Efficient key generation leveraging wireless channel reciprocity and discrete cosine transform

Furui Zhan[1]*, Nianmin Yao[1]
[1]School of Computer Science and Technology, Dalian University of Technology
Dalian, China
[e-mail: izfree@mail.dlut.edu.cn]
*Corresponding author: Furui Zhan

Received August 15, 2016; revised December 22, 2016; revised February 9, 2017; accepted February 22, 2017; published May 31, 2017

Abstract

Key generation is essential for protecting wireless networks. Based on wireless channel reciprocity, transceivers can generate shared secret keys by measuring their communicating channels. However, due to non-simultaneous measurements, asymmetric noises and other interferences, channel measurements collected by different transceivers are highly correlated but not identical and thus might have some discrepancies. Further, these discrepancies might lead to mismatches of bit sequences after quantization. The referred mismatches significantly affect the efficiency of key generation. In this paper, an efficient key generation scheme leveraging wireless channel reciprocity is proposed. To reduce the bit mismatch rate and enhance the efficiency of key generation, the involved transceivers separately apply discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) to pre-process their measurements. Then, the outputs of IDCT are quantified and encoded to establish the bit sequence. With the implementations of information reconciliation and privacy amplification, the shared secret key can be generated. Several experiments in real environments are conducted to evaluate the proposed scheme. During each experiment, the shared key is established from the received signal strength (RSS) of heterogeneous devices. The results of experiments demonstrate that the proposed scheme can efficiently generate shared secret keys between transceivers.

Keywords: Key generation, wireless channel reciprocity, discrete cosine transformation, received signal strength, heterogeneous devices

https://doi.org/10.3837/tiis.2017.05.022 **ISSN : 1976-7277**

1. Introduction

Secret key generation is implemented to establish secret keys for protecting wireless networks. Traditional solution for generating key is achieved by public key cryptography, which is based on the computation complexity of some problems, such as factoring the product of two large prime numbers or discrete logarithm problem. However, such solution generally requires public key infrastructure (PKI) and consumes too many resources. Therefore, it is inappropriate for resources-constraint sensor networks and mobile networks without key management center.

Comparing with the traditional method, key generation leveraging wireless channel reciprocity is a promising way for establishing shared secret keys between transceivers. For this solution, three factors significantly affect the performance of key generation [1], i.e., temporal variation, channel reciprocity and spatial de-correlation of wireless channel. Channel reciprocity guarantees the involved transceivers experience same or approximately the same channel state when they measure the channel within the coherence time [2]. Spatial de-correlation ensures that adversaries located more than half of a wavelength away from either user experience statistical independent channel states. Therefore, these two factors are essential requirements of key generation. In contrast, temporal variation significantly influences the efficiency of key generation. On the one hand, temporal variation reflects the uncertainty of channel measurements, which determines the rate of generating key and the quality of the generated key; on the other hand, according to existing works, temporal variation indirectly affects the reciprocity of measurements of different transceivers and thus influences the bit mismatch rate after quantization and encoding.

Several key generation schemes were proposed based on wireless channel reciprocity. These schemes typically consist of four components: 1) channel probing; 2) quantization; 3) information reconciliation; 4) privacy amplification.

1) Channel probing

During channel probing stage, the involved transceivers collect channel measurements by continuously probing the communicating channel. In each pass, both transceivers have to measure the channel within the coherence time. Several statistics can be used to exploit the channel state, such as channel impulse response (CIR), received signal strength (RSS) and channel state information (CSI). Taking into account the implementation complexity and availability of the off-the-shelf devices, most of existing schemes extract keys from RSS.

2) Quantization

After collecting channel measurements, the involved transceivers separately use their quantizers to convert measurements into bits. Furthermore, if the multi-bits quantizer is applied to quantify measurements, source coding is required to encode the generated symbols from the quantizer. Consequently, each transceiver can generate a bit sequence.

3) Information reconciliation

Generally, two bit sequences are not identical since measurements of two transceivers have some discrepancies. Therefore, information reconciliation needs to be implemented to correct these errors and generate the shared key. Several methods can be used to achieve information reconciliation, such as Cascade [3], low-density parity-check (LDPC) [4] and Golay code [5]. During information reconciliation, some information has to be exchanged to correct errors. As

a result, these information might be leaked to the adversary. Therefore, this stage significantly affects the efficiency and security of the key generation process.

4) Privacy amplification

After information reconciliation, privacy amplification is employed to eliminate the leaked information so that the adversary cannot guess the generated key by these information. Several methods were proposed to achieve privacy amplification, such as fuzzy extractors [6] and secure hash algorithm [7]. With the implementation of privacy amplification, a highly random and uniformly distributed secret key can be generated to protect the communication between transceivers.

As mentioned above, measurements of the involved transceivers might have some discrepancies, which might produce mismatches after quantization. Consequently, the efficiency of key generation is reduced. In this work, we propose an efficient key generation scheme leveraging wireless channel reciprocity and discrete cosine transform (DCT). Different from existing schemes, discrete cosine transform and inverse discrete cosine transform (IDCT) are used to pre-process measurements in the proposed scheme. Then, the outputs of IDCT are used as the shared secrets between transceivers. Further, uniform multi-bits quantization in conjunction with gray code is applied to convert these shared secrets to bits. During information reconciliation, distillation [8] and Cascade [3] are implemented to correct errors between two bit sequences. Finally, 2-universal hash functions are used to guarantee the randomness of the generated key.

To validate the proposed scheme, several experiments in real environments are conducted. These experiments are accomplished by heterogeneous devices, since communication between heterogeneous devices is more complicated but broader used than homogeneous ones. Moreover, RSS measurements are used to exploit channel states. The results demonstrate that: 1) when the outputs of IDCT instead of measurements are used for generating bit sequences, two bit sequences possess lower bit mismatch rate; 2) the proposed scheme can efficiently generate shared secret keys between transceivers. The results of NIST randomness tests [9] show that the generated key can be used to protect the communication between transceivers.

The remainder of this paper is organized as follows: Section 2 reviews several key generation schemes leveraging wireless channel reciprocity. In Section 3, system model and adversary model are described. RSS measurements collected in the experiments are depicted in Section 4. We illustrate the proposed key generation scheme in Section 5. In Section 6, performance of the proposed scheme is evaluated. Finally, we conclude our work in Section 7.

2. Related Work

During channel probing, various statistics can be used to exploit channel states, such as RSS, CIR and CSI. In contrast to other statistics, most of existing schemes use RSS to generate key, since it can be easily derived from the off-the-shelf devices without any modification. In this section, several related schemes are introduced by two parts: 1) RSS based approaches; 2) other approaches.

2.1 RSS based approaches

In [10], a key generation scheme was proposed to extract secret keys from signal envelope. Further, deep fades rather than the entire envelopes were used to establish the shared key. To detect deep fades, narrow-band filter and threshold detector were used in this scheme.

Moreover, secure fuzzy information reconciliator was applied as randomness extractor to generate uniformly random bits.

Mathur et al. proposed a key generation scheme leveraging wireless channel reciprocity in [11]. In this scheme, a level-crossing algorithm with two thresholds was applied to establish shared secret keys between transceivers. In order to evaluate this scheme, CIR and RSS were derived for generating secret keys in different experiments, respectively. The results of experiments demonstrated that secret bits without any errors can be established at a considerate rate.

In [8, 12], an environment adaptive scheme was proposed to establish secret keys from RSS variations on the wireless channel. In this scheme, the involved transceivers separately divided their measurements into smaller blocks and adaptively quantified each block. Several experiments in a variety of environments were conducted to validate this scheme. The results showed that variations of environment significantly affected the efficiency of key generation. Moreover, a multi-bits key generation scheme was also proposed to enhance the rate of generating key.

