Key Generation from Wireless Channels: A Review

Junqing Zhang, Trung Q. Duong, Senior Member, IEEE, Alan Marshall, Senior Member, IEEE, and Roger Woods, Senior Member, IEEE

Abstract—Key generation from the randomness of wireless channels is a promising alternative to public key cryptography for the establishment of cryptographic keys between any two users. This paper reviews the current techniques for wireless key generation. The principles, performance metrics and key generation procedure are comprehensively surveyed. Methods for optimizing the performance of key generation are also discussed. Key generation applications in various environments are then introduced along with the challenges of applying the approach in each scenario. The paper concludes with some suggestions for future studies.

Index Terms—Physical layer security, key generation, wireless communication

I. INTRODUCTION

A. Wireless Network Security

The inherent broadcast nature of wireless communication allows transmissions to be received by any user within the range, resulting in attackers’ ability to initiate various passive attacks such as eavesdropping, traffic analysis and monitoring, etc., or to execute active attacks like jamming, spoofing, modification, replaying and denial-of-service (DoS) attack, etc. [1].

There has been extensive research interest to protect wireless transmission [2]. Traditionally, the data is secured by classic encryption schemes [3], [4], which work on the assumption that the algorithm is complex enough so that the time taken by eavesdroppers to crack the cryptographic system is much longer than the validity of the information itself, therefore, the backward secrecy is guaranteed. As shown in Fig. 1, classic encryption schemes consist of symmetric encryption schemes and asymmetric encryption schemes, depending on the keys that the two cryptographic parties use. Symmetric encryption schemes use the same key and are usually employed for data protection thanks to their efficiency in data encryption. Asymmetric encryption schemes, also known as public key cryptography, use the same public key but different private keys and are usually applied for key distribution. An illustration of a classic encryption system is shown in Fig. 2a, where Alice and Bob represent two legitimate users who want to share information securely between each other.

This work was supported by the Queen’s University Belfast scholarship, Newton Institutional Links Grant 172719890, Royal Academy of Engineering Research Fellowship under Grant RF141514/22, and US-Ireland R&D Partnership USI033 ‘WiPhyLoc8’ grant involving Rice University (USA), University College Dublin (Ireland) and Queen’s University Belfast (Northern Ireland).

J. Zhang, T. Q. Duong and R. Woods are with ECIT, Queen’s University Belfast, Belfast, BT3 9DT, UK. (email: {jzhang20, trung.q.duong, r.woods}@qub.ac.uk)

A. Marshall is with Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, UK. (email: Alan.Marshall@liverpool.ac.uk)

Fig. 1. Research streams in wireless network security

Fig. 2. Illustration of wireless network security systems

Classic encryption schemes are faced with several vulnerabilities. Take public key cryptography as an example. Firstly, it depends on the computational hardness of some mathematical problems, e.g., discrete logarithm. This computational security nature may not hold in future due to the rapid development of hardware technology. In addition, it requires a key management infrastructure which should be secured as well. This
approach is therefore less attractive for many wireless sensor networks (WSNs) and ad hoc networks applications, because sensor nodes have limited computational capacity while ad hoc networks are decentralized.

While classic encryption schemes are applied in the upper layers of the communication protocols, the physical layer can also be exploited to enhance wireless security. Physical layer security (PLS) schemes leverage unpredictable and random characteristics of wireless channels in order to achieve information theoretic-security [5]–[10]. As shown in Fig. 1, PLS schemes are composed of keyless security and secret key-based secrecy [8]. Pioneered by Wyner’s wiretap channel model [11], keyless security does not require keys for encryption but employs code design and channel properties of legitimate users and eavesdroppers to achieve secrecy (see [8] and references therein). However, the legitimate users usually require full/part of instantaneous/statistical channel state information (CSI) of the eavesdroppers, which is not always available in practice and results in a very complex implementation.

