Comparison of discretization methods for helper data schemes with two-dimensional source

Guo, Z.

Award date:
2012

Comparison of Discretization Methods for Helper Data Schemes with Two-Dimensional Source

Z. Guo
September 2012

Technische Universiteit Eindhoven

Department of Mathematics and Computer Science

Master's Thesis

Comparison of Discretization Methods for Helper Data Schemes with Two-Dimensional Source

by Z. Guo

Supervisor:
Dr. Boris Škorić

Eindhoven, 2012

Abstract

Helper data schemes are an important primitive in the field of privacy-preserving biometrics and Physical Unclonable Functions. The schemes extract a secret, noise-free bit-string from a noisy measurement. In this thesis, we study helper data schemes in the case of two-dimensional Gaussian sources. Inspired by literature which shows that a hexagonal tiling has better error correction properties than a square tiling, we compare the performance of a secure sketch construction using these two tilings for discretization. By using an information-theoretic performance indicator, we find that the square tiling has better performance than the hexagonal tiling.

Table of Contents

1. Introduction

Humans can be identified by their physiological characteristics, such as fingerprint, face, voice, iris and so forth [1]. These characteristics are unique to each individual and can thus be used for authentication and identification systems. During enrollment, a person's biometric data is measured and stored in a database. Later, at verification, a fresh measurement of the person's biometrics is compared against the stored data. Biometrics have the advantage that they cannot be forgotten, in contrast to passwords.

However, storage of biometric data has its own problems if an inside attacker or hacker gets access to the enrolled data. There are privacy risks, such as the possibility that biometrics reveal medical conditions, and security risks, such as the attacker getting unauthorized access to services and the attacker leaving fake biometric evidence at a crime scene.

The problems can be solved by applying a one-way hash function to the biometric, in the same way as password storage in Unix. However, hashing the biometrics directly will cause a lot of errors. As we know, noise may appear while collecting the input physiological characteristics. For example, when a biometric system collects fingerprint, users swipe their fingers on the fingerprint reader with different pressure or at different angles. To solve these kind of problems, a helper data scheme is used to correct errors in the noisy data and then hash the biometrics afterwards. This will be explained in detail in section 2.3. Helper data schemes are also important in the field of Physical Unclonable Functions [2,3].

Biometric data is multi-dimensional. Results in two papers [4,5] hint that splitting the space into two-dimensional subspaces and then applying a hexagonal lattice for discretization may have better error correction performance than alternative discretization scheme, for instance, one-dimensional subspaces and square lattice.

In [4], an optimum Gray-code mapping was proposed for the signal constellation of modified QAM (m-QAM) and Triangular-Shaped Signal Set (TSSS). Here, the signal constellation of m-QAM is two-dimensional rectangular lattice based constellation, and the signal constellation of TSSS is two-dimensional hexagonal lattice based constellation. Then they compared the bit error probability between these two constellations. Their result shows that for the same peak signal-to-noise ratio (SNR), the bit error probability of the hexagonal lattice based constellation is smaller than the rectangular lattice based constellation.

In [5], a continuous Gaussian source was discretized by quantization index modulation. The discrete data, presented in both hexagonal tiling and square tiling, were used as the source data of a fuzzy extractor to extract randomness. Then they studied the effect of the quantization strategy on the performance. One of their efficiency measures is the amount of the information leakage. Their result shows that the information leakage of the hexagonal tiling is lower than the square tiling.

The two results show better performance of the hexagonal tiling, which led us to make a comparison between the two tilings in the field of helper data schemes. In our study, a helper data scheme is introduced with two types of source data: the enrollment measurement and their noisy equivalent, measured during verification. For simplicity, the biometric and the noise are both assumed to be Gaussian distributed. Then the continuous source data are discretized according to a hexagonal tiling and a square tiling. In both tilings we study the performance of the helper data scheme. The first criterion we use is the mutual information between the discretized enrollment measurement and verification measurement. This

corresponds to the maximum achievable entropy of a noiseless string, extracted from the biometrics using a hypothetical ideal error correction method on the tiling.

Our second criterion is as follows. The coordinates on the tiling are converted to bit-strings using a Gray code. Then we assume a hypothetical ideal binary error correction method on the space of bit-strings. The performance measure is a rough estimate of the mutual information between the Gray-coded coordinates at enrollment and verification, which is called score in this thesis. Here we distinguish between two sorts of Gray code: existing Gray code and hypothetical "ideal" Gray code.

According to the two criteria, we perform a numerical analysis. In the first criterion, we concluded that the helper data scheme with the hexagonal tiling is comparable or slightly poorer than the square tiling; in the second criterion, the helper data scheme with the hexagonal tiling is also comparable or slightly poorer than the square tiling for both Gray code and “ideal” Gray code.

The outline of the thesis is as follows. In Chapter 2, preliminaries including notations and definitions are described; especially the Gray code, and the helper data scheme are introduced. Chapter 3 describes the fundamental measures of information theory applied in our hexagonal/square tiling model, which contains the computation of probability distribution, entropy, mutual information, the scores of the Gray-coded hexagonal/square tiling, etc. Chapter 4 presents the numerical analysis of the two tilings in the helper data scheme, which leads to the comparison results of the performance between the hexagonal tiling and the square tiling. In the end, the conclusion is drawn in Chapter 5 with some illustration of future work.

2. Preliminaries

2.1 Notations and Definitions

2.1.1 Notations

The notations are listed below. Notice that the random variables and numerical values are denoted with capital letters and lowercases, respectively; the radius of the hexagon/square is normalized to 1, which is the half distance of two neighbor tiles.

$\hat{e}_i$	Unit vector on i-axis.
(j,k)	Label of a hexagon in the hexagonal tiling. $j,k \in \mathbb{Z}$
$(\hat{j},\hat{k})$	Label of a reconstructed hexagon in the hexagonal tiling. $\hat{j},\hat{k} \in \mathbb{Z}$.
(u,v)	Label of a square in the square tiling. $u,v \in \mathbb{Z}$.
$(\hat{u},\hat{v})$	Label of a reconstructed square in the square tiling. $\hat{u},\hat{v} \in \mathbb{Z}$.
l	The distance beween two points.
σ_X^2	Variance of the Gaussian distribution scaled the radius of the tiles. This scaled variance is defined as dividing the variance of the Gaussian distribution by the square of the half distance between two neighbor squares/hexagons.
σ_N^2	Variance of the Gaussian distributed noise scaled the radius of the tiles. This scaled variance is defined as dividing variance of the Gaussian distributed noise by the square of the half distance between two neighbor squares/hexagons.
x	point that is two-dimensional Gaussian distribution featured with mean 0 and variance σ_X^2.
x'	Noisy version of x at the reconstruction phase, which is two-dimensional Gaussian distribution with an additive 2-dimensional Gaussian distributed noise.
B	The scores in bit-string.

2.1.2 The a-b Coordinate System for Hexagonal Tiling

For hexagonal tiling, a special coordinate system is defined, which is shown in figure 2.1. The center point O of the center hexagon is defined as the origin. There are two axes in the coordinate system, a-axis and b-axis, which meets at the origin with an angle of 60 degree. As figure 2.1 shows, both a-axis and b-axis are orthogonal to a hexagon side. The half distance of two neighbor hexagons is defined as the unit length in the a-b coordinate system.

Within the a-b coordinate system, the hexagonal label (j,k) is determined by j on a-axis and k on b-axis. The area of a single hexagon is $2\sqrt{3}$ in the a-b coordinate system.

The transformation between the a-b coordinate system and the Cartesian coordinate system is needed for later use. Suppose there is a vector $\hat{v}$ that can be presented in the Cartesian coordinate system as $\hat{v} = r\hat{e}_x + t\hat{e}_y$, r,$t \in R$, where $\hat{e}_x$ is a unit vector on x-axis, $\hat{e}_y$ is a unit vector on y-axis; also the vector can be presented in the a-b coordinate system as $\hat{v} = c\hat{e}_a + d\hat{e}_b$, $c,d \in R$, where $\hat{e}_a$ is a unit vector on a-axis and $\hat{e}_b$ is a unit vector on b-axis.

$$\hat{v} = \begin{bmatrix} \hat{e}_a & \hat{e}_b \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} \hat{e}_x & \hat{e}_y \end{bmatrix} \begin{bmatrix} r \\ t \end{bmatrix}.$$

M is defined as the transformation matrix from the a-b coordinate system to the Cartesian coordinate system. The inverse of M, M^{-1} is defined as the transformation matrix from the Cartesian coordinate system to the a-b coordinate system.

$$\begin{bmatrix} r \\ t \end{bmatrix} = M \begin{bmatrix} c \\ d \end{bmatrix}, \begin{bmatrix} c \\ d \end{bmatrix} = M^{-1} \begin{bmatrix} r \\ t \end{bmatrix}.$$

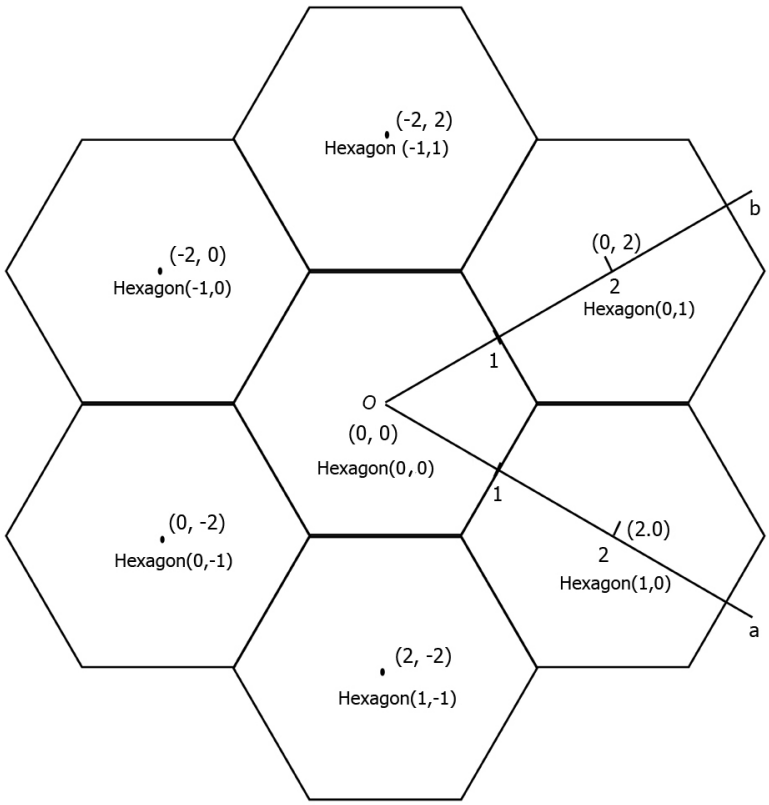

Figure 2.1. Hexagon label and the a-b coordinate system

The geometric relationship of unit vectors between the two coordinate systems are:

$$\hat{e}_a = \frac{\sqrt{3}}{2}\hat{e}_x - \frac{1}{2}\hat{e}_y,\ \hat{e}_b = \frac{\sqrt{3}}{2}\hat{e}_x + \frac{1}{2}\hat{e}_y.$$

With this relationship, we deduct as follows:

$$
\begin{aligned}
\hat{v} &= c\hat{e}_a + d\hat{e}_b \\
&= c(\frac{\sqrt{3}}{2}\hat{e}_x - \frac{1}{2}\hat{e}_y) + d(\frac{\sqrt{3}}{2}\hat{e}_x + \frac{1}{2}\hat{e}_y) \\
&= (\frac{\sqrt{3}}{2}c + \frac{\sqrt{3}}{2}d)\hat{e}_x + (-\frac{1}{2}c + \frac{1}{2}d)\hat{e}_y \\
&= r\hat{e}_x + t\hat{e}_y .
\end{aligned}
$$

So the relationship between $\begin{bmatrix} r \\ t \end{bmatrix}$ and $\begin{bmatrix} c \\ d \end{bmatrix}$ is:

$$
\begin{bmatrix} r \\ t \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix}. \qquad (2.1)
$$

Then M and M^{-1} is calculated:

$$
M = \begin{pmatrix} \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \text{ and } M^{-1} = \begin{pmatrix} \frac{1}{\sqrt{3}} & -1 \\ \frac{1}{\sqrt{3}} & 1 \end{pmatrix}.
$$

2.1.3 Square Tiling in the Cartesian Coordinate System.

The square tiling label (u,v) is defined in the Cartesian coordinate system with u on x-axis and v on y-axis, see figure 2.2. Similar to hexagonal tiling, the half distance of two neighbor square is defined as the unit length of the Cartesian coordinate system. Compared with the area of a single hexagon $2\sqrt{3}$, the area of a single square here is 4. Then the area of each hexagon is $\frac{\sqrt{3}}{2}$ of that of each square.

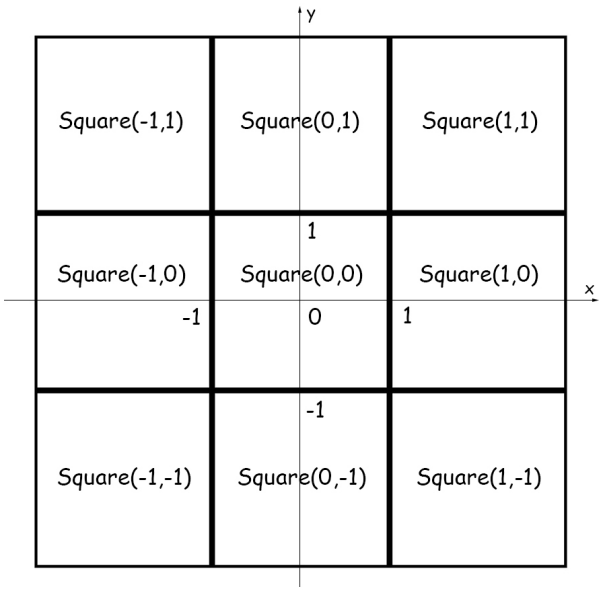

Figure 2.2. Square label and the Cartesian coordinate system

2.1.4 Distance between Two Points in the a-b Coordinate System

The distance l between point (m,n) and (m',n') is formulated in the a-b coordinate system. By using the law of cosines, l can be denoted by:

$$l^2 = (m-m')^2 + (n-n')^2 + (m-m')(n-n'). \tag{2.2}$$

Proof.

We have to look at four cases. Let $m-m'=m''$, $n-n'=n''$.

i. $m-m' > 0$, $n-n' > 0$

For this case, the angle between m and n is 120 degree. Using the Law of cosines:

$$l^2 = m''^2 + n''^2 - 2|m''n''|\cos(\frac{2\pi}{3}) = m''^2 + n''^2 - 2m''n''(-\frac{1}{2}) = m''^2 + n''^2 + m''n''.$$

ii. $m-m' > 0$, $n-n' < 0$

For this case, the angle m and n is 60 degree. As previous, using the Law of cosines:

$$l^2 = m''^2 + n''^2 - 2|m''n''|\cos(\frac{\pi}{3}) = m''^2 + n''^2 - 2(-m''n'')\frac{1}{2} = m''^2 + n''^2 + m''n''.$$

iii. $m-m' < 0$, $n-n' > 0$

For this case, the angle between m and n is 60 degree. Still:

$$l^2 = m''^2 + n''^2 - 2|m''n''|\cos(\frac{\pi}{3}) = m''^2 + n''^2 - 2(-m''n'')\frac{1}{2} = m''^2 + n''^2 + m''n''.$$

iv. $m-m' < 0$, $n-n' < 0$

For this case, the angle between m and n is 120 degree. And it comes to:

$$l^2 = m''^2 + n''^2 - 2|m''n''|\cos(\frac{2\pi}{3}) = m''^2 + n''^2 - 2m''n''(-\frac{1}{2}) = m''^2 + n''^2 + m''n''.$$

□

2.1.5 The Gaussian Distribution and the Gaussian Distributed Noise

In this section, we define the two-dimensional Gaussian distribution and the two-dimensional Gaussian distributed noise [6]. The variance on both x-axis and y-axis are equivalent in this model. Since the Gaussian distribution on x-axis and y-axis is independent, the two-dimensional Gaussian distribution of X and the two-dimensional Gaussian distributed noise in the Cartesian coordinate system are given by the follows :

$$f(x,y;\sigma_X^2) = \frac{1}{2\pi\sigma_X^2} e^{-\frac{x^2}{2\sigma_X^2} - \frac{y^2}{2\sigma_X^2}}, \tag{2.3}$$

$$f_N(x,y;\sigma_N^2) = \frac{1}{2\pi\sigma_N^2} e^{-\frac{x^2}{2\sigma_N^2} - \frac{y^2}{2\sigma_N^2}}. \tag{2.4}$$

For the a-b coordinate system the Gaussian distribution is denoted by the following formulas:

$$f(a,b;\sigma_X^2) = \frac{1}{2\pi\sigma_X^2} e^{-\frac{a^2+b^2+ab}{2\sigma_X^2}}, \tag{2.5}$$

$$f_N(a,b;\sigma_N^2) = \frac{1}{2\pi\sigma_N^2} e^{-\frac{a^2+b^2+ab}{2\sigma_N^2}}. \tag{2.6}$$

2.1.6 Entropy and Mutual Information

Mutual information is a measure of information overlap between two random variables. Since entropy and conditional entropy are introduced to calculate the mutual information, it is necessary to declare first the definition of entropy, conditional entropy, and thereafter mutual information [7].