To achieve key generation in static scenarios, a scheme that extracted secrets from RSS was proposed in [13]. Frequency-selectivity of multipath fading channels was applied to ensure channel variations during key generation. The experiment results showed that this scheme can successfully generate keys in over 97% of the cases in realistic settings.

Instead of pairwise key generation, a collaborative secret key extraction scheme was proposed to establish group key in mobile networks [14]. To guarantee the security of key generation, differences of RSS measurements at each device were sent to neighbors for generating the group key.

In addition, several schemes were also proposed to establish shared secret keys from RSS [15–17]. These schemes were validated either by simulations or experiments in real environments.

2.2 Other approaches

In [18] and [19], phase reciprocity of frequency selective fading channels was used to establish shared secret keys between transceivers. In [20], a practical key generation scheme based on channel frequency response was proposed. To ensure that the channel fading was frequency selective, a broadband chaotic signal was employed for transmission. Angle-of-Arrival (AOA) was derived as a signature for key generation in [21]. This scheme required an access point to have a programmable phased array antenna. The ultra-wideband (UWB) channels were applied to realize key generation in [22, 23]. In [24], a new fuzzy key generation method based on PHY-Layer fingerprints was proposed for mobile cognitive radio networks. Moreover, several schemes were proposed to extract keys from CIR [11, 25]. Physical-Layer secret key agreement in two-way wireless relaying systems was discussed in [26]. Recently, CSI was also applied to generate secret key in several schemes [27, 28]. To derive CSI, the involved transceivers were typically required to equip the wireless cards supporting 802.11n.

Some existing schemes have been validated by experiments in real environments. The results demonstrate that measurements collected by different transceivers are highly correlated but not identical. As a result, bit sequences extracted from these measurements might have many mismatches. To generate a shared key from these bit sequences, some information has to be exchanged during information reconciliation, which significantly influences the efficiency

and security of key generation. Therefore, it is valuable to pre-process measurements to reduce their discrepancies such that the bit mismatch rate after quantization can also be reduced.

3. System model & Adversary model

3.1 System model

According to wireless channel reciprocity, the involved transceivers experience identical channel states when they simultaneously measure the channel. However, most of the off-the-shelf communication systems are half duplex, i.e., the transceiver can only either transmit or receive information at any given time. Then, these transceivers can derive same or approximately the same channel state when they measure the channel within the coherence time. The mentioned situations can be described as follows:

$$r_{Bob}(\mathrm{t}) = \mathrm{s}_{Alice}(\mathrm{t}) * h_{AB}(\mathrm{t}) + n_{AB}(t) \tag{1}$$

$$r_{Alice}(\mathrm{t}+\tau) = \mathrm{s}_{Bob}(\mathrm{t}+\tau) * h_{BA}(\mathrm{t}+\tau) + n_{BA}(t+\tau) \tag{2}$$

In this case, Alice and Bob denote the involved transceivers. The signals $s(t)$ and $r(t)$ are the probing signal and the corresponding received signal. The time interval of the transmitting processes of different transceivers is defined as τ, which is required to be smaller than the coherence time. In addition, $h_{AB}(t)$ and $h_{BA}(t+\tau)$ are channel responses of these probing processes. Then, we can find

$$h_{AB}(\mathrm{t}) \cong h_{BA}(\mathrm{t}+\tau) \tag{3}$$

In addition, assume that another transceiver Eve is listening to the mentioned probing processes, which can be described as

$$r_{Eve}(\mathrm{t}) = \mathrm{s}_{Alice}(\mathrm{t}) * h_{AE}(\mathrm{t}) + n_{AE}(t) \tag{4}$$

$$r_{Eve}(\mathrm{t}+\tau) = \mathrm{s}_{Bob}(\mathrm{t}+\tau) * h_{BE}(\mathrm{t}+\tau) + n_{BE}(t+\tau) \tag{5}$$

According to the spatial de-correlation, if Eve is more than half of a wavelength away from either Alice or Bob, $h_{AE}(t)$ and $h_{BE}(t+\tau)$ are individually statistical independent with $h_{AB}(t)$ and $h_{BA}(t+\tau)$. As a result, $h_{AB}(t)$ and $h_{BA}(t+\tau)$ can be used as the shared secret to establish the shared key between Alice and Bob.

3.2 Adversary model

For key generation with wireless channel reciprocity, sufficient channel measurements need to be collected to generate the key of desired length. Moreover, some information might be also exchanged in other phases during key generation. Therefore, attacks of the adversary might significantly affect the implementation of key generation. In this case, Alice and Bob are assumed as legitimate transceivers, while Eve is the adversary. Then, the involved transceivers are considered as follows:

1) Alice and Bob have already been authenticated, i.e., the authentication of transceivers is not considered in this work. Furthermore, Eve cannot launch some attacks, e.g., the person-in-the-middle attack.

2) Eve can monitor the communication between Alice and Bob. Moreover, Eve can also measure the channels.
3) Eve knows all implemented algorithms as well as their parameters during key generation.
4) The communication between Alice and Bob cannot be constantly disrupted by Eve, since channel measurements are collected by continuously probing the channel. In addition, information reconciliation also requires multiple passes of interactions between legitimate transceivers.
5) To guarantee the spatial de-correlation, Eve cannot be very close (half of the wavelength) to either Alice or Bob.

4. Measurement

4.1 The setting of experiments

To validate the feasibility of key generation, several experiments were conducted in a campus dormitory building. **Fig. 1** shows the layout of the referred campus dormitory building. During each experiment, laptops with different wireless cards were used as the transmitter (Alice) and receiver (Bob), respectively. The involved transceivers connected each other in an ad-hoc mode and formed a peer-to-peer connection. Similar to [11], ICMP ping was implemented to continuously probe channel and collect sufficient channel measurements. The packets were sent from the transmitter to the receiver at a rate of 20 per second. Moreover, RSS was used as the statistic of the channel, which can be derived from the Radiotap header of each received packet.

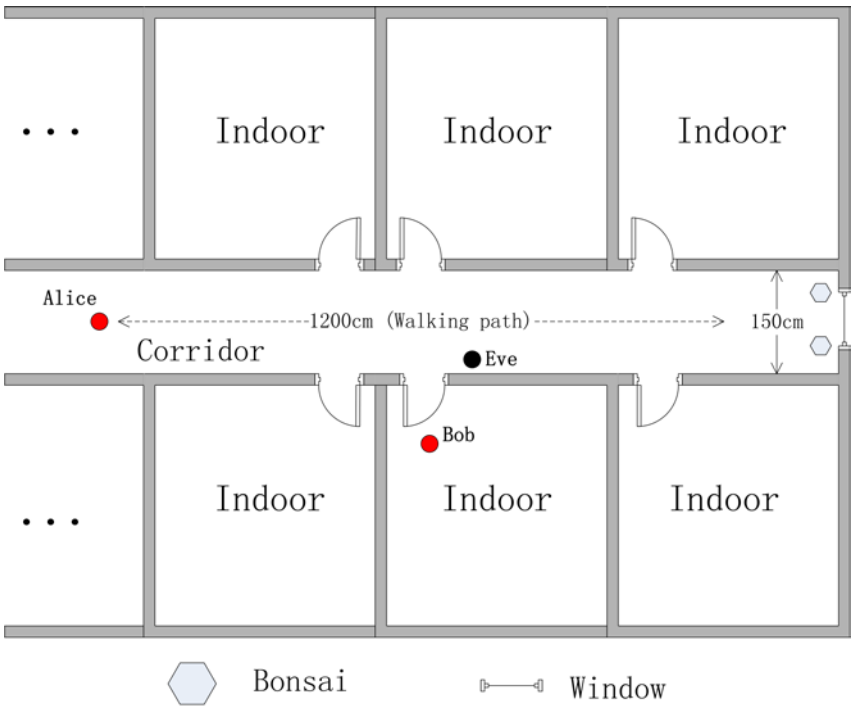

Fig. 1. Layout of the campus dormitory building

4.2 Measurements

Seven experiments were conducted in either static or mobile scenarios. **Table 1** describes wireless cards and mobility model in each experiment. The static scenario is free from the effect of mobility of any objects in the environment. Moreover, the transmitter and receiver are separated by a distance of about 2.5m during the experiment. In mobile scenarios, either the movements of the involved transceivers or the movements of intermediate objects are required to ensure channel variations during channel probing. For example, although the transmitter (Alice) and receiver (Bob) are stationary in scenario (c), some persons keep moving between these transceivers during this experiment.