Secret key-based secrecy dated back as early as 1919 when the concept of one-time pad, also known as Vernam cipher [12], was used to encrypt each message bit with a random secret key bit. Later on, Shannon laid the theoretical basis for perfect secrecy [13]. The message $M$ is encoded into codeword $C$ which does not reveal any information about the message, i.e.,

$$H(M|C) = H(M),$$

where $H(\cdot)$ denotes the entropy. This requires the information of the key sequence should be larger than, or at least equal to, the information of the message. One possible way to establish the key is to generate keys from the wireless channels. However, in practice, it is very challenging, if not impossible, to efficiently establish random keys between legitimate users which cannot be reused. Alternatively, a hybrid cryptosystem can be constructed by combining key generation and symmetric encryption, as illustrated in Fig. 2b. The security level of the system is enhanced by replacing public key cryptography with key generation.

### B. Key Generation

In this paper, we review secure key generation from the randomness of wireless channels. Unlike the computationally secure nature of the public key cryptography, wireless key generation is information-theoretically secure, because it is based on the random characteristics of wireless channels [14], [15]. In addition, this technique is lightweight and does not require any aid from other users. A comparison of the above mentioned schemes is given in Table I.

Key generation was theoretically proposed/investigated in [14] and [15] in 1993. Key generation model is shown in Fig. 3, where Alice and Bob want to establish a secure cryptographic key and an eavesdropper Eve located $d$-cm away from Alice, listens to all the transmissions. Alice, Bob and Eve can get correlated observations $X^n = (X_1, \ldots, X_n)$, $Y^n = (Y_1, \ldots, Y_n)$ and $Z^n = (Z_1, \ldots, Z_n)$, respectively. Alice and Bob will exchange a message $s$ over the public channel, which may be heard by Eve as well. For any $\epsilon > 0$ and sufficiently large $n$, if there exists $K^A = g_A(X^n, s)$ and $K^B = g_B(Y^n, s)$ making the key generation system satisfy

$$\Pr(K^A \neq K^B) < \epsilon,$$

$$\frac{1}{n} I(K^A; s, Z^n) < \epsilon,$$

$$\frac{1}{n} H(K^A) > R - \epsilon,$$

$$\frac{1}{n} \log |K| < - \frac{1}{n} H(K^A) + \epsilon,$$

then $R$ in (4) is the achievable key rate, where $I(\cdot)$ denotes mutual information and $K$ is key’s alphabet. (2) means that Alice and Bob can generate the same key with a high probability; (3) indicates the message exchange via public discussion leaks no information to Eve, which guarantees the security of the generated key; (5) ensures the key is uniformly distributed, which is desirable for the cryptographic applications. The largest achievable key rate is defined as key capacity and given as

$$C_K = \min[I(X; Y), I(X; Y|Z)].$$

There has been extensive research effort to implement the above theory in practice and to approach to the theoretical limits. The first practical key generation protocol was proposed in 1995 [16] and since then has triggered research interest in wireless key generation. Chapter 4 in [7] reviewed the wireless key generation from the information theory perspective. The authors in [17] surveyed the key generation development merging channel probing and quantization as one step, which is shown later as two separate ones. We note that a recent study in [18] has introduced the challenges and opportunities of the key generation but it has not considered implementation details. Although in [19], key generation schemes have been summarized, e.g., received signal strength (RSS)-based and channel phase-based schemes, a thorough review of key generation techniques is still needed as these schemes and techniques have evolved fast since then. In this paper, we provide a literature review on techniques of key generation systems. We also highlight research areas of key generation.
that need more understanding and provide suggestions for future research.

The rest of the paper is organized as follows. Section II and III introduce the key generation principles and evaluation metrics, respectively. Section IV details the channel parameters that can be used for key generation, including CSI and RSS. The key generation procedure is explained in Section V and optimized in Section VI. Applications in various environments are then reviewed in Section VII. Section VIII concludes the paper with future research suggestions.

II. KEY GENERATION PRINCIPLES

Key generation is based on three principles, i.e., temporal variation, channel reciprocity, and spatial decorrelation.

