高まる為である。

今回は、ノードの固有鍵など定数として表される値の長さを L1、エンコードされた値の長さを L2 とする。L1 と L2 の大きさはL1 < L2である。また、3.4 で述べた x,y の範囲に依存する長さを L3 とする。例えば、一番成功率が高い 0～6 の範囲を用いたならば、0～6 の数字は全て 3 ビットで表すことが出来る為、L3 の長さは 3 ビットとなる。

まず、従来方式について考える。従来方式において送信される情報は、乱数 r、参加するノード数 n、中心点の座標の３つである。乱数 r とノード数 n は共に定数なので長さは L1、中心点の座標は疑似乱数生成器でエンコードされた値と同じ長さなので、長さは L2 となる。乱数 r とノード数 n は１つずつ送信され、中心点の座標は n-1 個生成される。よって従来方式の通信量は$2L1+(n-1)L2$となる。

次に、提案方式について考える。提案方式にて送信される情報は、グループ番号gID_j、乱数r_j、各グループの中心点の座標$O_{j,h}(h=x,y)$、c_jの４つである。グループ番号gID_jの長さは、定数なので$L1$となる。乱数r_jと各グループの中心点の座標$O_{j,h}(h=x,y)$は、用いる x,y の範囲に依存して大きさが決まる為、$L3$ となる。最後に、c_jの長さは暗号化された値なので、$L2$ となる。以上の４つは各グループに１つずつ存在するので、１度のブロードキャストで送信されるデータの長さは、$m(L1+L2+3L3)$となる。そして、鍵長が十分な長さになるまで操作を繰り返すので、十分な長さになるまでの回数を t とすると、提案方式の通信量は$mt(L1+L2+3L3)$となる。t は $L3$ を 3 ビット、鍵 S のビット長を 129 ビットとすると、t=43 になる。ノードを３つずつのグループに分ける為、グループ数 m とノード数 n はm $= n/3$となり、通信量の合計は、$tn(L1+L2+3L3)/3$となる。それぞれの通信量の比較を示した表を以下の表 2 に示す。

表 2　通信量の比較

従来方式	提案方式
$2L1+(n-1)L2$	$tn(L1+L2+3L3)/3$

表１より比較すると、通信量はs_jなどを$tn/3$回送信しなければならない為、従来方式の方が少なくなる。

5.2 ノードの計算量

次に、ノードの計算量について比較する。ノードはネットワーク内に大量に存在する。その為、１つ１つのコストは低い方が望ましく、ノードの CPU の性能は低く、メモリ及び電源容量は小さくなる。以上の理由から、鍵共有を行う際、ノードが行う計算量は少ない方が望ましい。

ここでは、スカラ量の加算、乗算１回の計算量を C1、疑似乱数生成や共通鍵暗号の計算に必要な計算量を C2 とする。C1 と C2 の大きさの関係は、C1 < C2である。

まず従来方式について考える。従来方式では、各ノードは n-1 回の疑似乱数生成と、式(2)を用いてグループ鍵を計算する。よって、疑似乱数生成に必要な計算量は、$(n-1)C2$となる。式(1)より、n-1 回の２乗の計算と減算、n-2 回の加算を行う。よってグループ鍵の計算に必要な計算量は$(3n-5)C1+(n-1)C2$となる。

次に提案方式について考える。提案方式では、各ノードは 2 回の疑似乱数生成と、s_jと$d_{j,l}$の計算、$j+l$の暗号化を行う。s_jの計算には式(4)の 2 次元までの式を解く為、3C1 の計算量が必要となる。$d_{j,l}$の計算は、$j+l$の暗号化と１回の排他的論理和で計算される為、計算量は、C2+C1 となる。疑似乱数の生成にかかる計算量は提案方式と同じやり方で行う為、$2C2$となる。最後に、S_lを計算するために、$s_j+d_{j,l}$の計算を行う為、提案方式の計算量の合計は、$5C1+3C2$となる。しかし、鍵の長さが十分な長さになるまで繰り返す為、$t(5C1+3C2)$となる。それぞれの計算量を比較した表を以下の表 3 に示す。