We start with the definition of probability $p(x)$:

$$p(x) = \Pr\{X = x\}, x \in \mathcal{X}.$$

To compute the conditional entropy and the mutual information, the joint probability $p(x,x')$ and the conditional probability $p(x' \mid x)$ are introduced:

$$p(x,x') = \Pr\{X = x, X' = x'\},\ x \in \mathcal{X},\ x' \in \mathcal{X}',$$

$$p(x' \mid x) = \frac{\Pr\{X' = x', X = x\}}{\Pr\{X = x\}},\ x \in \mathcal{X},\ x' \in \mathcal{X}'.$$

The relationship between the joint probability and the conditional probability is:

$$p(x,x') = p(x)p(x' \mid x),\ x \in \mathcal{X},\ x' \in \mathcal{X}'.$$

In this thesis, entropy quantifies the expected value of the information contained in the tiling, which is defined as :

$$H(X) = \sum_{x \in X} p(x) \log \frac{1}{p(x)}. \tag{2.7}$$

Notice that here the log is to the base 2 and entropy is expressed in bits.

The conditional entropy $H(X'/X)$ is defined as:

$$\begin{aligned} H(X' \mid X) &= \sum_{x \in X} p(x) H(X' \mid X = x) \\ &= \sum_{x \in X} p(x) \sum_{x' \in X'} p(x' \mid x) \log \frac{1}{p(x' \mid x)}. \end{aligned} \tag{2.8}$$

The mutual information $I(X;X')$ is the relative entropy between the joint distribution and the product distribution $p(x)p(x')$:

$$I(X;X') = \sum_{x \in X, x' \in X'} p(x,x') \log \frac{p(x,x')}{p(x)p(x')}.$$

With the relationship between the joint probability and the conditional probability, the relationship between mutual information and entropy can be presented as:

$$\begin{aligned} I(X;X') &= \sum_{x \in X, x' \in X'} p(x,x') \log \frac{p(x,x')}{p(x)p(x')} \\ &= \sum_{x \in X, x' \in X'} p(x,x') \log p(x' \mid x) - \sum_{x \in X, x' \in X'} p(x,x') \log p(x') \\ &= \sum_{x \in X, x' \in X'} p(x)p(x' \mid x) \log p(x' \mid x) - \sum_{x \in X, x' \in X'} p(x \mid x')p(x') \log p(x') \\ &= \sum_{x \in X} p(x) \sum_{x' \in X'} p(x' \mid x) \log p(x' \mid x) - \sum_{x' \in X'} p(x') \log p(x'). \end{aligned}$$

From (2.7) and (2.8), we have the following result:

$$I(X;X') = H(X') - H(X' \mid X). \tag{2.9}$$

The binary entropy is defined as follows:

$$h(p) = -p\log p - (1-p)\log(1-p),\ p \in [0,1]. \tag{2.10}$$

Next we would like to introduce binary symmetric channel (BSC) capacity. A BSC is a binary channel that can only transmit symbol 0 or 1. In the BSC, the probability of bit-flip from 0 to 1 is the same as the bit-flip from 1 to 0. The capacity of BSC is the best correct rate one can achieve in one bit, which can be calculated as:

$$C = 1 - h(p). \tag{2.11}$$

We will use these information theory measures to measure the helper data scheme performance, which is introduced in section 3.2.

2.2 Gray Code

Gray code is used to transform the hexagon/square labels to bit-strings. With the following definitions, we will see what the gray code is and the normal Gray code mapping.

Hamming Distance [8]

For two bit-strings $e = \{e_1, e_2, \ldots, e_{n-1}\}$ and $f = \{f_1, f_2, \ldots, f_{n-1}\}$, the Hamming distance of these two bit-strings is defined as:

$$d_H(e,f) = \sum_{i=0}^{n-1} e_i \oplus f_i.$$

Gray code [9,10]

A set of bit-strings is Gray-coded if each pair of neighbor bit-strings has one bit difference. In other word, for a bit-string set $s = \{s_1, s_2, \ldots, s_n\}$, the Hamming distance of any neighbor bit-string pair $d_H(s_j,s_{j+1})$ is 1 bit, where $s_j, s_{j+1} \in s$. An example of normal Gray code mapping is shown in tabel 2.1, integers from 0 to 7 are assigned to binary code and the Gray code. In section 3.2, we will introduce a modified Gray-code mapping for the hexagonal/square tiling.

Int	Gray	Binary
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

Table 2.1. Gray Code versus Binary Code

2.3 Helper Data Scheme

A helper data scheme is introduced in this section based on the fuzzy extractor [2,11,12,13,14,15,16]. Before we start to introduce the helper data scheme, the noise, which is Gaussian distributed, includes two types: one is the user noise, which is cause by the user

input, for instance, users may take their face images with different angles; another one is the system noise, such as the noise of the fingerprint reader, the noise of the camera, etc.

In a biometric system, helper data are needed to reconstruct the noisy data. In figure 2.3, a helper data w_x is defined as a vector which initiates from x and center point O of the tile. Since the helper data only gives a relative position of x, This helper data has limitations: it can only direct the point to the center of the same tiling. Therefore, these helper data cannot correct every error introduced by the noise. If the noise is too large, the point x' may flow to the neighbor tiling, which will be corrected to the center of neighbor tile by the helper data instead of the right tile. The details of the helper data will be introduced in section 3.1.4 and 3.1.5.

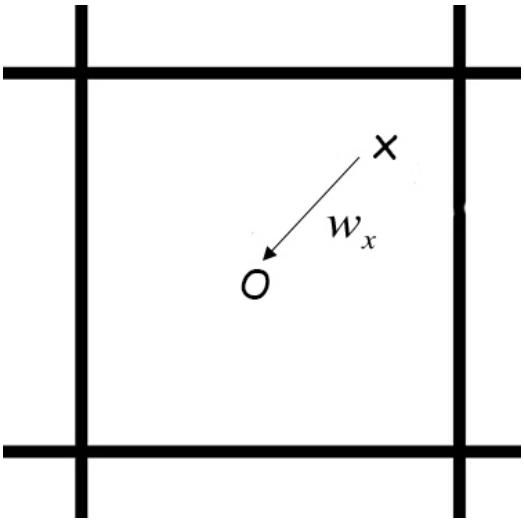

Figure 2.3. Helper data

Further, the helper data scheme is shown in figure 2.4. The left part shows the enrollment of the scheme. The biometric system collects the input data, which is clean and called enrolled data. After data collection, a Gen will generate secret data S and helper data W_X from X. Here helper data is stored in the database and should not leak much information about S. Then S is concatenated with salt C and hashed by a one-way hash function; the hashed data is denoted as H; afterwards, H and C are stored in the database. Notice that although physiological characteristics is unique, it is still possible that two enrolled data may have collision because of low system accuracy. Due to this reason a salt C, which is a random variable, can ensure the enrolled data collision-free. The right part of the figure is the verification of the scheme. The input data X' are the biometric data collected with noise. With the input data X', Rep will reconstruct secret data $\hat{S}$ from the noisy data X' with helper data W_X. Then the hash of secret data $\hat{S}$ concatenated with C is $\hat{H}$. At last, the system will check if $\hat{H}$ matches H to verify the input data X'.

In order to make the helper data scheme secure, we should take the attacker model into account. The attackers are assumed to have access to the stored data. Since helper data are supposed not to leak much information about secret, the attacker will fail to gain secret from the helper data. Also since we applied the one-way hash function to the secret, it is impossible for an attacker to obtain the secret from the hashed data. Here, with the salt concatenated with secret, the attacker can hardly get the same secret from different hashed data. The only way for an attacker to derive the secret is to use brute-force attack.

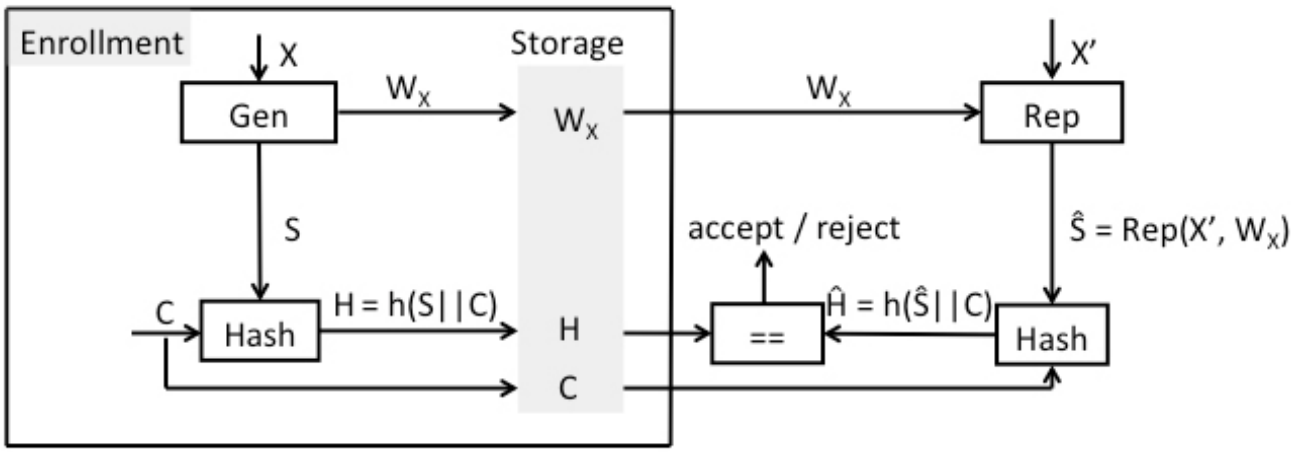

Figure 2.4. Helper data scheme

3. Measures of the Helper Data Scheme Performance

The performance of the helper data scheme consists of the following two parameters: one is the reconstruction error probability; and the other one is the conditional entropy $H(S|W_X)$. We will choose the mutual information $I(S;\hat{S})$ as a measure of the overall performance in theory. Here given an ideal error correction code, the mutual information $I(S;\hat{S})$ measures how many bits of information can be robustly extracted from X in theory. This will be introduced in section 3.1. Also we will look at the scores in bit-string, which is a roughly estimated mutual information, as a measure of the overall performance. Here we assume an ideal binary error correction codes exists; the bit-strings are transformed by modified Gray-code mappings; and given an ideal binary error correction code, the mutual information of the BSC is considered to measure how many bits can be robustly extracted from each bit of the bit-string. This will be discussed in section 3.2. Since it would be too difficult to make a fair comparison if we use actual error-correcting codes, we introduce ideal error-correcting codes to our two cases.

3.1 Computation of Theoretical Measures

To discretize continuous data to the hexagonal tiling, the boundary of each hexagon is defined with the method to determine JK from X in the a-b coordinate system in section 3.1.1. To compute the probability of each hexagon efficiently, we also define the boundary of each hexagon in the Cartesian coordinate system in section 3.1.2. With the defined boundary of each hexagon, we explain how to compute the probability of each hexagon and the noise probability in section 3.1.3. Afterwards in section 3.1.4, the calculation of entropy and conditional entropy are illustrated, which leads to the final computation of the mutual information. In the end, the same methods to obtain those measures are introduced for the square tiling in section 3.1.5.

3.1.1 Boundary of Each Hexagon in the a-b Coordinate System

In the a-b coordinate system, 6 vertices of the origin hexagon are shown in figure 3.1. Since the center point of a hexagon labeled (j,k) has the coordinate value $(2j,2k)$, where $j,\ k \in Z$, by transferring the origin hexagon to the hexagon (j,k), we have the 6 vertices of hexagon (j,k) listed in table 3.1. Notice that in the remainder of this thesis, $j,\ k \in Z$.

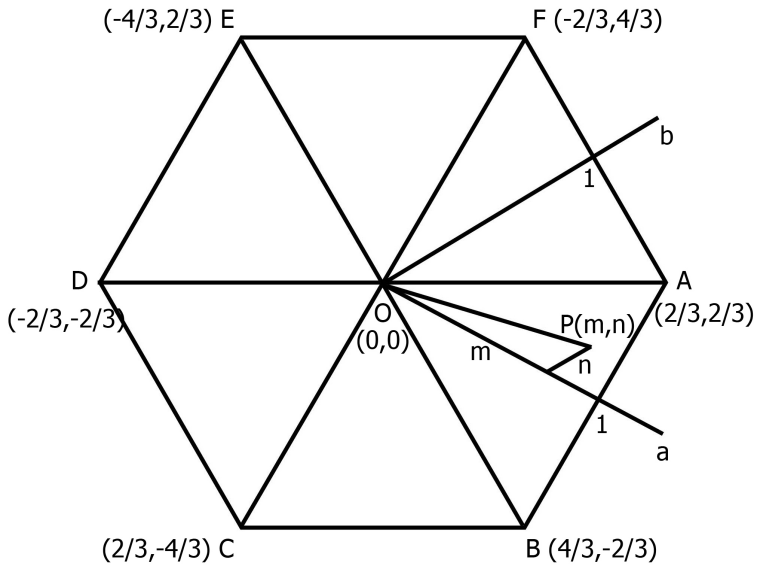

Figure 3.1. Origin hexagon in the a-b coordinate system

A	B	C	D	E	F
$(\frac{2}{3}+2j,\frac{2}{3}+2k)$	$(\frac{4}{3}+2j,-\frac{2}{3}+2k)$	$(\frac{2}{3}+2j,-\frac{4}{3}+2k)$	$(-\frac{2}{3}+2j,-\frac{2}{3}+2k)$	$(-\frac{4}{3}+2j,\frac{2}{3}+2k)$	$(-\frac{2}{3}+2j,\frac{4}{3}+2k)$

Table 3.1 Vertices of hexagon (j,k) in the a-b coordinate system

Based on the 6 vertices listed above, the 6 sides of the hexagon (j,k) is defined in table 3.2.

Side	Formula	Vector Formula
AB	$b=-2a+2+4j+2k,\ a\in[\frac{2}{3}+2j,\frac{4}{3}+2j]$	$\begin{pmatrix} a \\ b \end{pmatrix}=\begin{pmatrix} \frac{2}{3}+2j \\ \frac{2}{3}+2k \end{pmatrix}+\lambda\begin{pmatrix} \frac{2}{3} \\ -\frac{4}{3} \end{pmatrix},\ \lambda\in[0,1]$
BC	$b=a-2-2j+2k,\ a\in[\frac{2}{3}+2j,\frac{4}{3}+2j]$	$\begin{pmatrix} a \\ b \end{pmatrix}=\begin{pmatrix} \frac{4}{3}+2j \\ -\frac{2}{3}+2k \end{pmatrix}+\lambda\begin{pmatrix} -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix},\ \lambda\in[0,1]$
CD	$b=-\frac{1}{2}a-1+j+2k,\ a\in[-\frac{2}{3}+2j,\frac{2}{3}+2j]$	$\begin{pmatrix} a \\ b \end{pmatrix}=\begin{pmatrix} \frac{2}{3}+2j \\ -\frac{4}{3}+2k \end{pmatrix}+\lambda\begin{pmatrix} -\frac{4}{3} \\ \frac{2}{3} \end{pmatrix},\ \lambda\in[0,1]$
DE	$b=-2a-2+4j+2k,\ a\in[-\frac{4}{3}+2j,-\frac{2}{3}+2j]$	$\begin{pmatrix} a \\ b \end{pmatrix}=\begin{pmatrix} -\frac{2}{3}+2j \\ -\frac{2}{3}+2k \end{pmatrix}+\lambda\begin{pmatrix} -\frac{2}{3} \\ \frac{4}{3} \end{pmatrix},\ \lambda\in[0,1]$
EF	$b=a+2-2j+2k,\ a\in[-\frac{4}{3}+2j,-\frac{2}{3}+2j]$	$\begin{pmatrix} a \\ b \end{pmatrix}=\begin{pmatrix} -\frac{4}{3}+2j \\ \frac{2}{3}+2k \end{pmatrix}+\lambda\begin{pmatrix} \frac{2}{3} \\ \frac{2}{3} \end{pmatrix},\ \lambda\in[0,1]$
FA	$b=-\frac{1}{2}a+1+j+2k,\ a\in[-\frac{2}{3}+2j,\frac{2}{3}+2j]$	$\begin{pmatrix} a \\ b \end{pmatrix}=\begin{pmatrix} -\frac{2}{3}+2j \\ \frac{4}{3}+2k \end{pmatrix}+\lambda\begin{pmatrix} \frac{4}{3} \\ -\frac{2}{3} \end{pmatrix},\ \lambda\in[0,1]$

Table 3.2 Sides of hexagon (j,k) in the a-b coordinate system

According to formula (2.3), (2.4), (2.5), and (2.6), it is obvious that the calculation in the Cartesian coordinate system ((2.3) and (2.4)) is simpler than in the a-b coordinate system

((2.5) and (2.6)). Hence, we will introduce the expression of side-formulas in the Cartesian coordinate system in section 3.1.2.