Temporal variation is introduced by the movement of the transmitter, receiver or any objects in the environment, which will change the reflection, refraction and scattering of the channel paths. The randomness caused by such unpredictable movement can be used as the random source for key generation [20]–[26]. There is research effort to exploit the randomness in frequency domain [27]–[31] and spatial domain [32]–[36]. However, in a static environment where these features remain the same, the randomness is rather limited. Temporal variation is thus still required in order to introduce a sufficient level of randomness. It can be quantified by the autocorrelation function (ACF) of the signal, which is given as

$$R_X(t, \Delta t) = \frac{E\{(X(t) - \mu_X)(X(t + \Delta t) - \mu_X)\}}{\sigma_X^2},$$

(7)

where $E\{\cdot\}$ denotes the expectation operator, and $\mu_X$ and $\sigma_X$ represents the mean value and standard deviation of random variable $X(t)$, respectively.

Channel reciprocity implies that the multipath and fading at both ends of the same link, i.e., same carrier frequency, are identical which is the basis for Alice and Bob to generate the same key. The signals have to be measured by hardware platforms, which usually work in half duplex mode and introduce noise. Therefore, the received signals of the uplink and downlink path are asymmetric due to the non-simultaneous measurements and noise effects, which limits key generation applications within time-division duplexing (TDD) systems and slow fading channels. These effects can be mitigated using signal processing algorithms discussed in Section V-A. The signal similarity can be quantified by the cross-correlation between the measurements, which is given as

$$\rho_{XY} = \frac{E\{XY\} - E\{X\}E\{Y\}}{\sigma_X \sigma_Y}.$$  

(8)

Spatial decorrelation indicates that any eavesdropper located more than one half-wavelength away from either user experiences uncorrelated multipath fading, which can also be described by the cross-correlation between the signals of legitimate users and eavesdroppers. This property is essential for the security of key generation systems and has been claimed in most key generation papers. However, it may not be satisfied in all the environments. Channel variation is contributed by large-scale fading (i.e., path loss and shadowing) and small-scale fading [37]. In the Jake’s model with a uniform scattering Rayleigh environment and without a line-of-sight (LoS) path, if the number of scatters grows to infinity, the signal decorrelates over a distance of approximately one half-wavelength [37]. Some experiments have also shown this property [38]–[41]. However, when large-scale fading is dominant, special attention is required as the channel is more correlated [42]. There is research reporting that signals observed by eavesdroppers are correlated to signals of legitimate users [43]–[45], which makes key generation systems vulnerable and requires special consideration to combat eavesdropping. In general, spatial decorrelation has not been extensively studied and is worth more research input.

III. PERFORMANCE METRICS

Key generation is designed to establish cryptographic keys for encryption and/or authentication. These applications have special requirements on the key’s randomness, refresh rate, etc. Thus, key generation systems can be correspondingly evaluated in terms of three important metrics: randomness, key generation rate (KGR), and key disagreement rate (KDR).

A. Randomness

Randomness is the most important feature of key generation systems. Cryptographic applications have strict requirements on the randomness of the key sequence [4]. A statistical

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Description</th>
<th>Implementation</th>
<th>Complexity</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric encryption</td>
<td>Legitimate users use the same key to encrypt data.</td>
<td>Yes</td>
<td>Low</td>
<td>Efficiency in data encryption</td>
<td>Computationally secure; A secure key required prior.</td>
</tr>
<tr>
<td>Asymmetric encryption</td>
<td>Legitimate users use the same public key but different private keys to</td>
<td>Yes</td>
<td>High</td>
<td>Key distribution with different</td>
<td>Computationally secure; Public key infrastructure required; Not</td>
</tr>
<tr>
<td></td>
<td>distribute a session key.</td>
<td></td>
<td></td>
<td>private keys</td>
<td>applicable to low computational capacity devices.</td>
</tr>
<tr>
<td>Keyless security</td>
<td>Legitimate users securely communicate by leveraging code design and</td>
<td>Not reported</td>
<td>High</td>
<td>Information-theoretically secure;</td>
<td>Eavesdroppers’ CSI usually required.</td>
</tr>
<tr>
<td></td>
<td>channel properties.</td>
<td></td>
<td></td>
<td>Secret transmission without keys.</td>
<td></td>
</tr>
<tr>
<td>Key generation</td>
<td>Legitimate users generate key from the randomness of the common channel.</td>
<td>Yes</td>
<td>Low</td>
<td>Information-theoretically secure;</td>
<td>Limited by the channel dynamicity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lightweight; No aid from other users required.</td>
<td></td>
</tr>
</tbody>
</table>
randomness test suite provided by National Institute of Standards and Technology (NIST) [46] is widely used to test the randomness of random number generators (RNGs) and pseudo random number generators (PRNGs). In essence, a key generation system is a type of RNG, so NIST statistical test suite can also be applied.