Fig. 2 shows RSS measurements collected in these experiments. Obviously, we can find that measurements collected in mobile scenarios have wider variations than the static scenario. In addition, RSS measurements of different transceivers have different ranges of variation, since the involved wireless cards are different. Moreover, channel variations in the static scenario are mainly caused by hardware imperfections and thermal effects which are nonreciprocal [12].

Table 1. Transceivers and mobility model in each experiment

Scenario	Transmitter	Receiver	Mobility
(a)	Intel wireless N-2230	Atheros ar9285	Mobile/Stationary
(b)	Intel wireless N-2230	Atheros ar9285	Mobile/ Stationary
(c)	Atheros ar9285	Intel wireless 2200bg	Stationary/Mobile/Stationary
(d)	Atheros ar9285	Intel wireless 2200bg	Stationary/Stationary
(e)	Intel wireless N-2230	Atheros ar9285	Mobile/Mobile
(f)	Intel wireless N-2230	Intel wireless 2200bg	Mobile/Stationary
(g)	Intel wireless N-2230	Intel wireless 2200bg	Mobile/Stationary

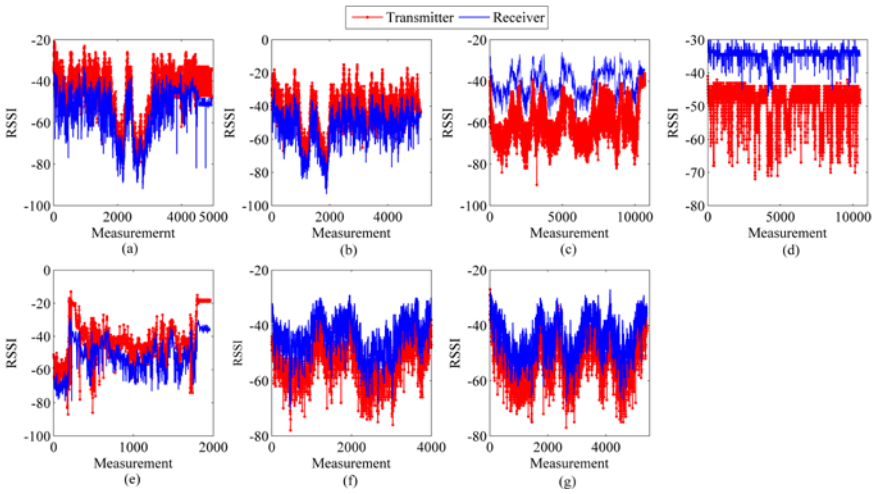

Fig. 2. RSS measurements collected in different experiments

To evaluate the reciprocity of measurements in each experiment, both mutual information and Spearman correlation coefficient are figured out. Mutual information measures the mutual dependence between two variables. Therefore, most existing schemes use mutual information to evaluate key generation rate and secrecy capacity. Assume that measurements collected by transmitter and receiver are X and Y, the mutual information between these variables is

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{6}$$

In formula (6), $H(X)$, $H(Y)$ and $H(X,Y)$ denote marginal entropies and joint entropy of X and Y. In this case, the referred entropies are Rényi entropies of order 2. Therefore, the mutual information is described as

$$I_2(X;Y) = H_2(X) + H_2(Y) - H_2(X,Y) \tag{7}$$

Where

$$H_\alpha(X) = \frac{\log\left(\sum_{i=1}^{n} p_i^\alpha\right)}{1-\alpha}, \alpha = 2 \tag{8}$$

In formula (8), $p_i\,(1 \le i \le n)$ is the probability of the i_{th} element in the alphabet of X. Similarly, the joint Rényi entropy can also be calculated based on the joint probability distribution of X and Y.

In contrast, Spearman correlation coefficient is a measure of rank correlation, which can be described as

$$r_s = \frac{\mathrm{cov}(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{9}$$

Where rg_X and rg_Y are rank variables of X and Y. $\mathrm{cov}(rg_X, rg_Y)$ is the covariance of the rank variables. σ_{rg_X} and σ_{rg_Y} are the standard deviations of rank variables. Spearman correlation coefficient is used to evaluate the reciprocity of measurements in several existing schemes.

Table 2. Evaluation of the measurements in each experiment

	Mutual information	**Spearman correlation coefficient**
(a)	1.9268	0.6011
(b)	2.4360	0.6291
(c)	1.3239	0.7717
(d)	0.7744	0.3831
(e)	2.2663	0.7136
(f)	2.4457	0.8293
(g)	2.0543	0.8394

Table 2 illustrates the mutual information and Spearman correlation coefficient in each experiment. We can find that the mutual information and Spearman correlation coefficient of the measurements in static scenario is greatly smaller than other scenarios, which indicates the reciprocity of measurements is not good. However, although the entire measurements in

mobile scenarios are highly correlated, they are not identical and have many discrepancies. Taking the measurements in the experiment (g) for example, the entire measurements and the sub-measurements ranging from 1601 to 1650 are illustrated in **Fig. 3-(a)** and **Fig. 3-(b)**, respectively. As illustrated, the involved transceivers are the laptops with wireless cards Intel wireless-N 2230 and Intel wireless 2200bg. According to **Fig. 3-(b)**, we can find that sub-measurements of different transceivers have many discrepancies. The mutual information and Spearman correlation coefficient of sub-measurements are 2.6439 and 0.4373, respectively. The mutual information of sub-measurements is larger than the entire measurements, which is caused by the number of samples. Actually, in this case, Spearman correlation coefficient can better reflect the reciprocity of sub-measurements. Consequently, many mismatches of bits might be produced if these measurements are separately quantified by the involved transceivers.

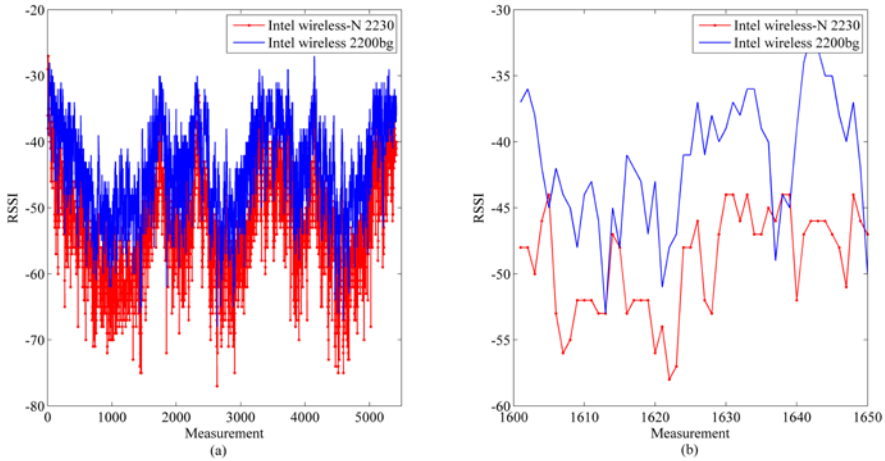

Fig. 3. The entire measurements and sub-measurements in the experiment (g)

5. The proposed scheme

As mentioned above, measurements of different transceivers have many discrepancies. Therefore, many mismatches of bits might be produced if these measurements are directly quantified into bits. In this work, an efficient key generation scheme leveraging wireless channel reciprocity and discrete cosine transform is proposed. The proposed scheme is expected to be efficient and available even facing the aforementioned challenge. The proposed scheme consists of five components: channel probing, pre-processing, multi-bits quantization & encoding, information reconciliation and privacy amplification.