Determine JK

For any point P *(m,n)*, there exists and only exists one parallelogram to encompass P, where the four vertices of the parallelogram are the center points of four neighbor hexagons. The parallelogram is marked gray in figure 3.2. P located in the parallelogram satisfies the following two inequalities:

$$\begin{cases} 2j \le m \le 2(j+1) \\ 2k \le n \le 2(k+1) \end{cases}$$

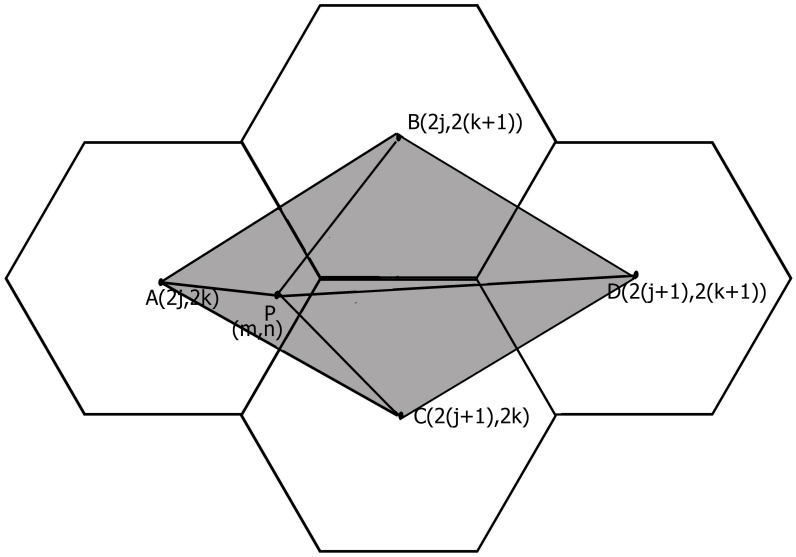

Figure 3.2. Locate point P in the a-b coordinate system

The center points of the four neighbor hexagons are A(2j, 2k), B(2j, 2(k+1)), C(2(j+1), 2k), and D(2(j+1), 2(k+1)). To determine JK from P, the squared distances between P to the center points (|AP|², |BP|², |CP|², and |DP|²) are calculated according to formula (2.2). Here to simplify the result, let $m = 2j + 1 + s$, $n = 2k + 1 + t$ with $s, t \in [-1,1]$.

$$\begin{aligned} &|AP|^2 \\ &= (m-2j)^2 + (n-2k)^2 + (m-2j)(n-2k) \\ &= (1+s)^2 + (1+t)^2 + (s+1)(t+1) \\ &= s^2 + t^2 + st + 3s + 3t + 3 \\ &|BP|^2 \\ &= (m-2j)^2 + [n-2(k+1)]^2 + (m-2j)[n-2(k+1)] \\ &= (s+1)^2 + (t-1)^2 + (s+1)(t-1) \\ &= s^2 + t^2 + st + s - t + 1 \end{aligned}$$

$$
\begin{aligned}
&|CP|^2 \\
&= [m-2(j+1)]^2 + (n-2k)^2 + (n-2k)[m-2(j+1)] \\
&= (s-1)^2 + (t+1)^2 + (t+1)(s-1) \\
&= s^2 + t^2 + st - s + t + 1 \\
&|DP|^2 \\
&= [m-2(j+1)]^2 + [n-2(k+1)]^2 + [m-2(j+1)][n-2(k+1)] \\
&= (s-1)^2 + (t-1)^2 + (s-1)(t-1) \\
&= s^2 + t^2 + st - 3s - 3t + 3
\end{aligned}
$$

Then, by comparing the squared distances ($|AP|^2$, $|BP|^2$, $|CP|^2$, and $|DP|^2$), the location of point P can be determined. by the minimum distance with. The hexagon where p locates has the minimum distance to P.

3.1.2 Boundary of Each Hexagon in the Cartesian Coordinate System

In this section, the boundary of each hexagon will be defined in the Cartesian coordinate system. These boundaries will be later used to calculate the probability of each hexagon. Figure 3.3 shows the 6 vertices and the 6 sides of the origin hexagon in the Cartesian coordinate system. In section 3.1, we have defined the boundary of each hexagon in the a-b coordinate system. Transformation according to formula (2.1) is applied from the a-b coordinate system to the Cartesian coordinate system. Then the sides of hexagon (j,k) in the Cartesian coordinate system are presented as follows:

EF and BC:

$$y = \pm 1 - j + k,\ x \in \left[-\frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k, \frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k \right]$$

CD and DE:

$$y = \sqrt{3}x + 2 - 4j - 2k,\ x \in \left[-\frac{2}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k, -\frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k \right]$$

$$y = -\sqrt{3}x - 2 + 2j + 4k,\ x \in \left[-\frac{2}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k, -\frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k \right]$$

AB and AF:

$$y = \sqrt{3}x - 2 - 4j - 2k,\ x \in \left[\frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k, \frac{2}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k \right]$$

$$y = -\sqrt{3}x + 2 + 2j + 4k,\ x \in \left[\frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k, \frac{2}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k \right]$$

In order to calculate the probability, each hexagon is splited to three regions with intent: one rectangle (blue region) and two triangles (red region) as figure 3.3 shows. These three regions can be illustrated in inequalities as follows:

Rectangle BCEF:

$$-1 - j + k \le y \le 1 - j + k,\ x \in \left[-\frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k, \frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k \right]$$

Triangle CDE:

$$-\sqrt{3}x-2+2j+4k \le y \le \sqrt{3}x+2-4j-2k,\ x \in \left[-\frac{2}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k, -\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k\right]$$

Triangle ABF:

$$\sqrt{3}x-2-4j-2k \le y \le -\sqrt{3}x+2+2j+4k,\ x \in \left[\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k, \frac{2}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k\right]$$

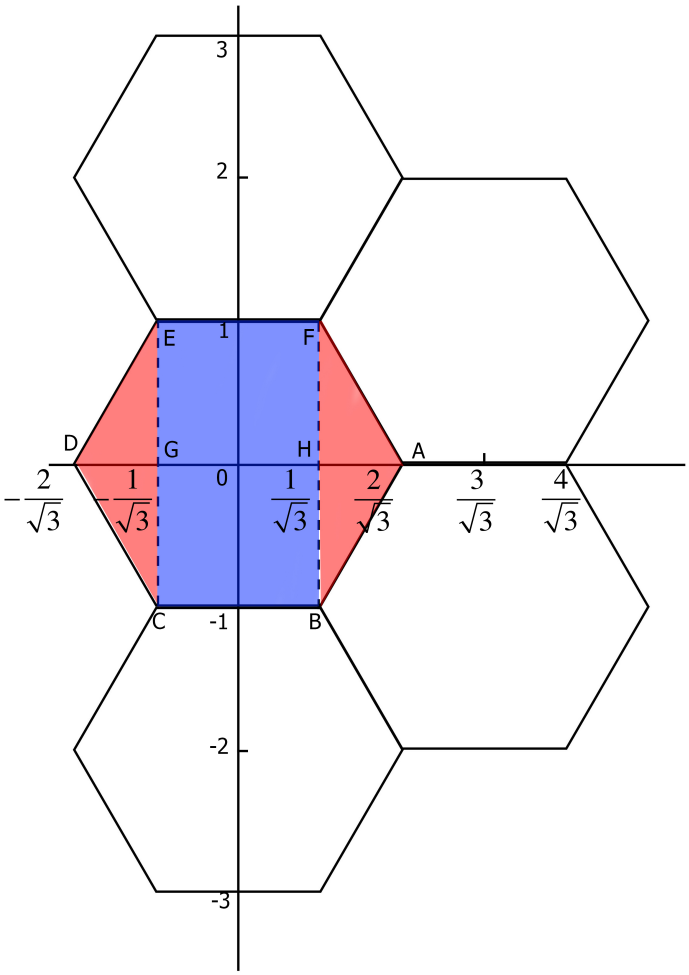

Figure 3.3. Boundary of each hexagon in the Cartesian coordinate system

3.1.3 Probability of Each Hexagon

In section 3.1.2, each hexagon is splited to three regions in the Cartesian coordinate system: one rectangle and two triangles, see figure 3.3. To compute the Gaussian probability for each part, the interval of the probability integral for each hexagon is defined:

Interval of integral for rectangle

$$x_{\min} = -\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k,\ x_{\max} = \frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k;$$
$$y_{\min} = -1-j+k,\ y_{\max} = 1-j+k.$$

Interval of integral for left triangle:

$$x_{\min} = -\frac{2}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k,\ x_{\max} = -\frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k;$$
$$y_{\min} = -\sqrt{3}x - 2 + 2j + 4k,\ y_{\max} = \sqrt{3}x + 2 - 4j - 2k.$$

Interval of integral for right triangle:

$$x_{\min} = \frac{1}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k,\ x_{\max} = \frac{2}{\sqrt{3}} + \sqrt{3}j + \sqrt{3}k;$$
$$y_{\min} = \sqrt{3}x - 2 - 4j - 2k,\ y_{\max} = -\sqrt{3}x + 2 + 2j + 4k.$$

Refer to formula (2.3), the probability integral of each part is defined as follows:

Integral of rectangle

$$f_{rec}(j,k;\sigma^2) = \int_{x=-\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k}^{\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k} \int_{y=-1-j+k}^{1-j+k} \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}-\frac{y^2}{2\sigma^2}} \mathrm{d}x\,\mathrm{d}y.$$

Integral of left triangle

$$f_L(j,k;\sigma^2) = \int_{x=-\frac{2}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k}^{-\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k} \int_{y=-\sqrt{3}x-2+2j+4k}^{\sqrt{3}x+2-4j-2k} \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}-\frac{y^2}{2\sigma^2}} \mathrm{d}x\,\mathrm{d}y.$$

Integral of right triangle

$$f_R(j,k;\sigma^2) = \int_{x=\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k}^{\frac{2}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k} \int_{y=\sqrt{3}x-2-4j-2k}^{-\sqrt{3}x+2+2j+4k} \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}-\frac{y^2}{2\sigma^2}} \mathrm{d}x\,\mathrm{d}y.$$

After calculation, the derived probabilities are as below:

$$f_{rec}(j,k;\sigma^2) = \frac{1}{4}\left(Erf\left[\frac{1+j-k}{\sqrt{2}\sigma}\right] + Erf\left[\frac{1-j+k}{\sqrt{2}\sigma}\right]\right)\left(Erf\left[\frac{-1+3j+3k}{\sqrt{6}\sigma}\right] + Erf\left[\frac{1+3j+3k}{\sqrt{6}\sigma}\right]\right),$$

$$f_L(j,k;\sigma^2) = \int_{x=-\frac{2}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k}^{-\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k} \frac{\sqrt{2\pi}}{4\sigma\pi} e^{-\frac{x^2}{2\sigma^2}} \left(Erf\left[\frac{\sqrt{3}x+2-2j-4k}{\sqrt{2}\sigma}\right] + Erf\left[\frac{\sqrt{3}x+2-4j-2k}{\sqrt{2}\sigma}\right]\right)\mathrm{d}x,$$

$$f_R(j,k;\sigma^2) = \int_{x=\frac{1}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k}^{\frac{2}{\sqrt{3}}+\sqrt{3}j+\sqrt{3}k} \frac{\sqrt{2\pi}}{4\sigma\pi} e^{-\frac{x^2}{2\sigma^2}} \left(Erf\left[\frac{-\sqrt{3}x+2+4j+2k}{\sqrt{2}\sigma}\right] + Erf\left[\frac{-\sqrt{3}x+2+2j+4k}{\sqrt{2}\sigma}\right]\right)\mathrm{d}x.$$

Noise applied on the center of each hexagon has identical distribution, that is to say, the noise probability distribution is independent of the location of hexagon. To analyse the noise probability distribution with hexagon (j,k) as center hexagon, a relative a-b coordinate system is introduced with the center of hexagon (j,k) as origin, shown in figure 3.4. The red labels are the hexagon labels relative to hexagon (j,k) with the absolute labels marked in black. Any hexagon in the relative a-b coordinate system can be denoted as (j',k') with the absolute hexagon label as $(\hat{j},\hat{k})$ in the a-b coordinate system, where $\hat{j} = j' + j$ and $\hat{k} = k' + k$.

Finally, the probability of the hexagon (j,k) and the noise probability of the hexagon (j',k') are shown in formula (3.1) and formula (3.2), respectively:

$$p(j,k) = f_{rec}(j,k;\sigma_X^2) + f_L(j,k;\sigma_X^2) + f_R(j,k;\sigma_X^2) \tag{3.1}$$

$$p(j',k') = f_{rec}(j',k';\sigma_N^2) + f_L(j',k';\sigma_N^2) + f_R(j',k';\sigma_N^2) \tag{3.2}$$

Notice that $p(j,k)$ and $p(j',k')$ are independent.

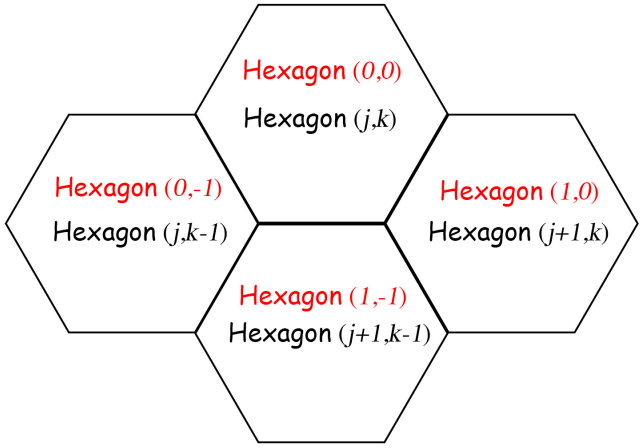

Figure 3.4. Noise probability distribution in the relative a-b coordinate system

3.1.4 Entropy and Mutual Information of Hexagonal Tiling

When users put their physiological characteristics into the biometric system, ideally the input data is exactly the same as biometrics data previously stored in the database. However, noise inevitably exists in reality, which causes errors. In this thesis, we assume that the enrolled data is the real physiological characteristics, and the verification data is the noisy equivalent.

From figure 3.5, a helper data is applied to direct a point P to the center point O of the hexagon (j,k). Without the helper data, if noise is added to the hexagonal/square tiling, P will be located in the circle area, of which P is the center. With the helper data, the circle area with center point P is transfered to a circle area with center point O, which can minimize the errors introduced by the noise and reduce the false rejection rate.

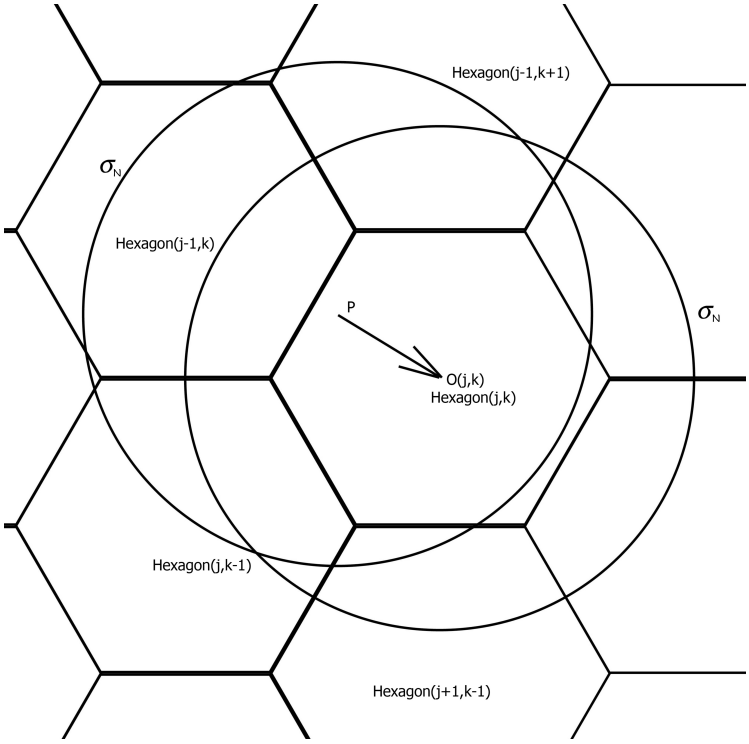

Figure 3.5. Helper data: direct points within a hexagon to the center

Refer to formula (3.1), entropy of the enrolled data and the noise is denoted as formula (3.3) and formula (3.4), respectively:

$$H(JK) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} p(j,k) \log \frac{1}{p(j,k)} \tag{3.3}$$

$$H(J'K') = \sum_{j'=-\infty}^{+\infty} \sum_{k'=-\infty}^{+\infty} p(j',k') \log \frac{1}{p(j',k')} \tag{3.4}$$

To compute the entropy of the reconstructed noisy data, which helper data are applied to, the probability of the noisy hexagon $(\hat{j},\hat{k})$ is first calculated as follows:

$$p(\hat{j},\hat{k}) = \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} p(j,k) \Pr\{\hat{J}\hat{K} = (\hat{j},\hat{k}) \mid JK = (j,k)\} \tag{3.5}$$

According to the relationship between the relative and absolute Cartesian coordinate systems described in section 3.3, the conditional probability $\Pr\{\hat{J}\hat{K} = (\hat{j},\hat{k}) \mid JK = (j,k)\}$ is equivalent to the conditional probability $\Pr\{J'K' = (j',k') \mid JK = (j,k)\}$, which is the definition of $p(j',k')$. Then formula (3.5) can be denoted as below:

$$p(\hat{j},\hat{k}) = \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} p(j,k) p(\hat{j}-j,\hat{k}-k) \tag{3.6}$$

Refer to formula (2.7), the entropy of the $\hat{J}\hat{K}$ is presented as:

$$H(\hat{J}\hat{K}) = \sum_{\hat{j}=-\infty}^{+\infty} \sum_{\hat{k}=-\infty}^{+\infty} p(\hat{j},\hat{k}) \log \frac{1}{p(\hat{j},\hat{k})} \tag{3.7}$$

Also, the conditional entropy based on formula (2.8) can be denoted as follows:

$$H(\hat{J}\hat{K} \mid JK) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} p(j,k) H(\hat{J}\hat{K} \mid JK = (j,k)) \tag{3.8}$$

Since $\Pr\{\hat{J}\hat{K} = (\hat{j},\hat{k}) \mid JK = (j,k)\} = p_{noise}(\hat{j}-j,\hat{k}-k)$, we have:

$$\begin{aligned}
&H(\hat{J}\hat{K} \mid JK = (j,k)) \\
&= \sum_{\hat{j}=-\infty}^{+\infty} \sum_{\hat{k}=-\infty}^{+\infty} \Pr\{\hat{J}\hat{K} = (\hat{j},\hat{k}) \mid JK = (j,k)\} \log \frac{1}{\Pr\{\hat{J}\hat{K} = (\hat{j},\hat{k}) \mid JK = (j,k)\}} \\
&= \sum_{\hat{j}=-\infty}^{+\infty} \sum_{\hat{k}=-\infty}^{+\infty} p(\hat{j}-j,\hat{k}-k) \log \frac{1}{p(\hat{j}-j,\hat{k}-k)} \\
&= \sum_{j'=-\infty}^{+\infty} \sum_{k'=-\infty}^{+\infty} p(j',k') \log \frac{1}{p(j',k')} \\
&= H(J'K')
\end{aligned} \tag{3.9}$$

Based on formula (3.8) and (3.9), and the independence between $p(j,k)$ and $p(j',k')$, formula (3.8) can be denoted as follows:

$$\begin{aligned}
H(\hat{J}\hat{K} \mid JK) &= \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} p(j,k) H(J'K') \\
&= H(J'K') \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} p(j,k) \\
&= H(J'K')
\end{aligned} \tag{3.10}$$

From formula (3.7) and (3.10) together with formula (2.9) , the mutual information of the hexagonal tiling is:

$$I(JK;\hat{J}\hat{K}) = H(\hat{J}\hat{K}) - H(\hat{J}\hat{K} \mid JK)$$
$$= \sum_{\hat{j}=-\infty}^{+\infty} \sum_{\hat{k}=-\infty}^{+\infty} p(\hat{j},\hat{k}) \log \frac{1}{p(\hat{j},\hat{k})} - H(J'K') \tag{3.11}$$

3.1.5 Computation of Measures in Square Tiling

In this section, we will introduce the boundary of the square, probability of each square, and the entropy and the mutual information of the square tiling.