As randomness is a probabilistic property, statistical analysis is employed to test a specific null hypothesis (H0), i.e., the sequence under test is random. A *P*-value is returned by each test, which summarizes the strength of the evidence against the null hypothesis. A significance level $\alpha$, typically in the range $[0.001, 0.01]$, is chosen. When $P$-value $\geq \alpha$, the sequence is accepted as random, otherwise, it is deemed to be non-random.

There are infinite statistical features of a random sequence, therefore, in practice, it is impossible to test all the features using a finite set of tests [46]. The NIST test suite has 15 tests to evaluate different randomness features, each for a specific feature of the randomness, e.g., the proportion of 1s and 0s (frequency test), periodic feature (DFT test), etc. Some tests require extremely long sequence. For example, the recommended input length is $10^6$ for the linear complexity, random excursions and random excursions variant tests and is judged to be very long in a key generation system. Thus, most of the key generation research has only adopted a subset of the randomness tests to assess a subset of the randomness features [20], [21], [23], [30], [47]–[50].

The readers are referred to [46] for a detailed description of all the randomness tests and advised to download the source code of the test suite to evaluate the randomness of their key generation systems.

**B. KGR**

KGR describes the amount of secret bits produced in one second/measurement. It mainly depends on environment conditions, which determines the amount of randomness available for extraction. A high KGR is essential for the real time key generation process as the cryptographic schemes require a certain length of keys. For example, advance encryption standard (AES) needs a key sequence with a minimum length of 128 bits.

**C. KDR**

KDR is the percentage of the different bits between the keys generated by Alice and Bob, which is defined as

$$KDR = \frac{\sum_{i=1}^{N} |K^A(i) - K^B(i)|}{N},$$

(9)

where $N$ is the length of keys. The KDR should be smaller than the correction capacity of information reconciliation techniques, otherwise, key generation fails, which is discussed in Section V-C.

**D. Summary**

There are also other assessment metrics such as scalability and implementation issues [19]. However, randomness, KGR and KDR are the most important and popular metrics which describe the success and efficiency of the system, which are therefore used for evaluation throughout this paper.

**IV. CHANNEL PARAMETERS**

Channel parameters are the most essential part of key generation, as it is the random source representing unpredictable channel characteristics. In this section, CSI and RSS are reviewed.

**A. CSI**

CSI is a fine-grained channel parameter which provides detailed channel information. CSI-based systems are able to provide a high KGR [51] and have been experimentally proved to be immune to predictable channel attacks [30]. In this paper, CSI mainly refers to channel impulse response (CIR) and channel frequency response (CFR).

1) **CIR**: A multipath channel can be modelled as several resolvable path components and its CIR $h(\tau, t)$ can be given as

$$h(\tau, t) = \sum_{l=0}^{L(t)} \alpha_l(t) e^{-j\phi_l(t)} \delta(\tau - \tau_l(t)),$$

(10)

where $\alpha_l(t)$, $\phi_l(t)$ and $\tau_l(t)$ are the amplitude attenuation, phase shift and time delay of the $l^{th}$ tap, respectively, $L(t)$ is the total path number and $\delta(\cdot)$ is the Dirac function.

CIR has been proved to be ideal for key generation [51]. It has both amplitude and phase information. In wideband systems, the phase shift $\phi_l(t)$ can be estimated and used for key generation [52]–[55]. It can also be used in narrowband systems [29], [47], [48], but the phase in this case is decreased into a single-dimension parameter which loses lots of channel information. Phases can be accumulated to each other and this special feature leads to interesting applications such as group and cooperative key generation [47], [48]. In addition, phases of all the paths are distributed uniformly on $[0, 2\pi]$, which are not affected by the path power. There is only one practical phase-based key generation system implemented in a narrowband system [29] and no practical wideband-based systems have been reported yet. This is because phase is vulnerable to noise, carrier frequency offset and asynchronous clocks/clock drift at the receiver, etc.