Channel probing is implemented to collect sufficient channel measurements. Actually, the proposed scheme can generate key from any statistic. Therefore, experiments in Section 4 are just illustrated for validating the proposed scheme. During pre-processing stage, both transceivers separately apply DCT and IDCT to their measurements. Then, the outputs of IDCT instead of measurements are used as the inputs of the following process. Multi-bits quantization and gray code are used to generate the bit sequence. To extract the shared key, distillation [8] and Cascade [7] are implemented during information reconciliation. Finally, 2-universal hash functions are applied to achieve privacy amplification so that the highly random and uniformly distributed key can be generated. **Fig. 4** illustrates the implementation of the proposed key generation scheme.

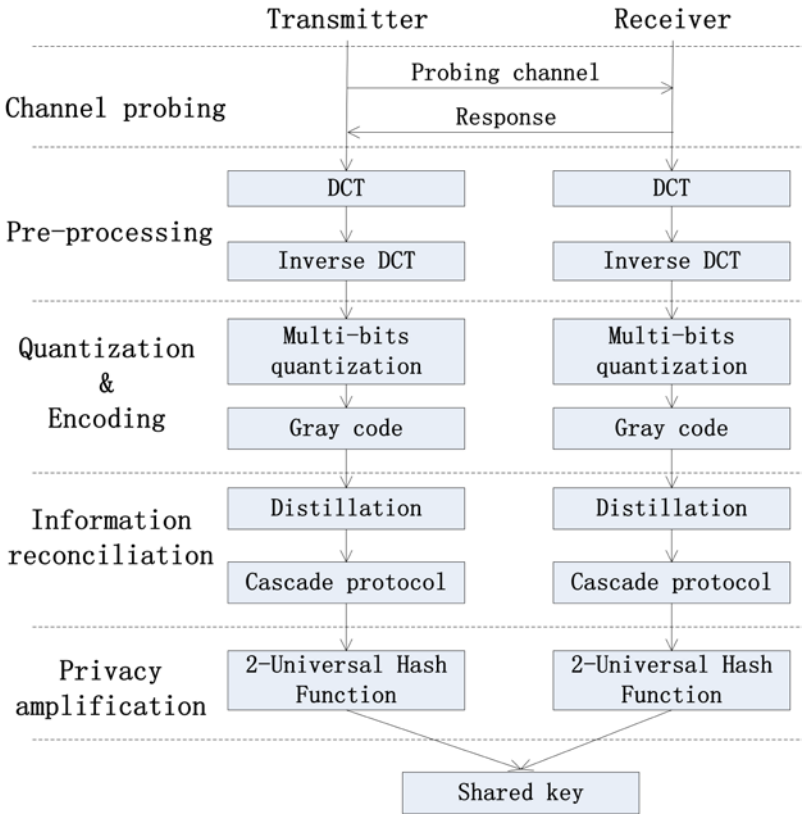

Fig. 4. Implementation of the proposed key generation scheme

5.1 Pre-processing

Discrete cosine transform and Karhunen-Loève transform (KLT) are two popular tools for data compression. In [16], KLT is used to convert measurements into uncorrelated components. However, the complexity of KLT significantly affects the efficiency of key generation. In contrast, DCT is a solution that can approach the compaction efficiency of KLT in many cases. Moreover, the computation complexity of DCT is greatly smaller than KLT. Therefore, in this work, DCT instead of KLT is used to pre-process measurements. On the one hand, it can convert measurements into uncorrelated components; on the other hand, the outputs of IDCT possess better reciprocity than measurements themselves, which is beneficial to reduce the bit mismatch rate.

According to the definition, DCT of the input sequence x can be calculated as

$$y(k) = a(k)\sum_{n=1}^{N} x(n)\cos\left(\frac{(2n-1)(k-1)\pi}{N}\right), k = 1, 2, ..., N \tag{10}$$

Where N denotes the number of coefficients, and

$$a(k)=\begin{cases} 1/\sqrt{N}, k=1 \\ \sqrt{2/N}, 2\le k\le N \end{cases} \tag{11}$$

The IDCT process corresponding to the aforementioned DCT is

$$x(n)=a(n)\sum_{k=1}^{N} y(k)\cos\left(\frac{(2n-1)(k-1)\pi}{N}\right), n=1,2,...,N \tag{12}$$

Where

$$a(n)=\begin{cases} 1/\sqrt{N}, n=1 \\ \sqrt{2/N}, 2\le n\le N \end{cases} \tag{13}$$

Based on the energy compaction property of DCT, it is possible to reconstruct the input sequence x from only a fraction of its DCT coefficients. That is, when parts of DCT coefficients are used to implement the IDCT procedure, the outputs of IDCT are actual the approximation of the input sequence, which means that significant patterns of the input sequence are reserved and some details are removed. According to the illustration in Section 4, measurements of different transceivers are highly correlated with each other accompanying with many discrepancies. Then, if both transceivers separately apply DCT and IDCT processes to reconstruct their measurements with fewer coefficients, the primary patterns of measurements are reserved and some small-scale fluctuations are removed. Comparing with measurements, the outputs of IDCT processes performed by different transceivers have better reciprocity and the bit mismatch rate after quantization turns out to be smaller.

During pre-processing phase, the involved transceivers have to separately use the same DCT and IDCT procedures to handle their measurements. Accordingly, the transmitter first determines the number of coefficients according to the energy compaction rate and smoothness of measurements. Then, this number is sent to the receiver. Finally, the receiver generate its outputs according to the received number. For both transceivers, the outputs of IDCT are used as the shared secrets for generating key.

We use the measurements illustrated in **Fig. 3-(a)** to analyze the effects of pre-processing. **Fig. 5-(a)** shows the first 1500 DCT coefficients of two transceivers. It can be found that most of the energy is allocated to the first few coefficients. According to **Fig. 5-(b),** more than 99% energy is allocated to the first 64 coefficients. Therefore, the outputs can be regarded as an excellent approximation of the input sequence while more than 64 coefficients are used to implement the IDCT process.

Furthermore, the results of IDCT performed by different number of coefficients are depicted in **Fig. 6**. Obviously, the outputs of all cases can be applied as good approximations of the input sequence. Moreover, the outputs contain more details if more coefficients are used to implement the IDCT process. As a result, the reciprocity of outputs become worse when more coefficients are used to reconstruct the input sequences.