Boundary of the Square

For the square tiling, the boundary of a square (u,v) is $x \in [-1+2u, 1+2u]$ and $y \in [-1+2v, 1+2v]$. See figure 3.6. In the figure, red lines are the boundary of square $(-1,1)$. To determine UV from a point (m,n), we can just simply find u and v that meet $m \in [-1+2u, 1+2u]$ and $n \in [-1+2v, 1+2v]$. Notice that in the rest of this thesis, $u, v \in Z$.

Probability distribution on the square tiling

Probability distribution of each square is similar to the distribution of each hexagon but with different boundaries, which is presented as:

$$p(u,v) = \int_{x=-1+2u}^{1+2u} \int_{y=-1+2v}^{1+2v} \frac{1}{2\pi\sigma_X^2} e^{-\frac{x^2}{2\sigma_X^2} - \frac{y^2}{2\sigma_X^2}} \mathrm{d}x\,\mathrm{d}y$$
$$= \frac{1}{4}\left(Erf\left[\frac{-1+2u}{\sqrt{2}\sigma_X}\right] - Erf\left[\frac{1+2u}{\sqrt{2}\sigma_X}\right]\right)\left(Erf\left[\frac{-1+2v}{\sqrt{2}\sigma_X}\right] + Erf\left[\frac{1+2v}{\sqrt{2}\sigma_X}\right]\right) \tag{3.12}$$

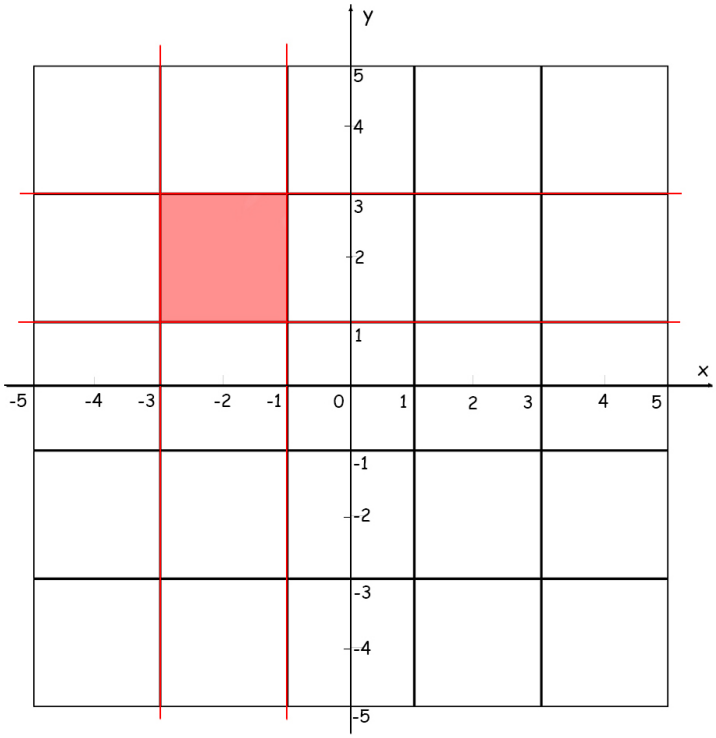

Figure 3.6. Boundary of each square in the Cartesian coordinate system

Similar to the noise probability of the hexagonal tiling $p(j',k')$, we also calculate the noise probability $p(u',v')$ in the relative Cartesian coordinate system with (u,v) as center, which can be presented as:

$$\begin{aligned} p(u',v') &= \int_{x=-1+2u'}^{1+2u'} \int_{y=-1+2v'}^{1+2v'} \frac{1}{2\pi\sigma_N^2} e^{-\frac{x^2}{2\sigma_N^2}-\frac{y^2}{2\sigma_N^2}} \mathrm{d}x\,\mathrm{d}y \\ &= \frac{1}{4}\left(Erf\left[\frac{-1+2u'}{\sqrt{2}\sigma_N}\right] - Erf\left[\frac{1+2u'}{\sqrt{2}\sigma_N}\right]\right)\left(Erf\left[\frac{-1+2v'}{\sqrt{2}\sigma_N}\right] + Erf\left[\frac{1+2v'}{\sqrt{2}\sigma_N}\right]\right) \end{aligned} \tag{3.13}$$

Here $(\hat{u},\hat{v})$ is the absolute square label, where $u' = \hat{u} - u,\ v' = \hat{v} - v$. The same as in the hexagonal tiling, $p(u,v)$ and $p(u',v')$ are independent.

Entropy and Mutual Information

From formula (2.7), the entropy of the square tiling is denoted as:

$$H(UV) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} p(u,v) \log \frac{1}{p(u,v)} \tag{3.14}$$

$$H(U'V') = \sum_{u'=-\infty}^{+\infty} \sum_{v'=-\infty}^{+\infty} p(u',v') \log \frac{1}{p(u',v')} \tag{3.15}$$

Similar to the hexagonal tiling, the helper data is applied to direct a point P to the center point O of the square (u,v), see figure 3.7. The two circles indicate the noise area. The the noise area of point P is transfered to the noise area of point O, which can minimize the errors introduced by the noise and reduce the false rejection rate.

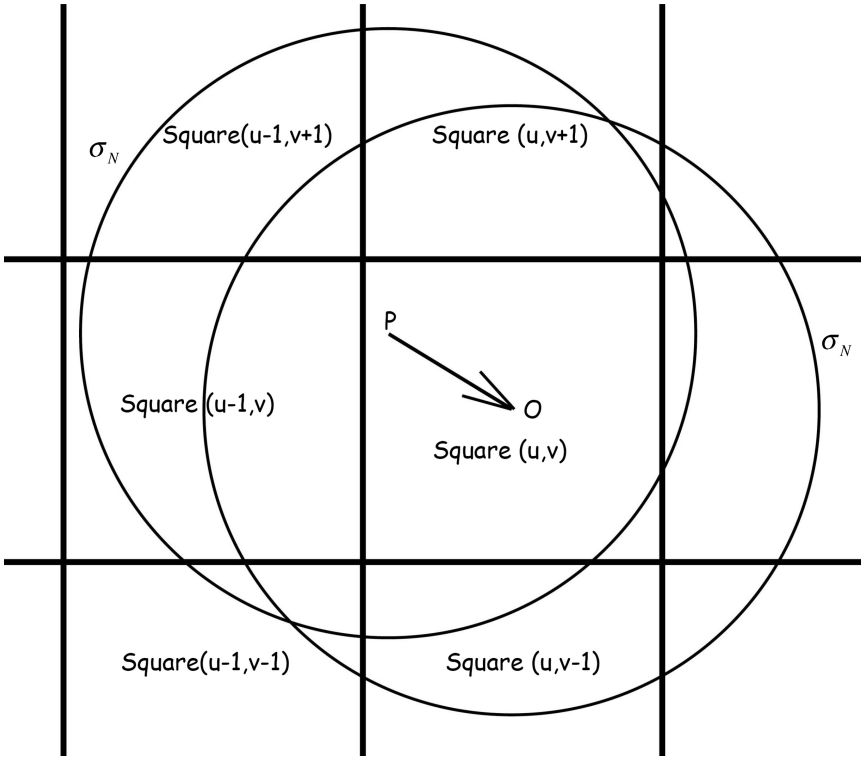

Figure 3.7. Helper data: direct points within a square to the center

Similar to the hexagonal tiling, the entropy of the $\hat{U}\hat{V}$ and the conditional entropy can be presented as:

$$H(\hat{U}\hat{V}) = \sum_{\hat{u}=-\infty}^{+\infty} \sum_{\hat{v}=-\infty}^{+\infty} p(\hat{u},\hat{v}) \log \frac{1}{p(\hat{u},\hat{v})} \tag{3.16}$$

$$p(\hat{u},\hat{v}) = \sum_{u'=-\infty}^{+\infty} \sum_{v'=-\infty}^{+\infty} p(u,v)p(u',v') \tag{3.17}$$

$$\begin{aligned} H(\hat{U}\hat{V} \mid UV) &= \sum_{u=-\infty}^{+\infty} \sum_{u=-\infty}^{+\infty} p(u,v) H(\hat{U}\hat{V} \mid UV = (u,v)) \\ &= H(\hat{U}\hat{V} \mid UV = (u,v)) \sum_{u=-\infty}^{+\infty} \sum_{u=-\infty}^{+\infty} p(u,v) \\ &= H(U'V') \sum_{u=-\infty}^{+\infty} \sum_{u=-\infty}^{+\infty} p(u,v) \\ &= H(U'V') \end{aligned} \tag{3.18}$$

Refer to formula (3.16), (3.18), and (2.9), the mutual information can be calculated as:

$$\begin{aligned} I(UV;\hat{U}\hat{V}) &= H(\hat{U}\hat{V}) - H(\hat{U}\hat{V} \mid UV) \\ &= \sum_{\hat{u}=-\infty}^{+\infty} \sum_{\hat{v}=-\infty}^{+\infty} p(\hat{u},\hat{v}) \log \frac{1}{p(\hat{u},\hat{v})} + H(U'V') \end{aligned} \tag{3.19}$$

3.2 Computation of Measures in Gray-coded Tilings

At the beginning of this section, we will introduce our Gray-code mapping to transform the two tilings to bit-strings. Then we can show the bit-flips in the two tilings. With the bit-flips, we will compute the BSC capacity. At last, a roughly estimated mutual information, scores in bit-string, is calculated by the BSC capacity and entropy.

Normal Gray Code Mapping		Modified Gray Code Mapping	
Int	Gray	Int	Gray
0	000	-3	111
1	001	-2	101
2	011	-1	100
3	010	0	000
4	110	1	001
5	111	2	011
6	101	3	010
7	100	4	110

Table 3.3. Normal Gray-code mapping versus modified Gray-code mapping

In section 2.2, the Gray code has been introduced. Since the hexagonal tiling labels can be negative integers, the Gray-code mapping should be modified to fit negative integers. The n-bit modified Gray-code mapping is splited into two parts, one is to map the integers from 0 to 2^{n-1} and the other is to map the negative integers from $-2^{n-1}+1$ to -1. To be more specific, a normal n-bit Gray code table from top to bottom maps the integers from 0 to 2^{n-1}; in the modified Gray-code mapping, we keep the upper $2^{n-1}+1$ integers and change the rest integers bottom-up with the negative integers from -1 to $-2^{n-1}+1$. Table 3.3 shows an example of a 3-bit

normal Gray-code mapping versus a modified Gray-code mapping. From the right panel of table, we see that each pair of neighbor integers has hamming distance 1, which satisfies the Gray code definition. With such a Gray-code mapping, we can transform the tiling labels to bit-strings. Notice that afterwards when we refer to Gray code, we mean the modified Gray-code mapping instead of the normal one.

Afterwards, bit-flips are introduced to measure bit error probability. A bit-flip occurs by the noise, which switches a bit from 0 to 1 or 1 to 0. In the hexagonal tiling, the bit-flips of each hexagon is the sum of the bit-flips both on a-axis and on b-axis; in the square tiling, the bit-flips of each square is the sum of the bit-flips both on x-axis and on y-axis. Take an example of the bit-flips in a 6-bit Gray-coded hexagonal tiling, the bit-flips for hexagon $(0,0)$ flowing to $(1,1)$ are the sum of 1 bit on a-axis and 1 bit on b-axis, which is 2 bit-flips in total, see figure 3.8. Notice that only part of the hexagonal tiling is shown in figure 3.8. Another example of a 6-bit Gray-coded square tiling is shown in figure 3.9. The bit-flips for square $(0,0)$ flowing to $(1,2)$ have 1 bit-flip on x-axis and 2 bit-flips on y-axis, which add up to 3 bit-flips. Then the expectation value of the bit-flips is defined as follows:

$$E_{bit\text{-}flips\_hex} = \sum_{i=1}^{n} i \Pr\{hexagonal\ tiling\ i\ bit\text{-}flips\} \tag{3.20}$$

$$E_{bit\text{-}flips\_square} = \sum_{i=1}^{n} i \Pr\{square\ tiling\ i\ bit\text{-}flips\} \tag{3.21}$$

Here $\Pr\{hexagonal\ tiling\ i\ bit\text{-}flips\}$ and $\Pr\{square\ tiling\ i\ bit\text{-}flips\}$ are the probability of having i bit-flips in the Gray-coded hexagonal tiling and the Gray-coded square tiling, respectively.

Notice that $\sum_{i=1}^{n} \Pr\{hexagonal\ tiling\ i\ bit\text{-}flips\} = 1$ and $\sum_{i=1}^{n} \Pr\{square\ tiling\ i\ bit\text{-}flips\} = 1$.

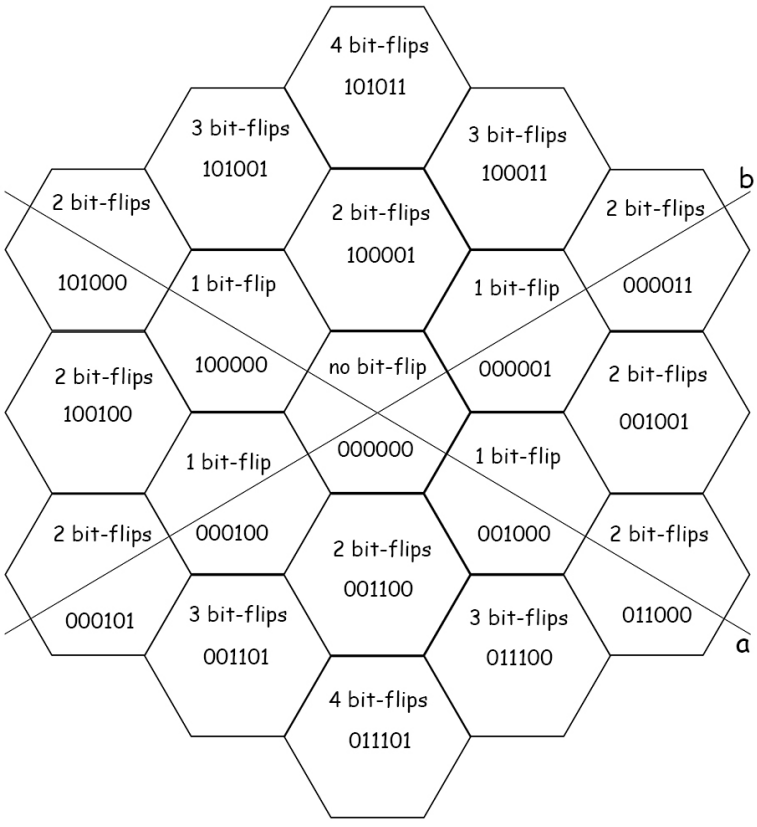

Figure 3.8. Bit-flips in the hexagonal tiling

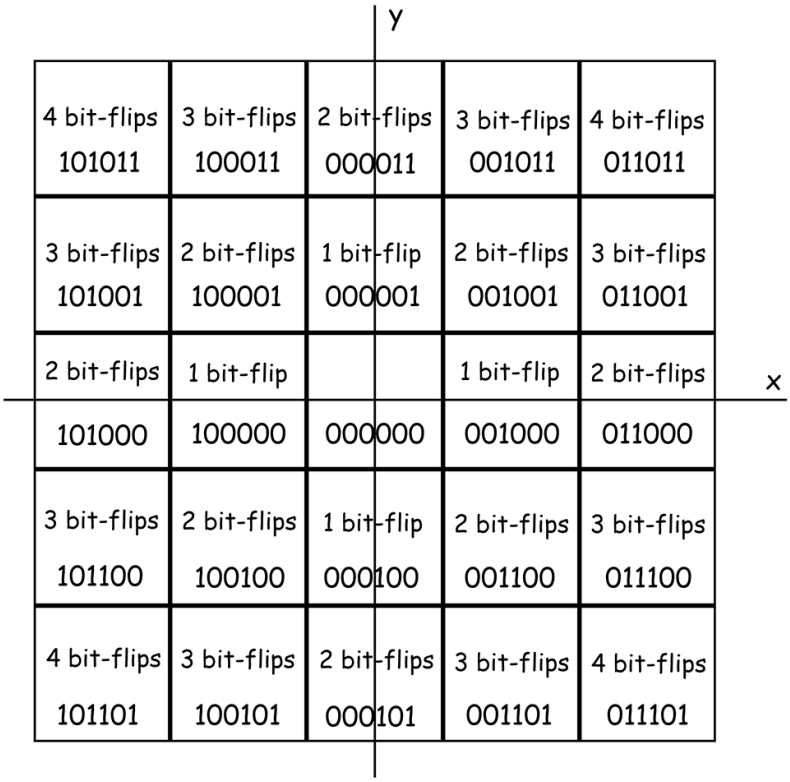

Figure 3.9. Bit-flips in the square tiling

In Gray code, when a random position is chosen and a random step is made, the probabilities that a bit 0 flips to bit 1 and a bit 1 flips to bit 0 are equivalent. Because of this, we will compute the bit error probability with two sorts of codes: one is a n-bit Gray code, which is the existing Gray code as we know; the other is a hypothetical “ideal” Gray code, which has the same bit-flips properties as the n-bit Gray code but has the entropy of the tiling as the code length. Here we propose the ”ideal” Gray code because it has no redundancy. It presents the use of some unknown Gray-like code that has better properties than known Gray codes. It allows us to study the performance issue under slightly different assumptions and see if it influences the conclusions. The bit error probabilities are formulated as follows:

$$p_{hex} = \frac{\sum_{i=1}^{n} i \Pr\{hexagonal\ tiling\ i\ bit\text{-}flips\}}{n} \tag{3.22}$$

$$p_{square} = \frac{\sum_{i=1}^{n} i \Pr\{square\ tiling\ i\ bit\text{-}flips\}}{n} \tag{3.23}$$

$$p_{hex\_ideal} = \frac{\sum_{i=1}^{n} i \Pr\{hexagonal\ tiling\ i\ bit\text{-}flips\}}{H(JK)} \tag{3.24}$$

$$p_{square\_ideal} = \frac{\sum_{i=1}^{n} i \Pr\{square\ tiling\ i\ bit\text{-}flips\}}{H(UV)}. \tag{3.25}$$

Refer to formula (2.11), the capacity of BSC can be presented as:

$$C_{hex} = 1 - h(p_{hex}) \tag{3.26}$$

$$C_{square} = 1 - h(p_{square}) \tag{3.27}$$

$$C_{hex\_ideal} = 1 - h(p_{hex\_ideal}) \tag{3.28}$$

$$C_{square\_ideal} = 1 - h(p_{square\_ideal}). \tag{3.29}$$

The computation of binary entropies can be found in formula (2.10).