Another aspect is amplitude of CIR. In an ultra wideband (UWB) system, the amplitude can be estimated by sending a pulse signal [39], [40], [56]–[59]. The UWB-based measurement systems are usually constructed by oscilloscope, waveform generator, etc. In a narrowband system when the transmission power is fixed, the amplitude of CIR is equivalent to the received power [29]. The power of CIR decreases with delay, e.g., it follows exponential distribution in an indoor environment, resulting in a high KDR for the paths with small power as they are vulnerable to the noise. This may be tackled by using the peak CIR only [20] which sacrifices the KGR, or using an adaptive quantization algorithm [58].

2) **CFR**: CFR provides channel effect in frequency domain and can be given as

$$H(f, t) = \int_{0}^{\tau_{\text{max}}} h(\tau, t)e^{-j2\pi f \tau}d\tau,$$

(11)

where $\tau_{\text{max}}$ is the maximum channel delay. Channel estimation in orthogonal frequency-division multiplexing (OFDM)
systems can get a noisy observation of CFR [60]–[63], which can be written as

$$\hat{H}(f, t) = H(f, t) + \hat{\omega}(f, t),$$  \hspace{1cm} (12)$$

where $\hat{\omega}(f, t)$ is the noise effect in frequency domain.

CFR-based systems have been mostly implemented in IEEE 802.11 OFDM systems [30], [31], [64], [65], as it is convenient to extract channel estimation. Only the amplitude of the channel estimation is used in practical implementation [30], [31], [65] as the phase estimation is usually impacted by the time and frequency offset. CFR may also be estimated by comparing the frequency spectra of the transmitted and received signal [66]. Unlike the CIR, the powers of the channel responses of all the frequencies are identical in an uncorrelated scattering environment [51], which is beneficial for the improvement of KGR [30], [31].

Channel estimation information is not available in most WiFi network interface cards (NICs) with the current exception of Intel WiFi Link 5300 wireless NIC [67]. Customized hardware platforms are also able to provide CSI, such as universal software radio peripheral (USRP) [68] and wireless open-access research platform (WARP) [69].

**B. RSS**

The transmitted signal $x(t)$ experiences the multipath channel and the received signal can be written as

$$y(t) = \int_{0}^{T_{\text{max}}} h(\tau, t)x(t - \tau)d\tau + n(t),$$  \hspace{1cm} (13)$$

where $n(t)$ is the noise effect. The instantaneous power of the signal $|y(t)|^2$ is usually not reported by NICs and transceivers. However, the average power level is usually available and referred as RSS.

RSS is currently the most popular channel parameter used in key generation, especially for practical implementation due to its availability. Most RSS-based key generation systems are applied either in IEEE 802.11 systems [20]–[22], [50], [70] or in IEEE 802.15.4 systems [23]–[27], [71].

RSS is a coarse-grained channel information metric and only one RSS value can be obtained from each packet, which limits the KGR. In addition, RSS is vulnerable to predictable channel attacks [21], [29]. What’s more, whilst there are lots of practical implementations, the theoretical modelling and analysis of RSS has not been reported yet. Finally, RSS may be interpreted in different ways, which requires special attention when the devices are provided by different manufacturers [21], [70], [72].

**C. Summary**

CIR $h(\tau, t)$ is the intrinsic random source for both CSI-based and RSS-based key generation systems. The parameters measured by users may be different but are always a function of $h(\tau, t)$.

The selection of the channel parameters for key generation will mainly be determined by the wireless techniques adopted. For example, RSS is available in all wireless systems, including systems modulated by direct sequence spread spectrum (DSSS) or frequency hopping spread spectrum (FHSS). The signal power is quite small in UWB systems but the CIR can be measured through the pulse transmission. A summary and comparison of key generation applications in different wireless networks is given in Table II.

**V. KEY GENERATION PROCEDURE**

Key generation procedure can be divided into four stages: channel probing, quantization, information reconciliation, and privacy amplification, as summarized in Table III and illustrated in Fig. 4. One user serves as the initiator, and the other as the responder. Without loss of generality, Alice is selected as the initiator. In order to simplify the flow chart, the stage synchronization between Alice and Bob is not shown.