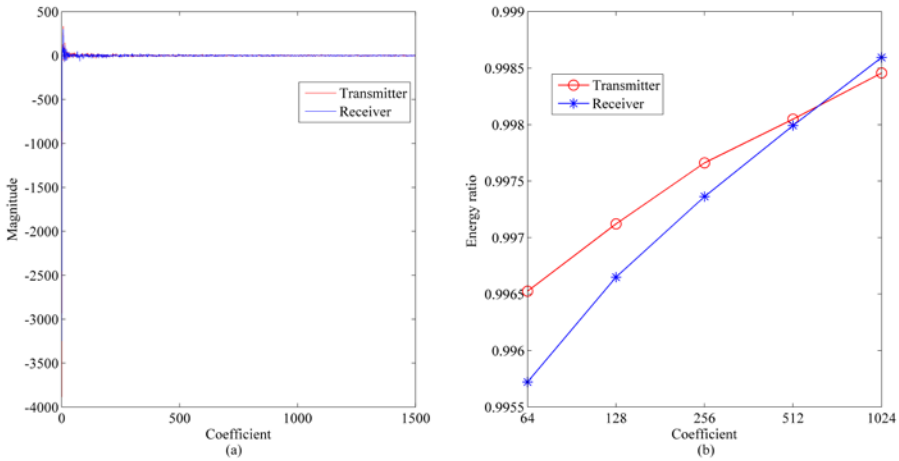

Fig. 5. DCT coefficients and energy compaction rates of the measurements in the experiments (g)

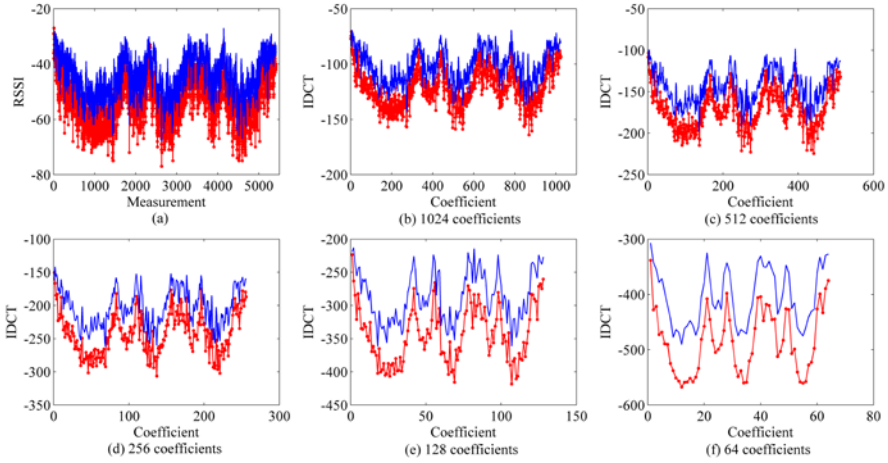

Fig. 6. Outputs of the IDCT processes performed by different number of coefficients

In **Fig. 7**, the Rényi mutual information of order 2 and Spearman correlation coefficients of the outputs in different cases are presented. The results also demonstrate that the reciprocity is better when fewer coefficients are used to implement the IDCT process, since more small-scale fluctuations are removed. Moreover, we can find that the reciprocities between the outputs in all illustrated cases are better than measurements illustrated in **Fig. 3-(a)**. Therefore, the pre-processing based on DCT and IDCT is beneficial to reduce the bit mismatch rate.

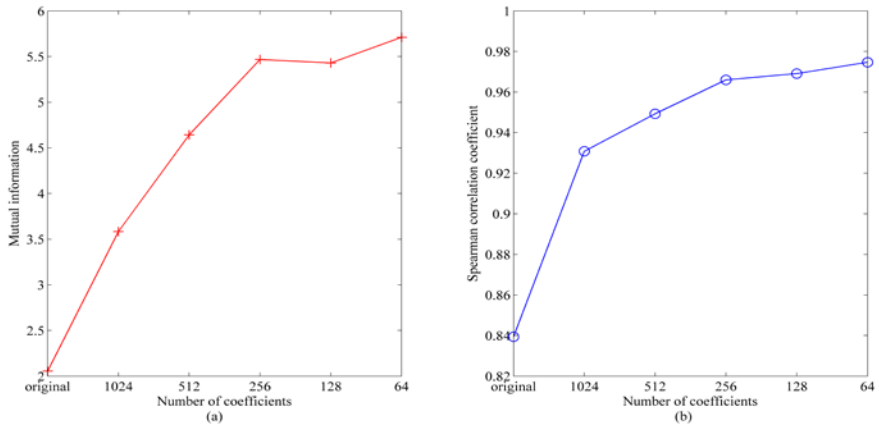

Fig. 7. Mutual information and Spearman correlation coefficients of the outputs generated in different cases

5.2 Quantization & Encoding

After the implementation of pre-processing, both transceivers have to separately convert their outputs of IDCT into bits. In this work, to enhance the key generation rate, uniform multi-bits quantization in conjunction with gray code is used to generate the bit sequence.

5.2.1 Uniform multi-bits quantization

For uniform multi-bits quantization, all quantization intervals are of the same width. That is, thresholds and quantization bins are evenly spaced. To agree on the quantization processes of different transceivers, the transmitter first determines the quantization level and sends it to the receiver. Then, both transceivers can implement the quantization process as follows:

1) According to the quantization level $level$, thresholds can be calculated as

$$threshold_i = \min + i \times \frac{\max - \min}{level}, 1 \le i \le level - 1 \tag{14}$$

With this formula, the i_{th} bin is determined to be $\{threshold_{i-1}, threshold_i\}$. The first bin and last bin are $\{\min, threshold_1\}$ and $\{threshold_{level-1}, \max\}$;

2) Find all inputs falling into each bin, and then they are marked as the symbol corresponding to this bin.

After the implementation of quantization, each transceiver can generate a symbol sequence. In **Fig. 8**, the 8-level uniform quantization for the input sequence is illustrated. In this case, the input sequence is the outputs of IDCT of transmitter and 64 coefficients are used to implement the IDCT process. Moreover, $threshold_j \left(1 \le j \le 7\right)$ denotes the j_{th} threshold in the 8-level quantization. Comparing with [11], all inputs are used to generate secret bits and thus the key generation rate is enhanced.

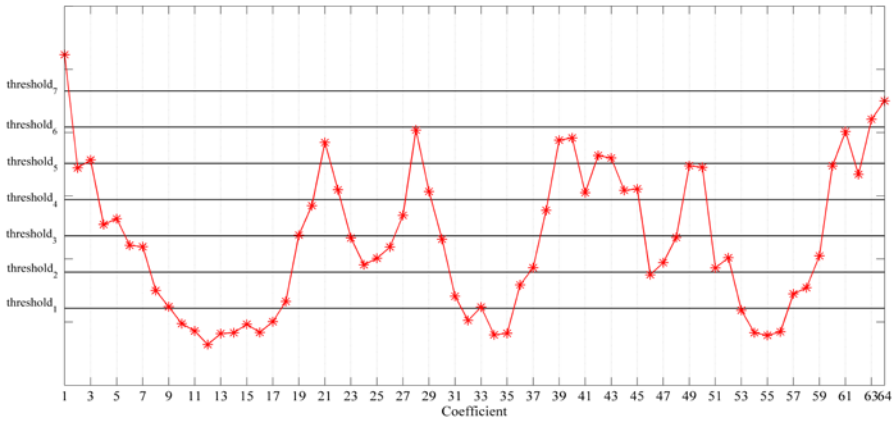

Fig. 8. Uniform 8-level quantization

5.2.2 Gray code

Gray code is a code assigning to a word of symbols so that adjacent code words have a single digit differing by 1. Accordingly, the resulting bit mismatch rate might be extremely low if the differences of majority unmatched symbols are 1. As a result, a low bit mismatch rate and high key generation rate can be achieved. In this work, the relationship between the quantization level and the length of gray code is determined as

$$length = \lceil \log_2 (level) \rceil \tag{15}$$

Table 3 shows the quantization bins of 8-level quantization and the corresponding gray codes. According to **Table 3**, each transceiver can convert its symbol sequence into a bit sequence.

Table 3. 8-level quantization bins & gray codes

Quantization bin	Symbol	Gray code
$\{\min, threshold_1\}$	1	000
$\{threshold_1, threshold_2\}$	2	001
$\{threshold_2, threshold_3\}$	3	011
$\{threshold_3, threshold_4\}$	4	010
$\{threshold_4, threshold_5\}$	5	110
$\{threshold_5, threshold_6\}$	6	111
$\{threshold_6, threshold_7\}$	7	101
$\{threshold_7, \max\}$	8	100

5.3 Information reconciliation

The involved transceivers apply uniform multi-bits quantization and gray code to generate their bit sequences. However, these bit sequences are not identical. Therefore, information reconciliation is required to correct bit errors and generate the shared key. Cascade protocol [7]

is a popular solution for achieve information reconciliation. According to [7], for the sufficiently reliable secret channels which with probability of any symbol being transmitted in-correctly as large as 15%, the efficient protocol that leaks an amount of information acceptably close to the minimum can be achieved. In order to find the error bits and correct them, the entire bit sequence is divided into several blocks and parities of each pair of blocks are used to check if they are match. The errors of each block can be corrected with the BINARY strategy. After several passes of interactions, the shared bit sequence between transceivers can be generated.