According to [7,13], the BSC capacity is defined as the maximum achievable amount of information that can be reliably sent over the channel, counted per transmitted bit. Since the n-bit Gray code has redundancy, we use the entropy, which quantifies average number of bits per hexagon/square needed to encoded, as our effective code length instead of length n. Then the scores in bit-string is to multiply the BSC capacity with the effective code length, which is denoted as follows:

$$B_{hex} = H(JK)\, C_{hex} \tag{3.30}$$

$$B_{square} = H(UV)\, C_{square} \tag{3.31}$$

$$B_{hex\_ideal} = H(JK)\, C_{hex\_ideal} \tag{3.32}$$

$$B_{square\_ideal} = H(UV)\, C_{square\_ideal}. \tag{3.33}$$

4. Numerical Analysis

In this chapter, we perform a numerical analysis of the helper data scheme performance. In the biometric system, square tiling is a common way to discretize the data, while hexagonal tiling is a novel way we would like to investigate. To judge which tiling is better, we compare the performance of the two tilings. In section 4.1, We will explain our numerical analysis method in detail. Then the numerical analysis of the hexagonal/square tiling is given, with a preliminary conclusion over superiority and inferiority of the two tilings in section 4.2.

4.1 Method

In this section, we will introduce how we implement the numerical analysis to compare the performance of the two tilings. In order to achieve a fair comparison of performance, several assumptions are given in section 4.1.1. Then measures of the performance are listed followed by the optimization of the measure computations.

We have split the numerical analysis into two parts: one is theoretical, in which we only analyze the performance of the two tilings in geometry; and the other is the analysis based on the Gray-coded tilings with existing Gray code and hypothetical “ideal” Gray code.

4.1.1 Assumptions

Assumption 1: We assume that the enrolled data are the real physiological characteristics without noise; the user input data are the real physiological characteristics with noise, the noisy equivalent.

Assumption 2: To have a fair comparison of the measures between the two tilings, the computation area of the hexagonal/square tiling are defined, which is shown as a parallelogram in figure 4.1(a) and a square in 4.1(b). The labels j,k,u,v of the hexagon/square in the computation area are set separately to $[-3\sigma_X,\ \ 3\sigma_X]$ with the rounding to the closest integer.

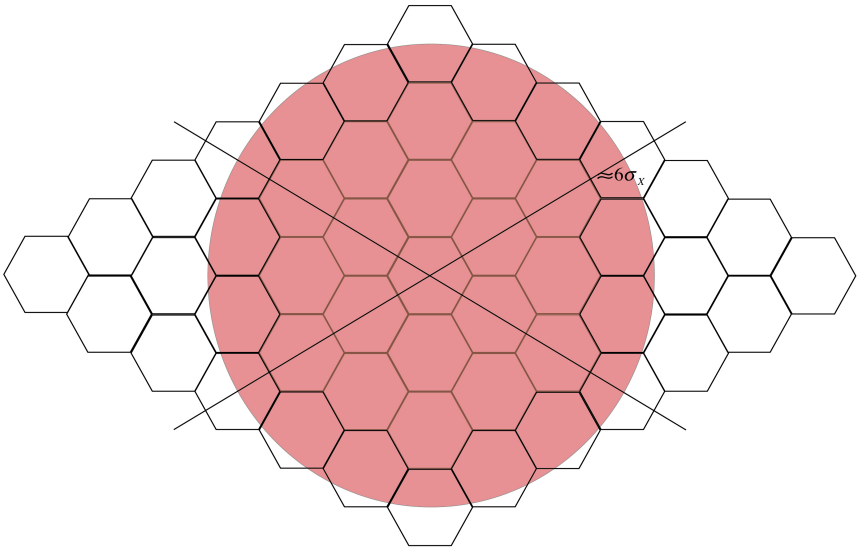

Figure 4.1(a). Gaussian distribution on the hexagonal tiling

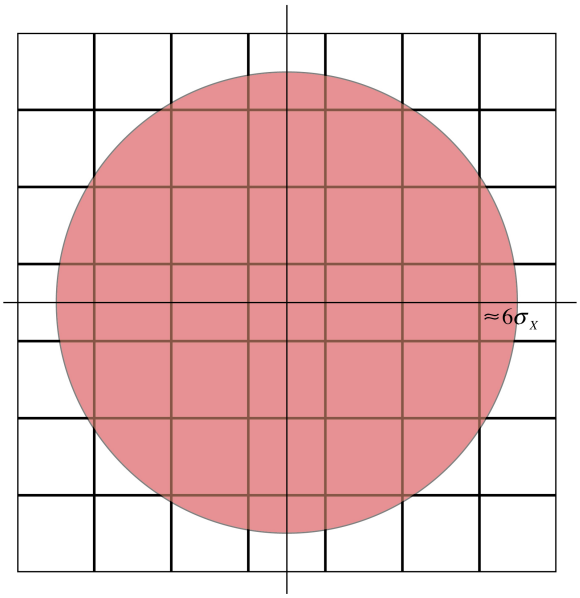

Figure 4.1(b). Gaussian distribution on the square tiling

Although we calculate the measures of the hexagonal/square tiling with different computation area, a fair comparison of the measures can be still achieved between the hexagonal tiling and the square tiling. That is because both computation areas have the same approximate area, which is marked as red circle in figure 4.1. The radius of the circle area on both tilings is equivalent to twice the rounding of $3\sigma_X$. In figure 4.1(a), the red circle exceeds the hexagonal tiling a bit. This exceeding area together with the hexagons out of the red circle are negligible in our computations. Hence, the computed measures of the parallelogram are almost the same as that of the red circle. In figure 4.1(b), the red area is entirely on the square tiling. Since the squares out of the red area are also negligible, the computations of the square can be approximated by that of the red circle. Therefore a fair comparison between the two tilings as we stated before can be reached.

Assumption 3: To calculate the measures, we assume that the noise only affects two rings of each hexagon/square and the rest of the noise far-away is negligible, see figure 4.2. Here, the labels and the bit-flips of two-ring hexagons/squares are listed in table 4.1(a) and 4.1(b), note that the labels are marked in the relative a-b/Cartesian coordinate system. Here we limit σ_N to range from 0 to 2 to make sure most of the noise are presented within two rings of each hexagon/square.

Hexagonal tiling label	Bit-flips(#Bits)
(1,0), (-1,0), (0,1), (0,-1)	1
(-1,1), (1,-1), (2,0), (-2,0), (0,2), (0,-2), (1,1), (-1,-1)	2
(2,-1), (1,-2), (-2,1), (-1,2)	3
(2,-2),(-2,2)	4

Table 4.1(a). Relative hexagon label and bit-flips of two rings

Square tiling label	Bit-flips(#Bits)
(1,0), (-1,0), (0,1), (0,-1)	1
(-1,1), (1,-1), (2,0), (-2,0), (0,2), (0,-2), (1,1), (-1,-1)	2
(1,2),(1,-2),(2,1),(2,-1),(-1,2),(-1,-2),(-2,1),(-2,-1)	3
(2,-2),(-2,2),(2,2),(-2,-2)	4

Table 4.1(b). Relative square label and bit-flips of two rings

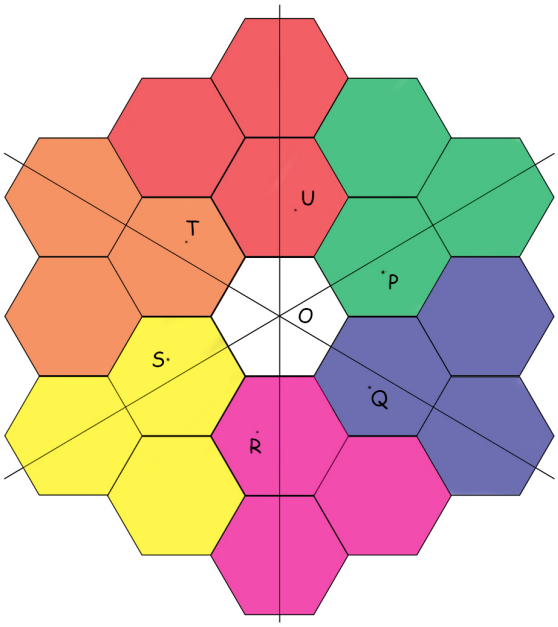

Figure 4.2(a). Split hexagonal tiling into 6 equivalents

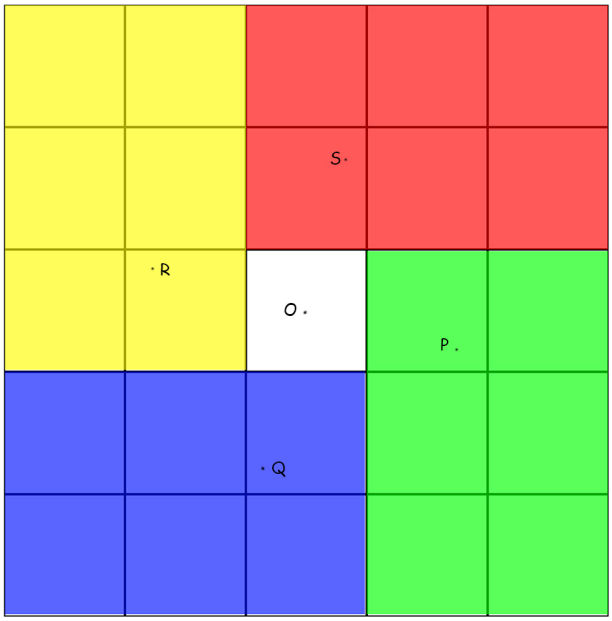

Figure 4.2(b). Split square tiling into 4 equivalents

Assumption 4: The BSC is used in our case to analyze the performance of the helper data scheme in binary biometric system. We assume that there exist an ideal binary error

correction code, which makes sure no errors occurring during the transformation from the hexagon/square labels to the bit-strings.

4.1.2 Measures of Hexagonal Tiling and Square Tiling

The measure of performance we use for the theoretical numerical analysis is mainly the mutual information, and we also compute the entropy of enrolled data and reconstructed noisy data as auxiliary evidence. In the binary numerical analysis, the measure of performance we use is the scores in bit-string. Overall, the measures that we have computed in Mathematica are listed below:

- The entropy of the hexagonal/square tiling without noise;
- The entropy of the hexagonal/square tiling with the Gaussian noise;
- The maximum mutual information versus different SNR σ_X/σ_N.
- The maximum scores versus different SNR σ_X/σ_N.

Recall section 3.2, n-bit Gray code have a mapping range $[-2^{n-1}+1, \quad 2^{n-1}]$. However, in assumption 2, we have defined the numerical analysis area, within which the labels are symmetric. Then when we use Gray code to map the tiling labels to bit-strings, the n-bit Gray code can not be fully used, which leads to the bit error probability on each bit not equivalent any more. Take an example of the hexagon label that j,k ranges from -3 to 3, the Gray code 110 will never be used. For such cases, replacing the crossover probability with the bit error probability is not accurate. However, we assume that this deviation is small enough to be neglected in our computation.

4.1.3 Optimization

To optimize the computations of measures in Mathematica, the hexagonal tiling is splited into 6 equivalents and the square tiling is splited into 4 equivalents, shown in figure 4.2.

Hexagonal Tiling - The hexagonal tiling is splited into six regions in different colors. For a point located in one of the six regions, there exist the matching points in other five regions which has the same distance to the origin O. For example, in figure 4.2(a), for point P, there exist Q, R, S, T, U in other five regions that the distance OP, OQ, OR, OS, OT, OU are the same. To be concrete, if one of the regions rotates 60 degree, it will cover the neighbor regions thoroughly. In the two-dimensional Gaussian distribution, the distance to the origin is the only determinant of the probability. Since our measures are all based on the probabilities, the six regions can be deemed as equivalents, which can make the computations 6 times faster.

Square tiling - Similar to the hexagonal tiling, the square tiling is splited into four regions, shown in figure 4.2(b). For a point located in one of the four regions, there exist the matching points in other three regions which has the same distance to the origin O. Rotating each region 90 degrees results in complete overlapping of the neighbor regions. Regarding to the property of the two-dimensional Gaussian distribution, the four regions can be deemed as equivalents.

4.2 Numerical Analysis

In this section, the entropy, mutual information, and scores of the hexagonal tiling are computed and further analyzed by comparing to the square tiling. The mutual information and the scores are the two criteria to measure the helper data scheme performance as we stated before.

4.2.1 Comparison between Hexagonal Tiling and Square Tiling Theoretically

We first present the results of comparison between the two tilings theoretically. Figure 4.3 shows the entropy of enrolled data, the entropy of reconstructed noisy data with helper data applied, and the mutual information of the two tilings when σ_N is set to 1. The horizontal axis σ_X ranges from 0 to 20; the vertical axis shows the entropy/mutual information in bits; the blue solid line is the entropy of reconstructed noisy data in the hexagonal tiling; the blue dashed line is the entropy of the enrolled data in the hexagonal tiling; the red solid line is the entropy of the reconstructed noisy data in the square tiling; the red dashed line is the entropy of the reconstructed noisy data in the square tiling; the green solid line is the mutual information of the hexagonal tiling; the black dashed line is the mutual information of square tiling. Here the horizontal axis is in logarithmic scale.

From the figure, we can find that the entropies of the hexagonal tiling are always larger than the square tiling regardless whether noise exists or not, whereas the mutual information for both tilings are almost the same. Also we observe that the entropy of the reconstructed noisy data is getting closer to the entropy of the enrolled data, while σ_X is increasing. This is because the noise has less effect on the hexagonal/square tiling with the increasing SNR.