**A. Channel Probing**

Channel probing is the key step to harvest the randomness from channel which requires two users to alternately measure the common channel through the received signals. As shown in Fig. 4, at time $t_{i,A}$, Alice transmits the $i^{th}$ probing signal to Bob who will measure some channel parameter through the received signal and store it in $Y_{i}$. At time $t_{i,B}$, Bob transmits

![Fig. 4. Key generation procedure](image-url)
## TABLE II
### KEY GENERATION APPLICATIONS IN WIRELESS NETWORKS

<table>
<thead>
<tr>
<th>Technique</th>
<th>Modulation</th>
<th>Parameter</th>
<th>Features</th>
<th>Testbed</th>
<th>Representative Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.11</td>
<td>MIMO OFDM</td>
<td>RSS, CSI</td>
<td>MIMO OFDM enables CSI measurements in both frequency and spatial domains</td>
<td>RSS: all NICs; CSI: Intel 5300 NIC, and customized hardware platforms, such as USRP [68] and WARP [69]</td>
<td>RSS-based: [22], CSI-based [30], [31]</td>
</tr>
<tr>
<td>a</td>
<td>OFDM</td>
<td>RSS, CSI</td>
<td>OFDM enables CSI measurements in frequency domain</td>
<td></td>
<td>RSS-based: [20], [21], [50], [70] CSI-based: [65]</td>
</tr>
<tr>
<td>g</td>
<td>OFDM, DSSS</td>
<td>RSS, CSI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>DSSS</td>
<td>RSS</td>
<td></td>
<td>MICAz [73], TelosB [74]</td>
<td>[23]–[27], [71]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Smartphones</td>
<td>[75]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[39], [40], [56]–[59]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[76]</td>
</tr>
</tbody>
</table>

### TABLE III
### KEY GENERATION PROCEDURE

<table>
<thead>
<tr>
<th>Stage</th>
<th>Purpose</th>
<th>Research Problems</th>
</tr>
</thead>
</table>
| | Channel measurements through the received signals. | • Channel parameters: The granularity of the chosen parameter determines the sampling efficiency.  
• Signal pre-processing: Improving signals’ cross-correlation by interpolation and/or filtering.  
• Channel probing rate: Removing redundancy between the measurements. |
| Channel probing | Conversion of channel measurements into binary values | • Selection of the threshold and quantization level.  
• Performance optimization between randomness, KGR and KDR. |
| | Reconciliation of the mismatch bits between Alice and Bob using protocols or error correction codes | • Optimization between the correction capacity and information leakage. |
| | Removal of information revealed in information reconciliation stage | • Cross design with information reconciliation.  
• Determination of the amount of leaked information. |

his \( i^{th} \) probing signal to Alice who will also measure the same channel parameter and store it in \( X'_i \). The sampling time difference \( \Delta t_i = |t_{i,A} - t_{i,B}| \) is deliberately kept smaller than the channel coherence time so the channel during the two probes can be regarded as constant. Alice and Bob will repeat the above process until sufficient results are collected.

Research in channel probing mainly considers channel parameter, signal pre-processing, and channel probing rate. The channel parameters valid for key generation have already been discussed in Section IV. Although the channel features at each end of the link are reciprocal, the measured received signals are asymmetric mainly due to non-simultaneous measurements (i.e., \( \Delta t_i \neq 0 \)) and the independent noise residing in the two separate hardware platforms. Therefore, signal pre-processing is used to improve the cross-correlation between the received signals, i.e., \( X_i = f(X'_i) \) in Fig. 4. The effects of non-simultaneous measurements and noise can be mitigated by interpolation [23], [24] and filtering [25], [30], [49], [77], [78], respectively.