According to [7], let E_i denotes the expected number of errors in K_v^1 after completion of pass i , where K_v^1 is the v_{th} block in pass 1. If the following requirements are satisfied

$$\begin{cases} k_i = 2k_{i-1}, i > 1 \\ \sum_{l=j+1}^{\lfloor k_1/2 \rfloor} \delta_1(l) \le \dfrac{\delta_1(j)}{4} \\ E_1 \le -\dfrac{\ln\left(1/2\right)}{2} \end{cases} \tag{16}$$

Where k_i is the size of block in pass i and $\delta_1(j)$ denotes the decoding error probability of j_{th} block after pass 1. Then, we can get

$$\begin{cases} E_1 = 2\sum_{j=1}^{\lfloor k_1/2 \rfloor} j\delta_1(j) = k_1 p - \dfrac{\left(1-(1-2p)^{k_1}\right)}{2} \\ E_i \le \dfrac{E_{i-1}}{2}, i \ne 1 \end{cases} \tag{17}$$

Moreover, the amount of information $I(\omega)$ per block of length k_1 (per block K_v^1) leaked after ω passes can be bounded as

$$I(\omega) \le 2 + \frac{1-(1-2p)^{k_1}}{2}\lceil \log k_1 \rceil + 2\sum_{l=2}^{\omega}\sum_{j=1}^{\lfloor k_1/2 \rfloor} \frac{j\delta_1(j)}{2^{l-1}}\lceil \log k_1 \rceil \tag{18}$$

The relationship among the bit mismatch rate, passes of Cascade and amount of the leaked information is illustrated by the above formulas. The amount of the leaked information is an important factor which must be considered during privacy amplification.

In this work, distillation [8] is implemented before Cascade so that the bit mismatch rate can be reduced to satisfy the requirement of Cascade. During each pass of distillation, a pre-defined symbol is used to replace the abrupt transitions. Accordingly, many abrupt transitions can be eliminated after several passes. However, the increase of the pre-defined symbol might reduce the randomness of symbol sequence. Therefore, distillation in conjunction with Cascade is implemented to generate the shared bit sequence so that the efficiency and randomness can be guaranteed. The passes of distillation and Cascade are determined by the transmitter. In our experiments, at most three iterations of distillation and

four passes of Cascade can correct all errors. As a result, the shared key can be generated after information reconciliation.

5.4 Privacy amplification

Privacy amplification is implemented to guarantee the randomness of the shared key. As a classic solution, many schemes used for privacy amplification are achieved based on leftover hash lemma. According to leftover hash lemma, if the family H of functions $h:\{0,1\}^n \to \{0,1\}^l$ is pairwise independent, where $l = H_\infty(x) - 2\log(1/\varepsilon) - O(1)$, then a $(k, \varepsilon/2)$-extractor can be established. In this case, $H_\infty(x)$ denotes the min-entropy of the source x and ε refers to the statistical distance.

In [12], a method based on 2-universal hash function is proposed to achieve privacy amplification. In this work, this method is also applied to generate the shared secret key. We use the 2-universal hash family consists of all the functions $h:\{1,...,M\} \to \{0,1\}^m$ of the form

$$g_{a,b}(x) = (ax+b) \bmod p_M \tag{19}$$

$$h_{a,b}(x) = g_{a,b}(x) \bmod m \tag{20}$$

for every $a \in \{1,..., p_M - 1\}$ and $b \in \{0,..., p_M - 1\}$. M is determined by the expected length of secret key. The integer p_M is a prime number larger than M. To achieve privacy amplification, both transceivers first divide their bit sequences into smaller blocks. Then, the transmitter determines the values of all parameters and sends them to the receiver. Finally, the transmitter and receiver can implement the same privacy amplification process to generate the shared secret key.

6. Performance evaluation

In this work, several metrics are used to evaluate the performance of the proposed scheme. We briefly describe these metrics as follows:

Symbol difference: symbol difference characterizes the absolute difference of symbols which are generated by quantizers of different transceivers. The symbol difference equaling to 1 is termed as significant symbol difference since it significantly affects the bit mismatch rate when gray code is used to generate the bit sequence.

Bit mismatch rate: bit mismatch rate is the ratio of the number of mismatches between two bit sequences to the size of these bit sequences. In this case, two bit sequences are of same length and are not corrected by information reconciliation. Bit mismatch rate affects the efficiency and security of key generation.

Key generation rate: key generation rate is defined as the average number of secret bits extracted per collected measurement.

Randomness: randomness represents the uncertainty associated with the generated bit sequence. Randomness determines whether a key can be used for crypto applications.

Computation complexity: we mainly analyze the computation complexity of pre-processing, since other components are similar to the existing schemes. Specifically, we analyze the computation complexities of DCT and IDCT.

6.1 Symbol difference

In this work, the involved transceivers separately apply uniform multi-bits quantization to generate their symbol sequences. Due to the discrepancies between the inputs of quantizers, the generated symbol sequences have many mismatches. Consequently, they might be converted to mismatches between two bit sequences after encoding. According to the characteristics of gray code, the proportion of significant symbol difference greatly affects the bit mismatch rate. In **Fig. 9-(a)** and **Fig. 9-(b)**, symbol differences between two symbol sequences generated in different cases are illustrated. These symbol sequences are extracted from the measurements illustrated in **Fig. 3**.

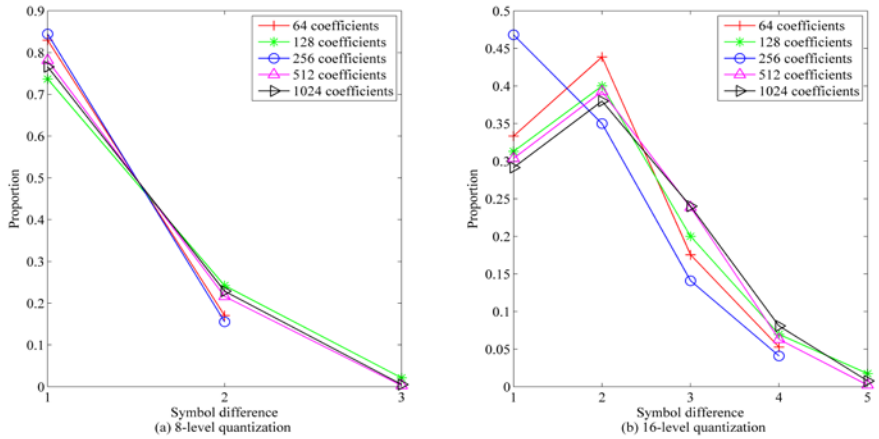

Fig. 9. Symbol differences in different cases

According to **Fig. 9-(a)**, the proportions of significant symbol differences are larger than 0.7 in all 8-level quantization scenarios. The maximum value of these symbol differences is 3. Furthermore, **Fig. 9-(b)** demonstrates that most symbol differences are 1, 2 and 3 when 16-level quantization is used. The proportions of such symbol differences range from 91.16% to 95.91% in all cases. The results of **Fig. 9-(a)** and **Fig. 9-(b)** also show that the distribution of symbol differences is not affected by the number of DCT coefficients used in pre-processing, i.e., variations of symbol differences are similar when the quantization level is determined. In addition, all symbol differences are equal to 1 when 4-level quantization is applied and the bit mismatch rate is greatly lower than other cases.