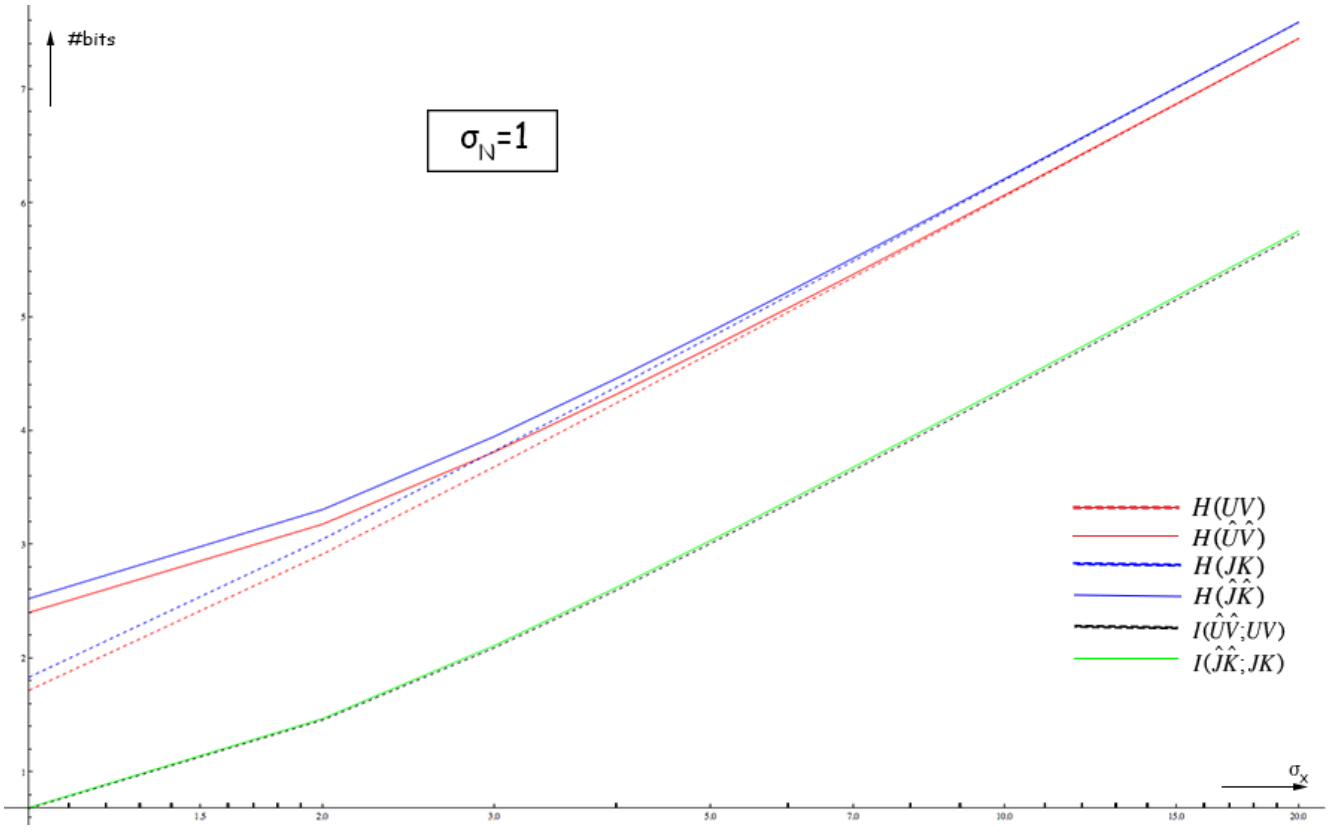

Figure 4.3. Entropy/mutual information versus σ_X

The comparison of the mutual information between the hexagonal tiling and the square tiling with the fixed SNR 20 is plotted in figure 4.4. The horizontal axis σ_N ranges from 0 to 2; the vertical axis is the mutual information and the conditional entropy; the blue line is the mutual information of the hexagonal tiling; the red line is the mutual information of the square tiling; the green line is the conditional entropy of the hexagonal tiling; the black line is the conditional entropy of the square tiling.

In the figure, both of the red line and the blue line increase rapidly at first, with the blue line slightly over the red line. Then both lines keep almost flat at the same level. When σ_N continuously increases, the mutual information of the square tiling decreases slightly, while the mutual information of the hexagonal tiling has a clear decline. The maximum mutual information conveys the maximum information that can be extracted from noisy data, which can indicate the best performance of the two tilings. The maximum mutual information for the hexagonal tiling which appears at σ_N=1.3 is very close to that of the square tiling which appears at σ_N=1.6.

For both tilings, the large increment of the mutual information appears with the σ_N ranging from 0 to 0.3. We also find that both the conditional entropies are approximately 0, when the σ_N ranges from 0 to 0.3. Recall formula (3.11), the mutual information is almost equal to the entropy of the reconstructed noisy data, which increases rapidly when σ_X and σ_N grow. The mutual information should continuously grow without the presence of noise; however, when noise exists, the conditional entropy also grows, which slows down the increment of the mutual information and results in a nearly flat curve. Especially when the increment of the conditional entropy is larger than the increase of the entropy of the reconstructed noisy data, the mutual information of the hexagonal tiling descends in the end.

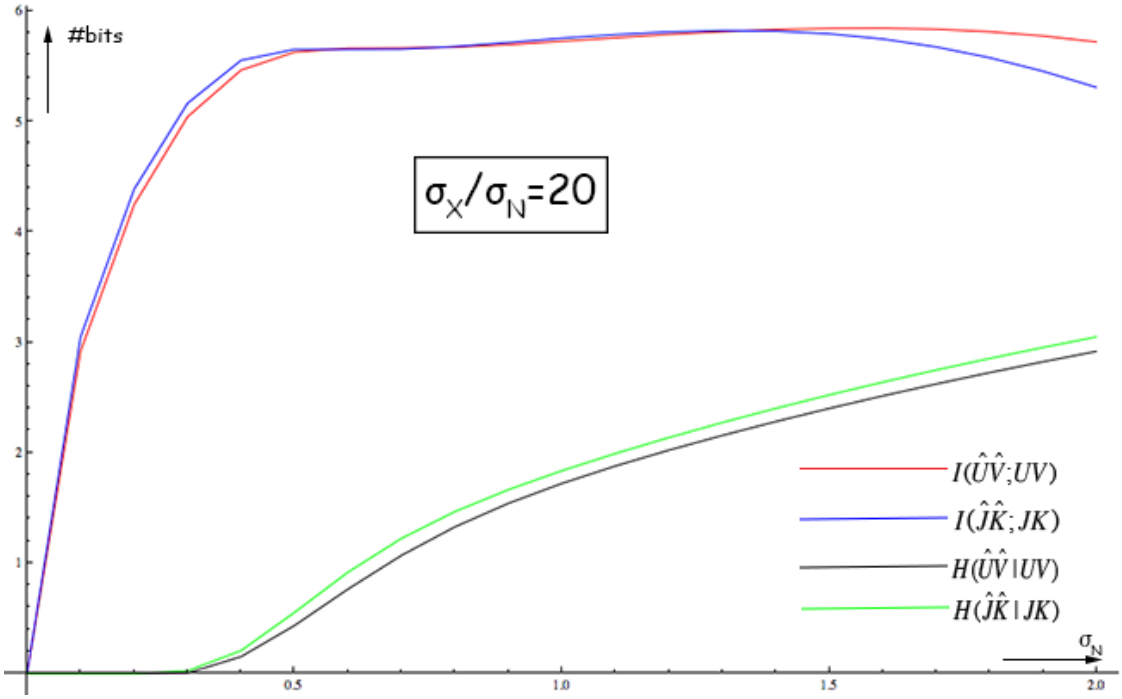

Figure 4.4. Mutual information with SNR 20 versus σ_N

In figure 4.5, the comparison of the mutual information between the two tilings versus σ_N is given, over several SNRs. The horizontal axis σ_N ranges from 0 to 2; the vertical axis is in number of bits; the blue lines are the mutual information of the hexagonal tiling with different SNRs; the red lines are the mutual information of the square tiling with different SNRs.

The same as in figure 4.4, the hexagonal tiling and the square tiling in this figure are very close to each other with each SNR; both lines increase at almost the same speed at first with blue lines slightly over the red lines; and then reach almost flat at the same levels; while the red lines continue to keep more or less flat, the blue lines give significant drops in the end. According to these lines, the difference of the mutual information between the two tilings are consistent over different SNRs. When σ_N is small, the performance of the hexagonal tiling is slightly better than the square tiling; when σ_N continuously increase, the performances of the two tilings are firstly comparable and then the performance of the square tiling is more stable and thus superior to the hexagonal tiling.

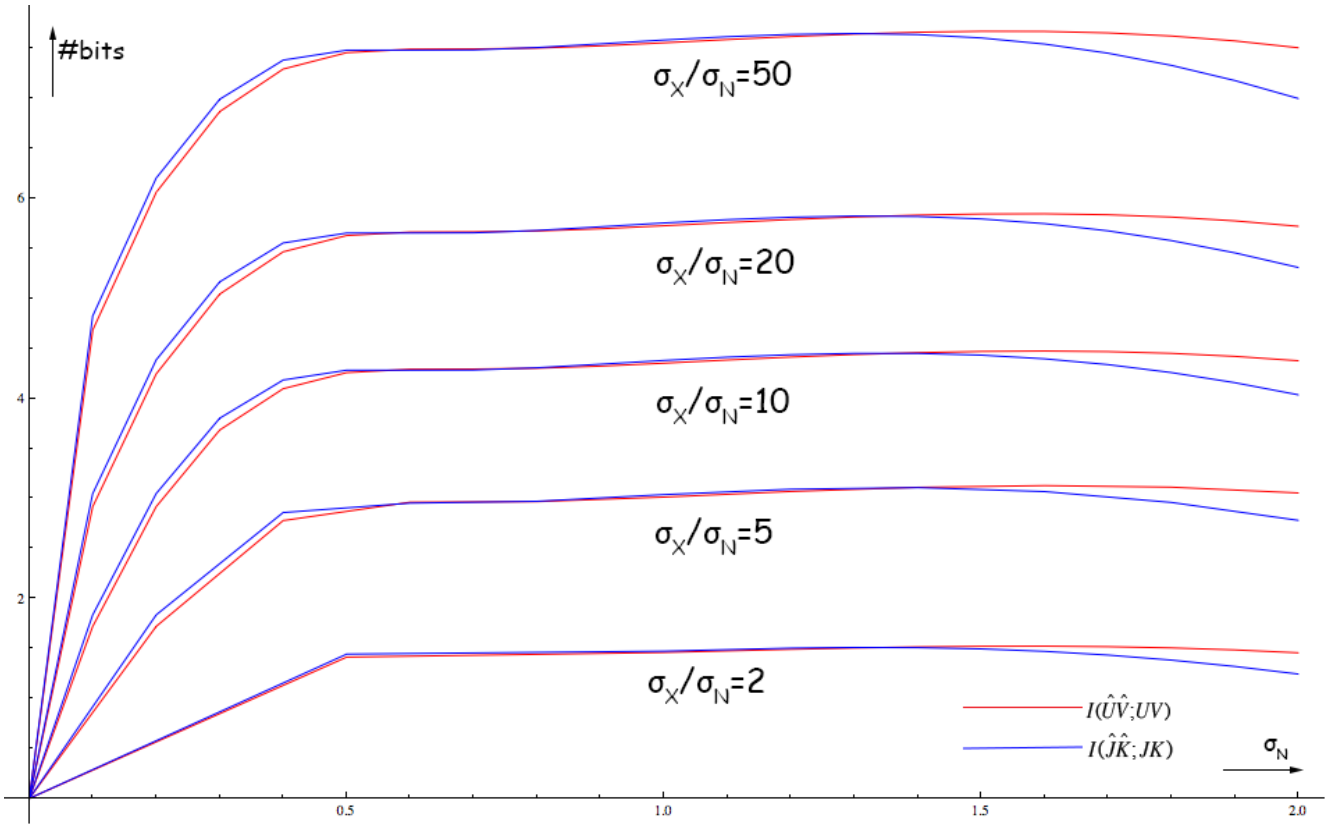

Figure 4.5. Mutual information with several SNRs versus σ_N

Next, a more important result is shown in figure 4.6. The maximum mutual information of the hexagonal tiling and the square tiling with different SNRs are plotted in these two figures. The horizontal axis is SNR σ_X/σ_N ranging from 0 to 50 in figure 4.6(a), and zoomed in to ranging from 0 to 2 in figure 4.6(b). Here in both figures, the vertical axes are the maximum mutual information in bits; the blue solid line indicates the hexagonal tiling while the red dash line indicates the square tiling.

In figure 4.6(a), we can hardly distinguish the difference of the mutual information between the hexagonal tiling and the square tiling. When SNR continuously increases, the mutual information of both tilings also increases, with the ascending speed slowing down gradually.

In figure 4.6(b), it is still hard to distinguish the difference of mutual information between the two tilings. We have listed the the maximum mutual information of the two tilings versus different SNRs in detail in table 4.2. It is observed that the square tiling always has slightly larger mutual information than the hexagonal tiling except for SNR 0.8. The difference is very small, which only appears at the second decimal or the third decimal. Notice that we only precise the the maximum mutual informations to 3 decimal places.

SNR σ_X/σ_N	$\max_{\sigma_N} I_{hex}$	Hexagonal tiling opt for σ_N	$\max_{\sigma_N} I_{square}$	Square tiling opt for σ_N
0.5	0.216	1.3	0.222	1.7
0.8	0.522	1.1	0.521	1.3
1	0.691	1.1	0.692	1.2
1.2	0.857	1.2	0.862	1.4
1.4	1.027	1.2	1.034	1.5
1.5	1.111	1.3	1.112	1.5
1.6	1.194	1.3	1.204	1.5
1.8	1.356	1.3	1.366	1.5
2	1.510	1.3	1.523	1.6
5	3.108	1.4	3.128	1.6
10	4.451	1.3	4.474	1.6
20	5.823	1.3	5.846	1.6
50	7.648	1.3	7.671	1.5

Table 4.2 *Samples of maximum mutual information versus different SNRs*

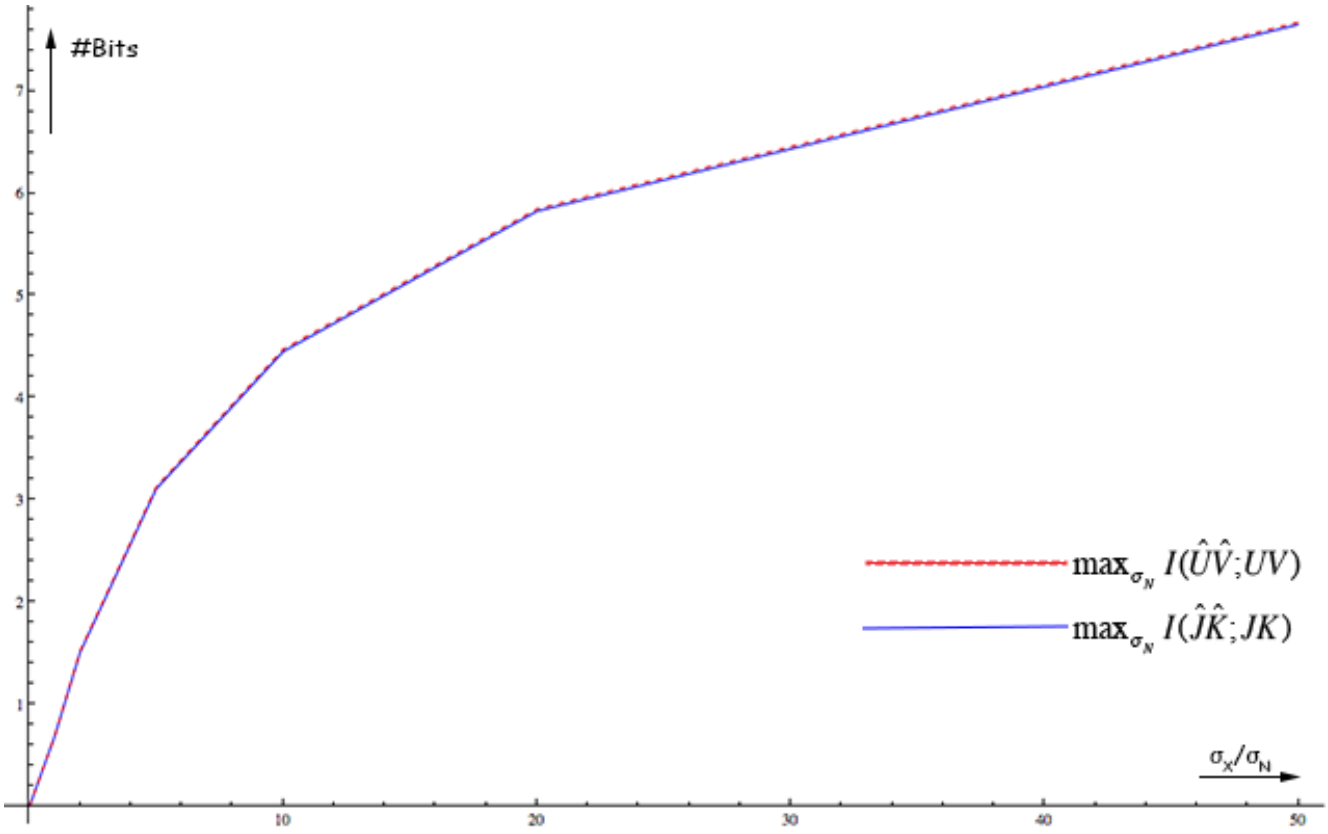

Figure 4.6(a). Maximum mutual information versus SNR 0 to 50

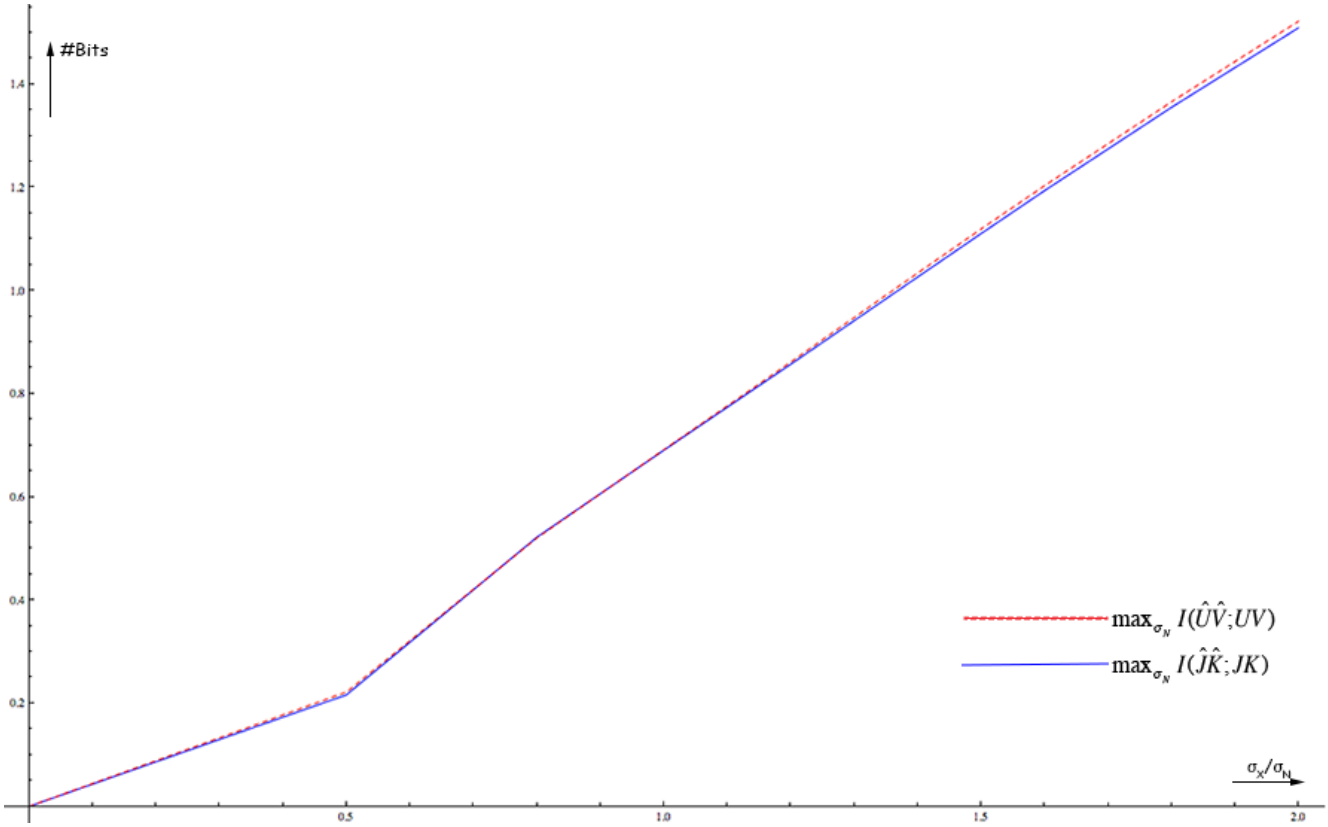

Figure 4.6(b). Maximum mutual information versus SNR 0 to 2

In the numerical analysis in theory, we have the following observations. The entropy of the hexagonal tiling is larger than the square tiling. For small σ_N, the hexagonal tiling has slightly larger mutual information than the square tiling; in the middle range of σ_N, the two performance of the two tilings are comparable; for large σ_N, the square tiling has more stable and larger mutual information than the hexagonal tiling. Furthermore, the maximum mutual information of the two tilings are very close with the square tiling slightly larger than the hexagonal tiling. To sum up, the square tiling has slightly better overall performance than the hexagonal tiling in theory.