表 3　計算量の比較

従来方式	提案方式
$(3n-5)C1+(n-1)C2$	$t(4C1+2C2)$

表２より、ノードの計算量では、提案方式＜従来方式となる。提案方式では次元数が２に限定される為、ノード数 n によって計算量は増えない。

5.3 安全性

次に安全性について考える。ここでは２つの攻撃に対しての耐性を説明する。1 つ目の攻撃は、通信される情報を盗聴し鍵を盗もうとする攻撃である。２つ目は、ノードを解析することによって情報を得ようとする攻撃を想定する。

まず、従来方式について考える。従来方式において BS から送信される情報は乱数 r、ノード数 n、中心点の座標の３つのみである。乱数 r が攻撃者に盗聴されたとしても、r はグループ鍵に直接関係しているわけではないので問題ない。また、r から計算される座標点を全数探索出来たとしても、その数が膨大であれば、その組み合わせの数はさらに膨大になる。よって、乱数 r から座標点を得る事は困難である。ノード数 n が盗聴されたとしても、n-1 次元に存在する球は無限大に存在する為、n が漏洩しても問題ない。また、同様に中心点の座標が漏洩したとしても、同じ中心点を持

つ同心円は無限大に存在する。ノードを解析された場合、その時のグループ鍵とそのノードの固有鍵が漏洩する。しかし、他のノードの座標は円上にある事はわかるが、どこに存在するかは分からない。よってノード解析されたとしてもそのノード以外の情報は漏洩しない。

次に、提案方式について考える。提案方式も基本的には従来方式と同じである。送信される情報は、乱数 r、グループ番号、グループ毎の中心点、c_jの４つである。グループ番号が漏洩したとしても得られる情報はノード数のみであり、そのグループ内の鍵s_jの情報は漏洩しない。また、c_jについても乱数r_jと l を足し、s_jで暗号化したものと$d_{j,l}$と排他的論理和を行い送信しているため、c_jから、$d_{j,l}$を求める事は困難である。また、ノードが解析されたとしても、従来方式同様、他のノードの情報が漏洩する事はない。

5.4 記憶容量

次に、ノードの記憶容量について考える。ノードのメモリ容量は小さいとされる為、必要な記憶容量は小さい方がよい。また、評価には通信量で用いた *L1*、*L2* を用いる。

まず、従来方式について考える。従来方式ではノードは自分の固有鍵のみを持っていればよいので、*L1* となる。

提案方式でも同様でノードは自分の固有鍵のみを記憶していればよいので、提案方式に必要な記憶容量は $L1$ となる。以上の結果を比較したものを以下の表３に示す。

表 4　記憶容量の比較

従来方式	提案方式
$L1$	$L1$

表３より､記憶容量では従来方式と提案方式に差はない。

5.5 鍵更新

最後に鍵更新の行いやすさについて説明する。IoT ネットワークでは、ノードの追加、削除など通信を行うグループが変更される場合がある。この時にグループ鍵を設定しなおす必要があり、その行いやすさを比較する。この比較には、通信量、計算量、記憶容量を総合して考える必要がある。

まず、従来方式について考える。従来方式において鍵更新を行うには、乱数 r を新たに生成して行う。乱数 r を更新することによって、疑似乱数生成器への入力が変わり異なる位置の座標点が出力される。この新たな座標点に対して、BS が新たな中心点を計算しブロードキャストする。各ノードは更新された r から座標点を生成し、新しい中心点との距離を新しい鍵とする。以上より、従来方式では乱数 r を変えるだけで簡単に鍵更新を行うことが出来る。

次に、提案方式について考える。提案方式も基本的なアルゴリズムは同じなので、乱数r_jを変更することで座標点の配置を変え、鍵更新を行う。

6. まとめ

本稿では、マンハッタン距離を用いる事で従来方式の問題であった計算量の軽減を図り、従来方式と比較した。しかし、３章に示した通り、マンハッタン距離をグループ鍵共有に用いるとすると、成功率が低いという課題が存在する。この為、提案方式をペア鍵共有に用いる事が現実的であると考える。扱う点の数が３つ以上の場合は、正方形になる為、成功確率の問題が発生するが、２点の場合は直線上での処理となる。これならば、成功確率の問題を解決できる可能性がある。よってマンハッタン距離を鍵共有に用いるならペア鍵が妥当であると考える。

参考文献

[1]濱崎　純, 金子　良, 岩村　惠市, “ユークリッド距離を用いた幾何学的グループ鍵共有法,” 信学論, J100-B 巻, 5 号, 375-383, 2017 年 5 月.