6.2 Bit mismatch rate

Bit mismatch rate is one of the critical metrics for key generation. Due to the impact on information reconciliation, the efficiency of key generation is significantly affected by the bit mismatch rate. **Fig. 10** draws the variations of bit mismatch rates in different cases. We can find that the illustrated bit mismatch rates cannot meet the requirement of efficient Cascade. Therefore, the involved transceivers have to apply distillation to reduce some mismatches between these sequences. In our experiments, at most three iterations of distillation can ensure that the bit mismatch rate is lower than 15%. For example, when the measurements illustrated in **Fig. 3-(a)** are processed by IDCT of 256 coefficients and 8-level quantization as well as 3-bits gray code is applied to generate bit sequences, the resulting bit mismatch rate is 24.25%. However, the bit mismatch rate can be reduced to 8.32% after one iteration of distillation.

Consequently, Cascade can be used to efficiently correct the bit errors between two bit sequences and the shared key can be generated.

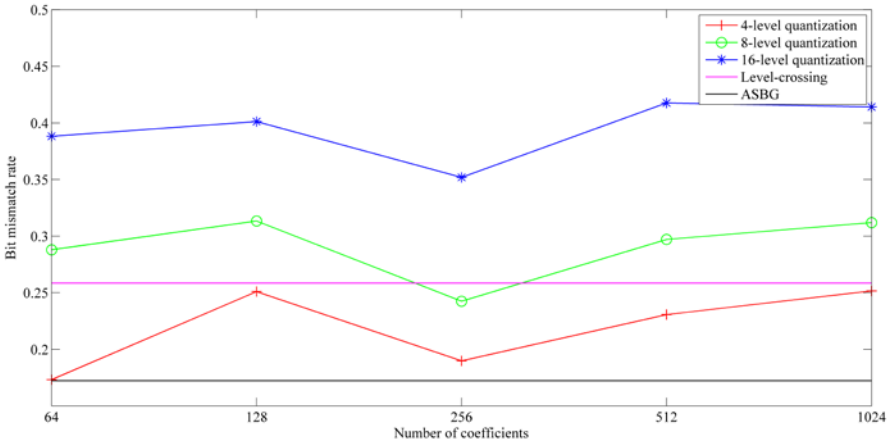

Fig. 10. Bit mismatch rates in different cases

In **Fig. 10**, we also compare the bit mismatch rate of the proposed scheme with two other classic schemes: ASBG [8] and level-crossing [11]. In this case, the value of α used in these schemes is set to 0.8. The size of excursion used in level-crossing is 2 and each block in ASBG has 50 measurements. The results show that when 4-level quantization is used during the proposed scheme, bit mismatch rate of the proposed scheme is lower than the level-crossing scheme but higher than the ASBG scheme. However, comparing with the level-crossing and ASBG schemes, although the bit mismatch rate of the proposed scheme is not the lowest, the key generation rate of the proposed scheme is the highest since all measurements are used to generate the shared key.

6.3 Key generation rate

Key generation rate is another critical metric which intuitively reflects the rate of establishing key. In this work, key generation rate can be figured out as the number of bits generated by privacy amplification to the number of measurements collected during channel probing. During the implementation of the proposed scheme, several factors might affect the key generation rate, such as the number of coefficients used for measurements reconstruction, the length of gray code, iteration of distillation, the amount of leaked information during information reconciliation and the entropy of input for privacy amplification. **Fig. 11** shows key generation rates in different cases where different number of coefficients and quantization levels are used. We can find that the key generation rate is increased when more coefficients are used during pre-processing. Moreover, the key generation rate is enhanced by the quantization level. In addition, we can find that the key generation rate of the proposed scheme is higher than the level-crossing and ASBG schemes when more than 256 coefficients are used during pre-processing phase.

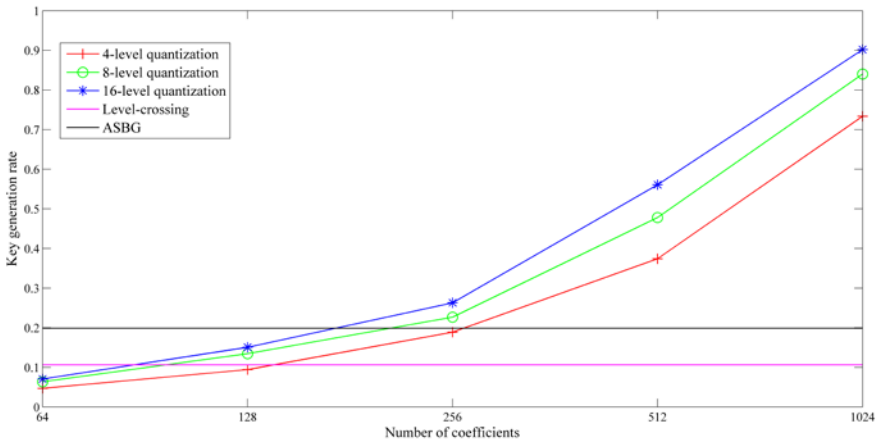

Fig. 11. Key generation rates in different cases

6.4 Randomness

NIST randomness test suite [9] is a classic tool for evaluating randomness of a bit sequence. There are totally 15 different statistical tests. Moreover, the cumulative sums test includes two sub tests: the forward test and the reverse test. In this work, we select 9 tests according to the input size recommendation. The P-value of these tests, which can be considered as the evaluation threshold, is set to 0.01. **Table 4** shows the results of tests for keys in different cases. In this table, $\langle m,n \rangle$ denotes the key generation scheme that applies m coefficients for pre-processing and n level quantization. In addition, the measurements used for evaluation is the measurements illustrated in **Fig. 3-(a)**. We can find that the keys generated in all cases pass the randomness tests. Consequently, they can be used to protect the communication between the involved transceivers.

Table 4. The results of randomness evaluation

	$\langle 64,4 \rangle$	$\langle 64,8 \rangle$	$\langle 64,16 \rangle$	$\langle 256,4 \rangle$	$\langle 256,8 \rangle$	$\langle 256,16 \rangle$
Approx.Entropy	0.6875	0.5874	0.1925	0.9435	0.9593	0.5162
BlockFrequency	0.4827	0.7716	0.6449	0.4247	0.8986	0.9150
Cum.Sums(FWD)	0.6512	0.8474	0.0497	0.8915	0.6582	0.8583
Cum.Sums(REV)	0.5711	0.7082	0.0183	0.3404	0.8438	0.9971
FFT	0.2806	0.3630	0.8551	0.9111	0.5411	0.5358
Frequency	0.9292	0.8787	0.0248	0.5069	0.4675	0.7078
Longest Run	0.9292	0.6768	0.5289	0.7109	0.3337	0.5192
Runs	0.7906	0.3610	0.8836	0.5191	0.8884	0.0241
Serial	0.6924 /0.340	0.6015 /0.3899	0.1512 /0.4880	0.9846 /0.9070	0.4861 /0.1144	0.3776 /0.2928

6.5 Computation complexity

During the proposed scheme, DCT and IDCT are implemented to pre-process the collected measurements. In general, the computation complexity of DCT is $O\left(n\log\left(n\right)\right)$ where n is

the number of measurements. In contrast, the computation complexity of KLT is $O\left(n^3\right)$. Therefore, DCT instead of KLT is used to pre-process the measurements during the proposed scheme.

In this paper, performance evaluation of the proposed scheme is achieved based on the measurements illustrated in **Fig. 3-(a)**. Actually, for the measurements collected in all experiments in mobile scenarios, the implementation of pre-processing can successfully reduce the bit mismatch rate. Consequently, the proposed scheme can use these measurements to efficiently generate shared secret keys between the involved transceivers.