4.2.2 Comparison between Gray-coded Hexagonal Tiling and Gray-coded Square Tiling

In this section, we present the results of comparison between the two Gray-coded and "ideal" Gray-coded tilings in bit-string. Notice that scores of the two Gray-coded tilings are defined by formula (3,30) and (3.31); and scores of the "ideal" Gray-coded tilings are defined by formula (3.32) and (3.33). We will first look at the scores of the Gray-coded tilings over different SNRs, shown in figure 4.7. The horizontal axis σ_N ranges from 0 to 2; the blue lines are the scores of the Gray-coded hexagonal tiling with different SNRs; the red lines are the scores of the Gray-coded square tiling with different SNRs.

In the figure, both red lines and blue lines increase fast at first with the blue lines slightly over the red lines; and then the lines becomes nearly flat with the red lines much higher than the blue lines. Here the flat parts can be separated by several jumps, which are resulted from the discontinuity of the code length increase. This results indicate that the hexagonal tiling has slightly better performance than the square tiling with small σ_N and the square tiling has better performance than the hexagonal tiling with large σ_N, which is in accord with what we have

observed in figure 4.5 with the superiority of the performance of the square tiling over the hexagonal tiling enlarged.

According to three-sigma rule, most of the noise of each hexagon/square are distributed in the center tile when σ_N is small. Here when σ_N is smaller than 0.5, about 95% of the noise presents in the center tile. In this case, the bit error probabilities approach 0, which leads the BSC capacities close to 1. Then the scores of the Gray-coded tilings almost equal to the entropy of the tilings. Recall what we observed in figure 4.3, the entropy of the hexagonal tiling is always larger than the square tiling, which leads the scores of the Gray-coded hexagonal tiling larger than the square tiling. However, when σ_N is larger, the noise of the hexagon/square will spread out of the center tile. The expectation of the bit-flips increases with the increment of σ_N. Notice that the expectation value of bit-flips will never exceeds 0.5, thus the function of binary entropy is monotonically increasing. Then the BSC capacities will decrease accordingly. Although the entropies of the tilings are still increasing, the decreasing BSC capacities will slow down the increasing speed of the scores of the Gray-coded tilings. When the decrement of BSC capacities are more dominant, the crossover of two curves happens. The several jumps in the flat curves are because of the discontinuous code length increase. When the code length increases, the bit error probabilities will decrease suddenly, which results in the abrupt increment of the BSC capacity. The increasing BSC capacity multiplies the increasing entropy, producing the jumps in the score curves.

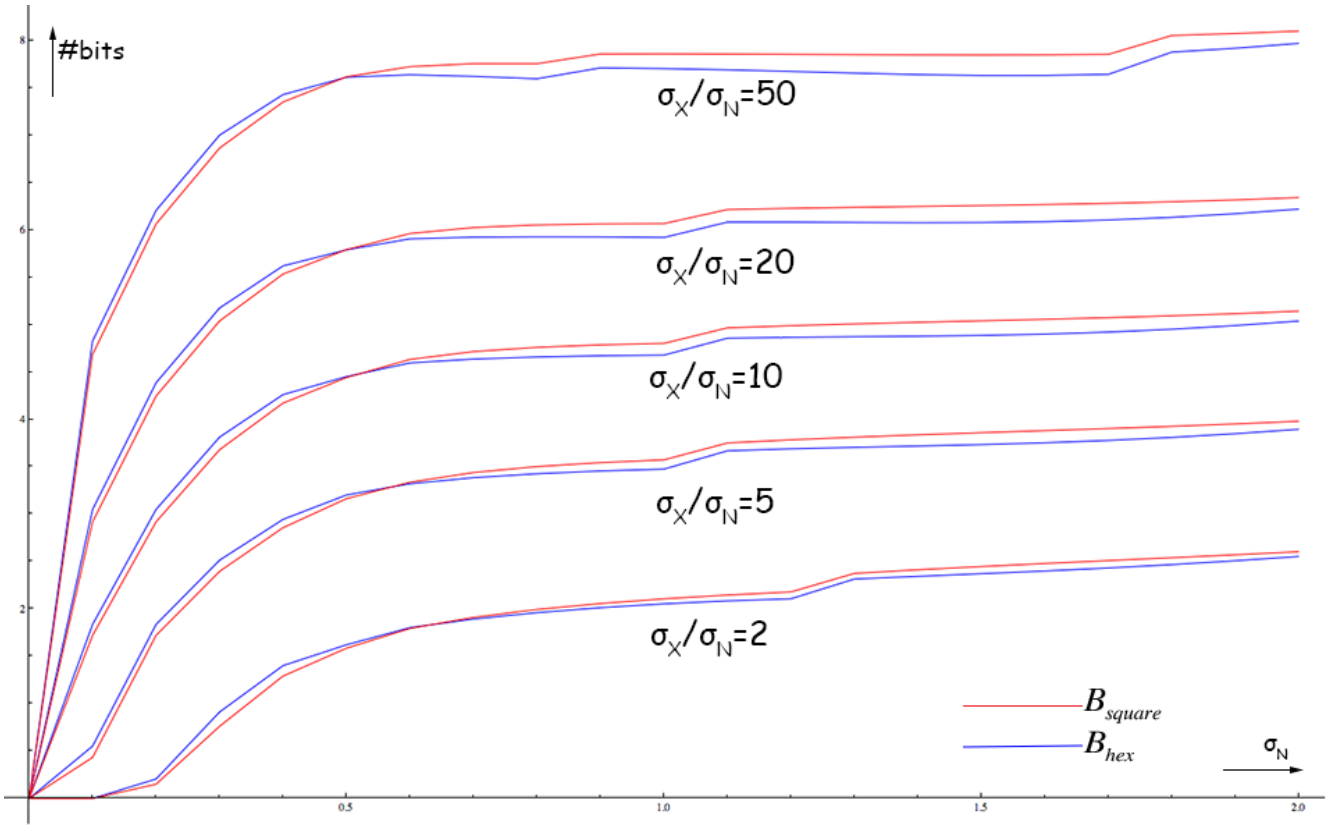

Figure 4.7. Scores of the Gray-coded tilings with several SNRs versus σ_N

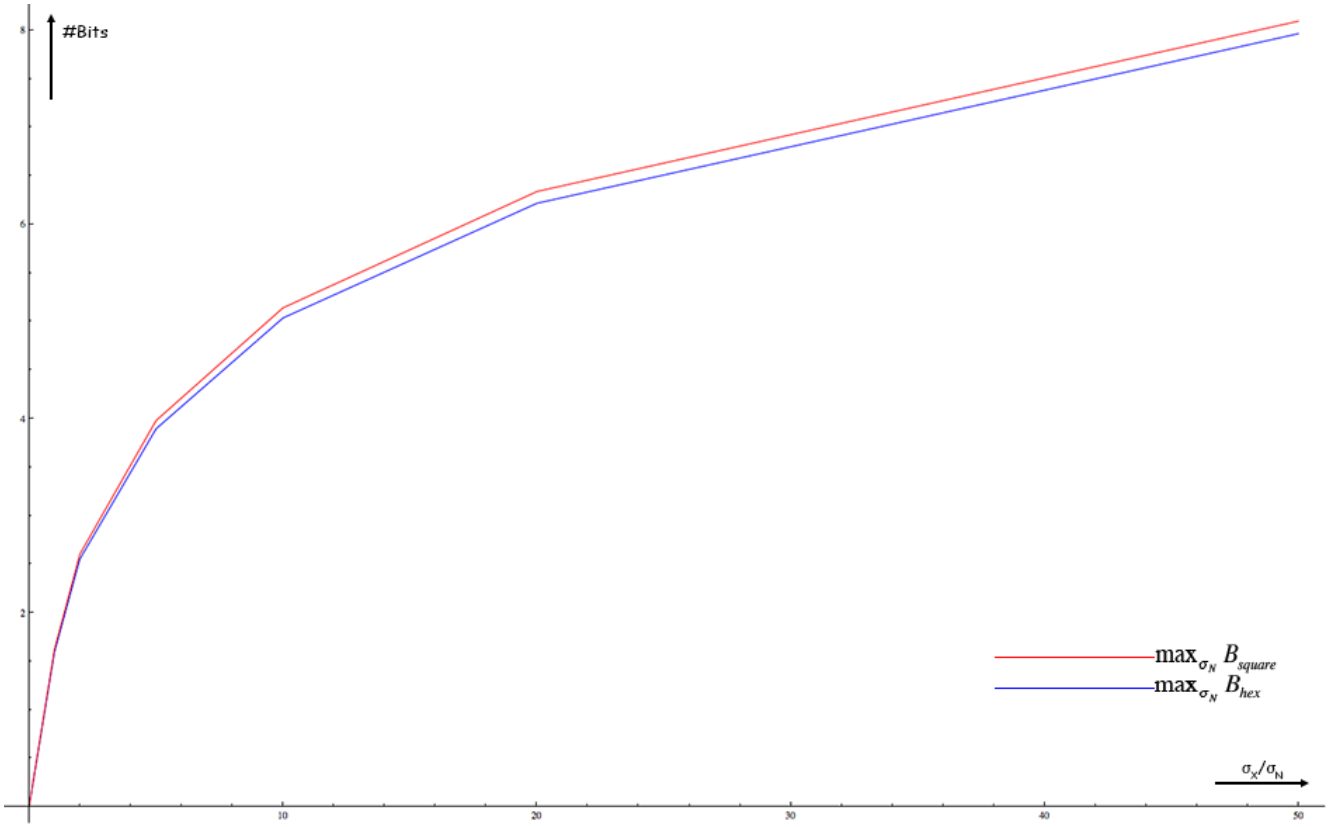

Figure 4.8(a). Maximum scores of Gray-coded tilings versus SNR 0 to 50

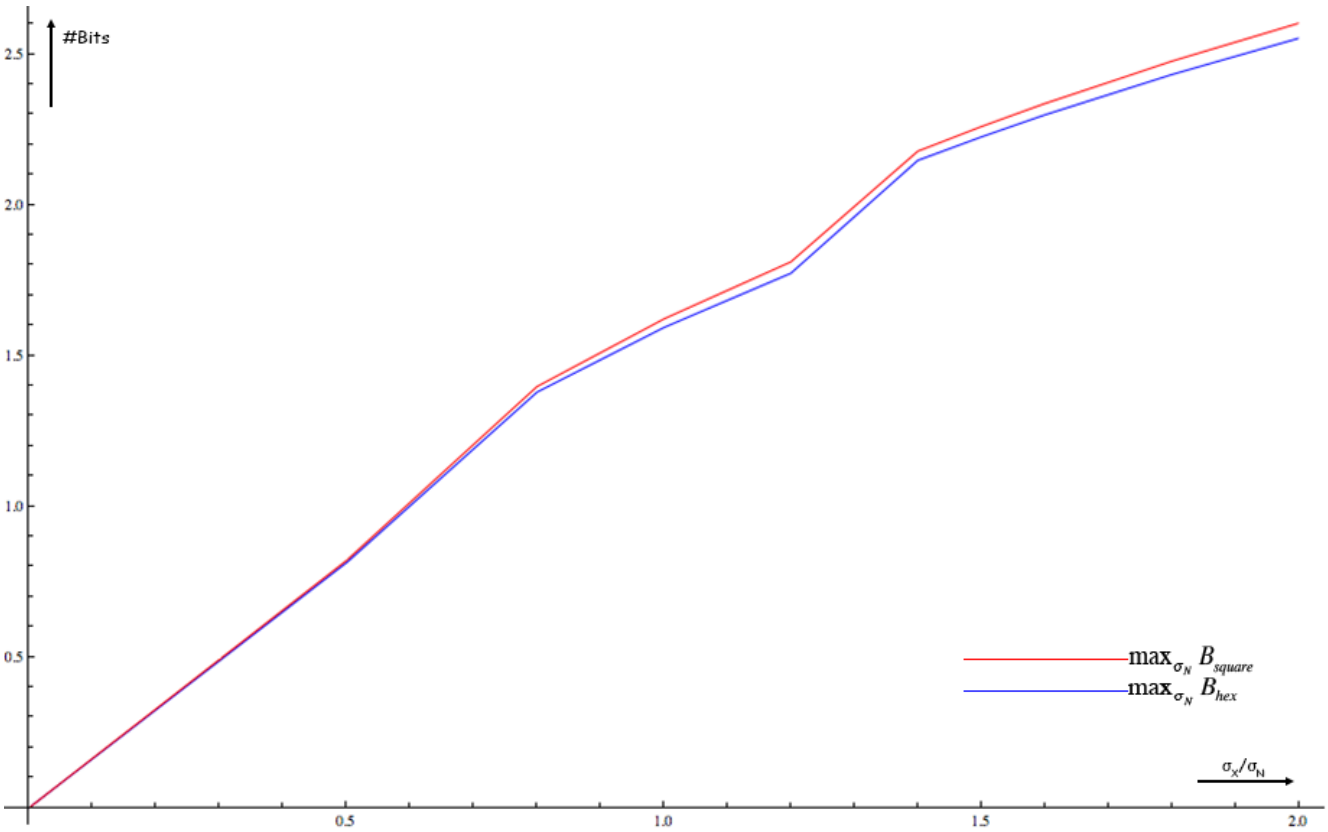

Figure 4.8(b). Maximum scores of Gray-coded tilings versus SNR 0 to 2

The maximum scores of the two Gray-coded tilings versus different SNRs are shown in figure 4.8. The horizontal axis is the SNR ranging from 0 to 50 in figure 4.8(a), zoomed in with SNR ranging from 0 to 2 in figure 4.8(b). In these two figures, the blue lines and the red lines are the maximum scores of the Gray-coded hexagonal tiling and square tiling, respectively.

In figure 4.8(a), it can be observed that the maximum scores of the hexagonal tiling is smaller than the square tiling; further observed in figure 4.8(b), the difference between the hexagonal tiling and the square tiling is not apparent until SNR reaches 0.8. This observation is also consistent with figure 4.6, and the superiority of the square tiling over the hexagonal tiling is enlarged. We have listed the the maximum scores of the two tilings versus different SNRs in detail in table 4.3. Same as table 4.2, the accuracy of the samples are 3 digits decimal. From these tables, the difference of the maximum scores between the hexagonal tiling and the square tiling mainly appears at the first decimal or the second decimal. Here, similar as table 4.2, we only list the maximum scores of the Gray-coded tiling and the "ideal" Gray-coded tilings with 3 decimal digits accuracy.

SNR σ_X/σ_N	$\max_{\sigma_N} B_{hex}$	$\max_{\sigma_N} B_{square}$
0.5	0.813	0.821
0.8	1.379	1.397
1	1.593	1.622
1.2	1.773	1.811
1.4	2.148	2.179
1.5	2.226	2.260
1.6	2.299	2.336
1.8	2.433	2.477
2	2.553	2.603
5	3.895	3.979
10	5.036	5.141
20	6.217	6.339
50	7.966	8.095

Table 4.3. Samples of maximum scores (Gray-coded tilings) versus different SNRs

Next we look at the scores of the "ideal" Gray-coded tilings over different SNRs in figure 4.9. The horizontal axis σ_N ranges from 0 to 2; the blue lines corresponds to the scores of the "ideal" Gray-coded hexagonal tiling with several SNRs; the red lines corresponds to the scores of the "ideal" Gray-coded square tiling with several SNRs.