There may be redundancy within the sampled measurements \( X^m \) and \( Y^m \), which are therefore resampled by a probing rate \( T_p \), chosen to be larger than the coherence time. An optimal probing rate is determined based on the modelling of the ACF of the signal [64] when the channel is changing in the same rate. However, the channel randomness is caused by unpredictable movement, leading to different change rate
of the channel condition. A fixed probing rate results in potential problems such as inefficient probing when the channel changes fast or redundancy between the samples when the channel changes slowly. Therefore, a proportional-integral-derivative (PID) controller-based adaptive probing system has been designed to tune the probing rate according to the channel conditions [79], which could generate key sequences both securely and effectively in a dynamically changing environment. Channel phase-based system in [47] does not suffer from the above problem as it can probe each other continuously. This is because besides the phase shift incurred by the channel, there is also a random initial phase introduced at each side, which is not affected by the channel coherence time.

B. Quantization

Similar to an analog-to-digital converter (ADC), quantization in key generation is also a method to map the analog channel measurements into binary values. The quantization level $QL$ in key generation has the same meaning as in ADC, which is the number of key bits quantized from each measurement. Due to the discrepancy between received signals of any two users, the quantization level is adjusted according to the signal-to-noise ratio (SNR) of the channel. In multi-bit quantization, Gray coding may be used in order to reduce the key disagreement.

The thresholds are the reference levels used to divide the measurements into different groups. Mean value $\mu$ (or together with standard deviation $\sigma$) [20], [21] and cumulative distribution function (CDF) [24] are commonly used to determine the thresholds. Mean value and standard deviation-based quantization scheme has simple implementation. The thresholds are determined as

$$\eta^+ = \mu + \alpha \times \sigma; \quad (14)$$
$$\eta^- = \mu - \alpha \times \sigma. \quad (15)$$

When $\alpha \neq 0$, the measurements between $\eta^+$ and $\eta^-$ will be dropped. The samples above $\eta^+$/below $\eta^-$ will be converted to 1/0. The CDF-based quantizer is detailed in Algorithm 1, which is more flexible as it can be designed as multi-bit quantizer. In addition, its thresholds can be tuned to guarantee the same proportion of 0s and 1s, an important feature for the randomness.

Algorithm 1 CDF-based quantization algorithm

1: $F(x) = \Pr(X^n < x)$
2: $\eta_i = F^{-1}\left(\frac{i}{2^n}\right), i = 1, 2, ..., 2^{QL} - 1$
3: $\eta_0 = -\infty$
4: $\eta_{2^{QL}} = \infty$
5: Construct Gray code $b_i$ and assign them to different intervals $[\eta_{i-1}, \eta_i]$
6: $K(j, QL) = b_i$, if $\eta_{i-1} \leq X_j < \eta_i$

In essence, the quantizer design is the adjustment of the quantization level and threshold in order to approach an optimal performance of the randomness, KGR and KDR. This results in different design variations, e.g., adaptively adjusting the threshold in order to follow the slow variation of the signal and finally avoiding long 1s or 0s and improving the randomness feature [21]; multi-bit quantization for a higher KGR [24]; dropping bits which are not all at the same side of the threshold for a better agreement [20]. Performance evaluation and comparison of the quantization schemes can be found in [41], [80].

C. Information Reconciliation

Although signal pre-processing algorithms can be adopted to improve the cross-correlation of the channel measurements, there may still be key disagreement between Alice and Bob after quantization. The mismatch can be corrected using information reconciliation techniques, which can be implemented with protocols such as Cascade [21], [49], [79], [81] or error correcting code (ECC) like low-density parity-check (LDPC) [51], [82], [83], BCH code [84], [85], Reed-Solomon code [86], Golay code [23], [26], [29], and Turbo code [87], etc. ECC-based reconciliation schemes are more efficient than Cascade, but they also leak more information [81] and have higher complexity [7]. The selection of the ECC depends on the complexity and correction capacity. For example, the maximum correction capacity rate of $[n, k, \eta]$ BCH code is given as

$$\zeta = \frac{t_{\text{max}}}{n} = \frac{2^m - 1 - \zeta}{2^m - 1}, \quad (16)$$

which approaches 0.25 when $m$ becomes large.