Conclusion

In this paper, an efficient key generation scheme leveraging wireless channel reciprocity and discrete cosine transform is proposed. After collecting sufficient channel measurements, discrete cosine transform and inverse discrete cosine transform are used to pre-process these measurements. Then, the outputs of IDCT are used as the inputs of uniform multi-bits quantization such that the key generation rate can be enhanced. Gray code is used to generate a bit sequence for each transceiver. After the implementations of information reconciliation and privacy amplification, the shard secret key can be generated between the involved transceivers. To validate the performance of the proposed scheme, several experiments in real environments are conducted. According to the results of these experiments, the pre-processing of measurements is available to reduce the bit mismatch rate. Consequently, the proposed scheme is able to efficiently generate shared secret keys between transceivers. The results of randomness evaluation demonstrate that the generated key can be used to protect the communication between the involved transceivers. In future, more experiments will be conducted to comprehensively validate the proposed scheme.

References

[1] J. Zhang, T. Q. Duong, A. Marshall and R. Woods, "Key Generation From Wireless Channels: A Review," *IEEE Access*, vol. 4, pp. 614-626, 2016. Article (CrossRef Link)

[2] Rappaport, Theodore S. *Wireless communications: principles and practice*, 2nd Edition, Prentice Hall PTR, New Jersey, 2002. Article (CrossRef Link)

[3] Brassard, Gilles, and Louis Salvail, "Secret-key reconciliation by public discussion," *Theory and Application of Cryptographic Techniques*, pp. 410-423, 1993. Article (CrossRef Link)

[4] Liu, Yanpei, Stark C. Draper, and Akbar M. Sayeed, "Exploiting channel diversity in secret key generation from multipath fading randomness," *IEEE Transactions on information forensics and security*, vol. 7, no. 5, pp. 1484-1497, 2012. Article (CrossRef Link)

[5] S. Mathur, R. Miller, A. Varshavsky, W. Trappe and N. Mandayam, "Proximate: proximity-based secure pairing using ambient wireless signals," in *Proc. of the 9th international conference on Mobile systems, applications, and services*, pp. 211-224, 2011. Article (CrossRef Link)

[6] Q. Wang, H. Su, K. Ren and K. Kim, "Fast and scalable secret key generation exploiting channel phase randomness in wireless networks," in *Proc. of IEEE INFOCOM*, vol. 8, no. 1, pp. 1422-1430, 2011. Article (CrossRef Link)

[7] Zhang, Junxing, S. K. Kasera, and N. Patwari, "Mobility Assisted Secret Key Generation Using Wireless Link Signatures," in *Proc. of IEEE INFOCOM*, pp. 1-5, 2010. Article (CrossRef Link)

[8] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "Secret key extraction from wireless signal strength in real environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 917–930, 2013. Article (CrossRef Link)

[9] Lawrence E. Bassham, III , Andrew L. Rukhin , Juan Soto , James R. Nechvatal , Miles E. Smid , Elaine B. Barker , Stefan D. Leigh , Mark Levenson , Mark Vangel , David L. Banks , Nathanael Alan Heckert , James F. Dray , San Vo, "SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *National Institute of Standards & Technology*, Gaithersburg, MD, 2010. Article (CrossRef Link)

[10] B. Azimi-Sadjadi, A. Kiayias, A. Mercado and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *Proc. of ACM Conference on Computer and Communications Security, CCS 2007*, Alexandria, USA, pp. 401-410, Oct. 2007. Article (CrossRef Link)

[11] S. Mathur, W. Trappe, N. Mandayam, C. Ye and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *Proc. of International Conference on Mobile Computing and Networking, MOBICOM 2008*, pp.128-139, 2008. Article (CrossRef Link)

[12] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proc. of the 15th annual international conference on Mobile computing and networking*, pp. 321–332, 2009. Article (CrossRef Link)

[13] M. Wilhelm, I. Martinovic and J. B. Schmitt, "Secure key generation in sensor networks based on frequency-selective channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 339-343, 2013. Article (CrossRef Link)

[14] H. Liu, Y. Wang, Y. Chen, C.Koksal and J. Yang, "Group secret key generation via received signal strength: protocols, achievable rates, and implementation," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2820-2835, 2014. Article (CrossRef Link)

[15] J. Croft, N. Patwari and S. K. Kasera, "Robust uncorrelated bit extraction methodologies for wireless sensors," in *Proc. of ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp.70-81, 2010. Article (CrossRef Link)

[16] N. Patwari, J. Croft, S. Jana and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Transactions on Mobile Computing*, vol. 9, no. 9, pp. 17-30, 2009. Article (CrossRef Link)

[17] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *Proc. of IEEE INFOCOM*, IEEE, pp. 927–935, 2012. Article (CrossRef Link)

[18] A. A. Hassana, W. E. Starkb, J. E. Hersheyc and S. Chennakeshua, "Cryptographic key agreement for mobile radio," *Digital Signal Processing*, vol. 6, no. 4, pp. 207-212, 1996. Article (CrossRef Link)

[19] H. Koorapaty, A. Hassan, and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Communication Letters*, vol. 4, no. 2, Feb 2000. Article (CrossRef Link)

[20] M. F. Haroun and T. Aaron Gulliver, "Secret key generation using chaotic signals over frequency selective fading channels," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1764 – 1775, 2015. Article (CrossRef Link)

[21] A. Badawy, T. Khattab, T. El-Fouly, A. Mohamed, D. Trinchero, and C.-F. Chiasserini, "Secret key generation based on aoa estimation for low snr conditions," in *Proc. of IEEE Vehicular Technology Conference*, 2015. Article (CrossRef Link)

[22] R. Wilson, D. Tse, R. Scholtz et al., "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, 2007. Article (CrossRef Link)

[23] Jingjing Huang and Ting Jiang, "Secret Key Generation from Common Randomness over Ultra-wideband Wireless Channels," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 10, pp. 3557-3571, 2014. Article (CrossRef Link)

[24] Ning Gao, Xiaojun Jing, Songlin Sun, Junsheng Mu and Xiang Lu, "A New Fuzzy Key Generation Method Based on PHY-Layer Fingerprints in Mobile Cognitive Radio Networks," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 7, pp. 3414-3434, 2016. Article (CrossRef Link)

[25] J. W. Wallace and K. S. Rajesh, "Automatic Secret Keys From Reciprocal MIMO Wireless Channels: Measurement and Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 5, no.3, pp. 381-392, 2010. Article (CrossRef Link)
[26] T. Shimizu, H. Iwai, and H. Sasaoka, "Physical-layer secret key agreement in two-way wireless relaying systems," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3 PART 1, pp. 650 – 660, 2011. Article (CrossRef Link)
[27] S. Primak, K. Liu, and X. Wang, "Secret key generation using physical channels with imperfect csi," in *Proc. of IEEE 80th Vehicular Technology Conference (VTC Fall)*, pp. 1-5, 2014. Article (CrossRef Link)
[28] X. Wu, Y. Peng, C. Hu, H. Zhao, and L. Shu, "A secret key generation method based on csi in ofdm-fdd system," in *Proc. of 2013 IEEE Globecom Workshops*, pp. 1297-1302, 2013. Article (CrossRef Link)

Furui Zhan is a Ph. D. candidate of Dalian University of Technology. He received the B. S. degree from Central South University, Changsha, China and the M. E. degree from Harbin Engineering University, Harbin, China. His main research interests include wireless sensor network and wireless network security.

Nianmin Yao is now a professor in the School of Computer Science and Technology at Dalian University of Technology, Dalian, China. He received the B. S., M. E. and Ph. D. degrees from Jilin University, Changchun, China. He has been a visiting scholar at University of Connecticut. His main research interests include network security, wireless sensor network etc.