In the figure, both red lines and blue lines increase fast at first with the blue lines slightly higher than the red lines; and then both lines gradually decline with the red lines significantly higher than the blue lines. These two curves reflect that the hexagonal tiling has slightly better performance than the square tiling with small σ_N and the performance of the square tiling is better than the hexagonal tiling with large σ_N, which further corroborates what we have observed in figure 4.5 and 4.7.

In “ideal” Gray code, the bit error probability is the expectation of bit-flips dividing the entropy of the tiling index instead of the Gray code length n. Since the entropy is always smaller than the code length n, the score of the “ideal” Gray-coded tiling is smaller than that of the Gray-coded tiling, which is also corroborated by comparing figure 4.9 to figure 4.7. Also the increment of entropy is continuous, so the curves of “ideal” Gray code have a smooth behavior without the jumps occurring in figure 4.7.

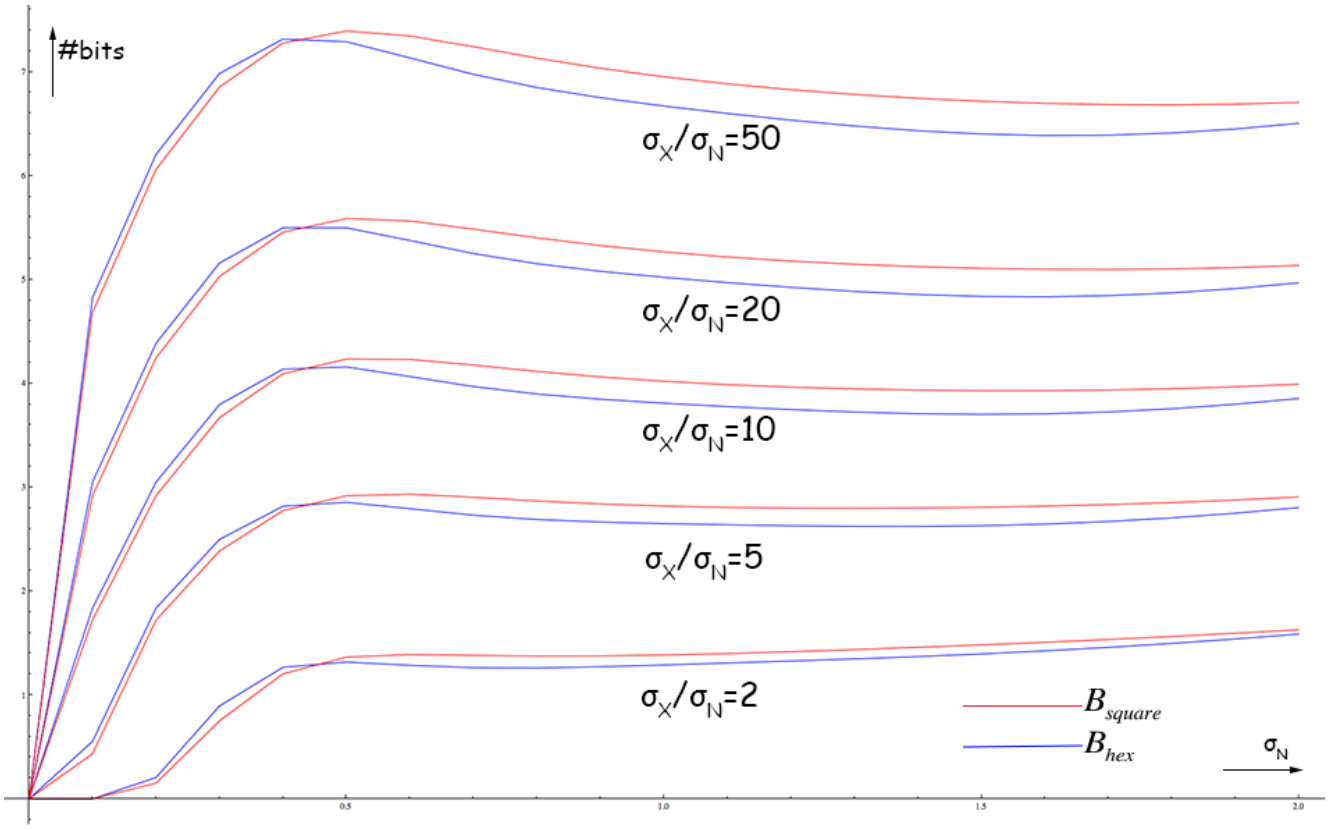

Figure 4.9. Scores of an “ideal” Gray-coded tilings with several SNRs versus σ_N

Figure 4.10 shows the maximum scores with “ideal” Gray code. The horizontal axis is the SNR ranging from 0 to 50 in figure 4.10(a), zoomed in with SNR ranging from 0 to 2 in figure 4.10(b). The blue lines and the red lines represent the maximum scores of the “ideal” Gray-coded hexagonal tiling and square tiling, respectively.

In figure 4.10(a), it can be observed that the maximum scores of the “ideal” Gray-coded hexagonal tiling is in general smaller than the square tiling; further observed from figure 4.10(b), when SNRs are smaller than 1.3, the maximum scores of the “ideal” Gray-coded hexagonal tiling exceeds the square tiling with very small difference. This observation is consistent with what have been indicated in figure 4.6: the square tiling has superior performance to the hexagonal tiling. We have listed the maximum scores of the two tilings versus different SNRs in detail in table 4.4. In this table, the difference of the maximum scores between the hexagonal tiling and the square tiling mainly takes place at the second decimal.

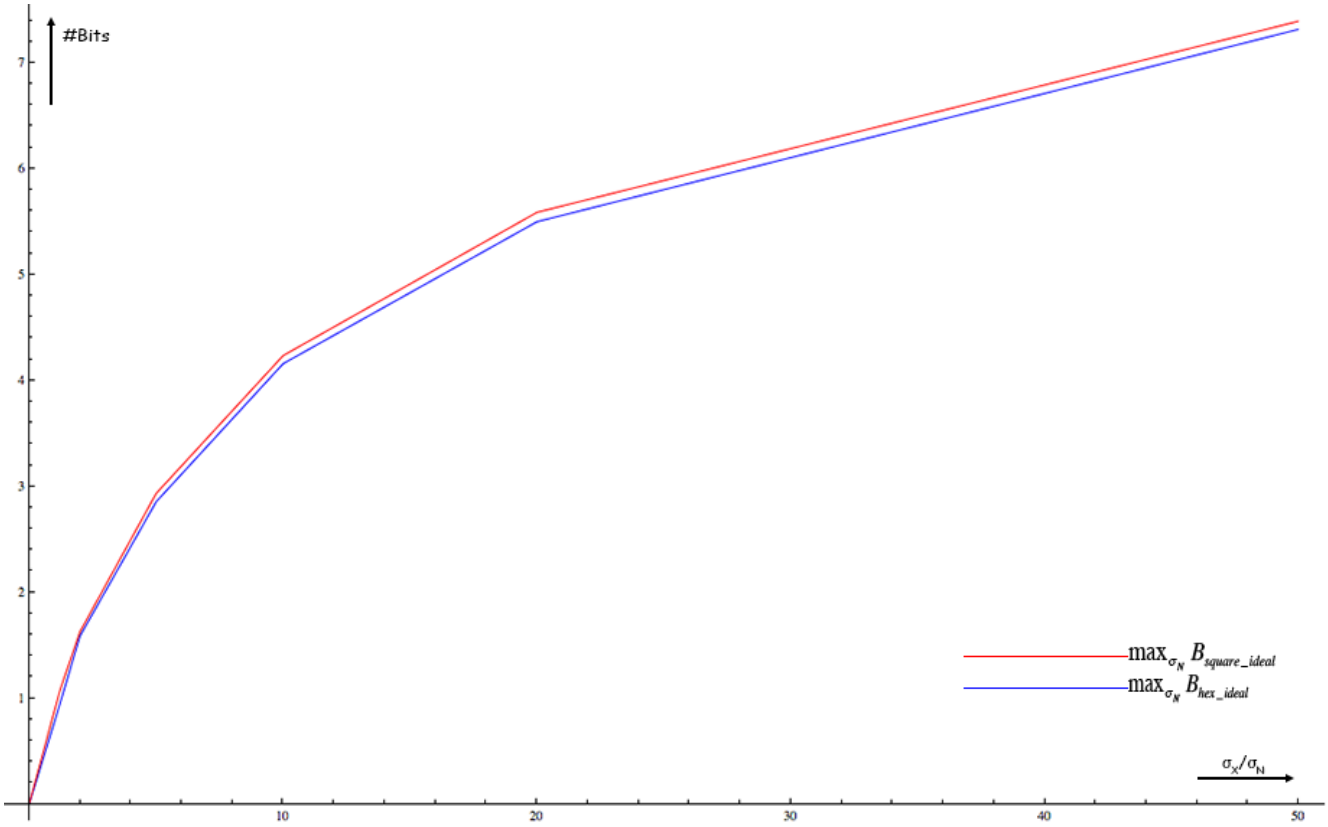

Figure 4.10(a). Maximum scores of "ideal" Gray-coded tilings versus SNR 0 to 50

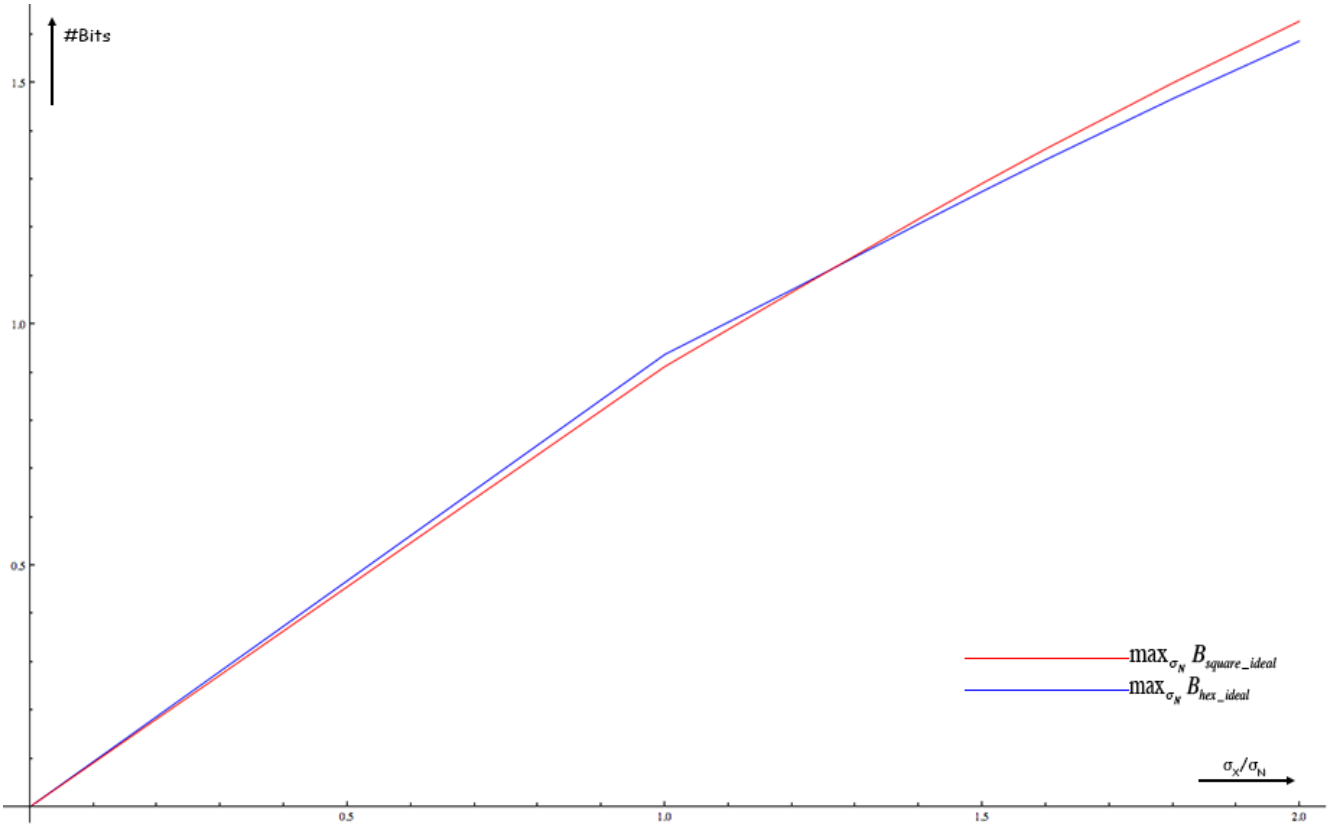

Figure 4.10(b). Maximum scores of "ideal" Gray-coded tilings versus SNR 0 to 2

SNR σ_X/σ_N	$\max_{\sigma_N} B_{hex}$	$\max_{\sigma_N} B_{square}$
1	0.937	0.912
1.2	1.071	1.066
1.4	1.208	1.218
1.5	1.275	1.292
1.6	1.341	1.363
1.8	1.467	1.500
2	1.587	1.628
5	2.855	2.933
10	4.159	4.235
20	5.499	5.588
50	7.315	7.393

Table 4.4. Samples of maximum scores ("ideal" Gray-coded tilings) versus different SNRs

In conclusion, the numerical analysis based on maximum scores of the Gray-coded tilings and the "ideal" Gray-coded tilings further demonstrates that the performance of the hexagonal tiling is comparable or slightly poorer than the square tiling.

5. Conclusion and Future Work

In this thesis, we studied a helper data scheme for a two-dimensional Gaussian source discretized with hexagonal tiling. This kind of scheme can be used for privacy-preserving biometrics applications and Physical Unclonable Functions. We did a numerical analysis based on four assumptions, see section 4.1.1, to compare the performance of the helper data scheme based on the two tilings.

The performance measures are defined in two ways. One is the mutual information between the enrolled and reconstructed tile index. It provides an upper bound on the length of a bit string that can be robustly extracted from the source by any ideal error correction procedure.

The other performance measure is an rough estimate of the mutual information after applying a Gray code to the tile coordinates. It is assumed that errors in the Gray-coded coordinates behave according to the Binary Symmetric Channel. The performance measure provides an approximate upper bound on the size of the robust bit string under the assumption that an ideal binary error correction code is used.

Our numerical analysis shows that the performance of the helper scheme with hexagonal tiling is comparable to or slightly poorer than the square tiling interms of the first performance measure; in terms of the second one, the hexagonal tiling demonstrates a comparable or slightly poorer performance than the square tiling. These results are contrary to our expectations, given the efficiency of the hexagonal tiling as known from the literatures.

The following two points are proposed that the future work could focus on. The first one is to use new method for discretization of the continuous data. In [17], a soft-discretization, which is based on fuzzy set theory, is introduced to discretize the associated attribute of continuous-valued data. This soft-discretization in our view is a possible solution and worth a try in the future. The second one is to apply other codes instead of the Gray code. Since the Gray code we applied is one-dimensional binary code, the bit-flips of the Gray code is not optimal for hexagonal tiling. One could try to find a new binary or non-binary code to improve the helper data scheme.

Acknowledgments

The author shows his deepest gratitude to his supervisor Boris Škorić for his patient guidance and intellectual support. Further, the author also appreciates all the spiritual supports from his families and friends.

References

[1] Jain, A. K., Ross, A., and Prabhakar, S.. (2004). An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology 14(1): 4-20.

[2] Tuyls, P., B. Škorić, and T.A.M. Kevenaar, Security with noisy data : on private biometrics, secure key storage and anti-counterfeiting, 2007, London: Springer. xv, 339 p.

[3] Pappu, R. and Recht, B. and Taylor, J. and Gershenfeld, N., Physical One-way Functions, 2002, Science. 297(5589): p. 2026-30.

[4] Yamazato, T., Oshita, S., Sasase, I., and Mori, S.. An Arrangement Technique of Gray-Code Table for Signal Constellation of Modified QAM and Triangular-Shaped Signal Set, 1991, IEICE. p. p.2579-2585.

[5] Buhan, I., Doumen, J., Hartel, P., Veldhuis, R.. Constructing practical fuzzy extractors using QIM, Centre for Telematics and Information Technology, University of Twente, 2007, Enschede, Technical Report TR-CTIT-07-52.

[6] Bryc, W. (1995). The normal distribution : characterizations with applications. New York, Springer-Verlag.

[7] Cover, T. M. and J. A. Thomas (2006). Elements of information theory. Hoboken, N.J., J. Wiley.

[8] Morelos-Zaragoza, R. H. (2006). The Art of error correcting coding. Chichester, Wiley.

[9] F. Gray. Pulse code communication, March 17, 1953

[10] Madhow, U. (2008). Fundamentals of digital communication. Cambridge, Cambridge University Press.

[11] Linnartz, J. P. and P. Tuyls (2003). New shielding functions to enhance privacy and prevent misuse of biometric templates, Springer.

[12] Dodis, Y., Ostrovsky, R., Reyzin L., Smith, A.. (2004). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, Springer.

[13] B. Škorić. Physical aspects of digital security Lecture notes, 2011.

[14] Juels, A. and M. Sudan. A fuzzy vault scheme. Designs, Codes and Cryptography, 2006. 38(2): p. 237-257.

[15] Sadeghi, A.R. and D. Naccache. Towards Hardware-Intrinsic Security: Foundations and Practice, 2010, Springer-Verlag New York Inc.

[16] Juels, A. and M. Wattenberg (1999). A fuzzy commitment scheme, ACM.

[17] Peng, Y., Flach, P. A., Soft Discretization to Enhance the Continuous Decision Tree Induction, Integrating Aspects of Data Mining, Decision Support and Meta-Learning. Christophe Giraud-Carrier, Nada Lavrac, Steve Moyle, (eds.), pp. 109–118. September 2001.