Secure sketch [84] is introduced as an example, which is also illustrated in Fig. 4. An ECC $C$ is adopted to correct the disagreement. Alice first randomly selects a codeword $c$ from $C$ and then calculates $s$ by exclusive OR-ing her key sequence $K^A$ with $c$, i.e., $s = \text{XOR}(K^A, c)$, which is then sent to Bob by the public channel. Bob will calculate $c'$ by exclusive OR-ing his key sequence $K^B$ with the correctly received $s$, i.e., $c' = \text{XOR}(K^B, s)$, and decode $c'$ from $c''$. He calculates $K^{B'}$ by exclusive OR-ing $c'$ with $s$, i.e., $K^{B'} = \text{XOR}(c', s)$. When the Hamming distance between $c$ and $c''$ is smaller than the correction capacity $t$ of the correction code, i.e., $\text{dis}(c-c') < t$, Bob can agree on the same key as Alice, i.e., $K^{B'} = K^A$.

The key agreement can be confirmed by implementing cyclic redundancy check (CRC) or other protocols and tools, e.g., automated validation of Internet security protocols and applications (AVISPA) software was used in [57]. There will be a risk that the KDR exceeds the correcting capacity rate of the information reconciliation which results in a failure and restart of the entire key generation process from channel probing.

D. Privacy Amplification

Some information is transmitted publicly in the information reconciliation stage, which can be heard by the eavesdropper as well. This can potentially compromise the security of the key sequence. Privacy amplification is then employed to remove the revealed information from the agreed key sequence at Alice’s and Bob’s side [88]. This can be implemented by extractor [47], or universal hashing functions, such as leftover hash lemma [21], [50], cryptographic hash functions (e.g.,
secure hash algorithm) [86], [87], and Merkle-Damgard hash function [79].

Privacy amplification and information reconciliation always appear together, which requires a cross design between these two stages. However, in practice, it is difficult to quantify the amount of the leaked information, or to identify where the leakage occurs in the data.

E. Summary

The key generation implementation is usually low cost, as it only requires non-complex operations, e.g., sampling and storing data in the channel probing stage. All these operations can be implemented using the off-the-shelf hardware, with only a change to the drivers.

The key generation procedures vary according to the system implementation. All the key generation systems need channel sampling and quantization while information reconciliation and privacy amplification may be not applied due to specific implementation and environment where the systems achieve perfect agreement after quantization [20], [25].

VI. PERFORMANCE OPTIMIZATION

The design criterion of key generation systems is to attain an optimal performance, which can be achieved by a careful consideration of the key generation stages.

KGR can be improved by the appropriate selection of channel parameter, channel probing rate, and quantization scheme, etc, which are summarized as follows:

- Randomness extraction from the fine-grained CSI [30], [31], [51], [64].
- More channel information extraction by leveraging multiple antenna diversity [22], [32].
- Introduction of relay nodes in order to make use of the channel information between the users and the relay nodes [48], [89]–[91].
- Employment of random initial phase in order to achieve multiple probes in one coherence time [47];
- Adaptive channel probing [79];
- Multi-bit quantization [21], [24];

The above methods can also be combined to further improve the KGR if the system permits. For example, a MIMO OFDM system can extract keys very efficiently as it is able to measure the CSI using multiple antennas [30].

The KDR will usually be high if the sampled channel parameters are quantized directly, especially in low SNR environments. The KDR can be reduced with the aid of the signal processing algorithms discussed in Section V-A and using a more robust quantization algorithm such as level crossing [20]. A KDR comparison of different quantization schemes can be found in [41], [80].

The three evaluation metrics, i.e., randomness, KGR, and KDR, contradict each other. For example, a fast probing rate will produce a high KGR but may result in temporal redundancy and compromise the randomness. A bigger quantization level can also produce a higher KGR, however, it may lead to a larger KDR especially in low SNR environments. Randomness usually cannot be compromised. Therefore, when designing a key generation system, a relatively optimal tradeoff should be achieved between KGR and KDR according to the system requirements and environments. For example, the KDR in [20] can be kept as low as $10^{-8}$ by adjusting the parameters in their level crossing algorithm but the KGR will be very small. A comparison of selected key generation systems in terms of techniques and performance is given in Table IV.

VII. APPLICATION SCENARIOS

Key generation has already been prototyped in several different areas. In this section, a review of applications in different environments is carried out and the challenge of each environment is discussed.

A. Wireless Local Area Networks (WLANs)

WLAN connectivity is now incorporated into most laptops, tablets and smartphones, making it the most popular wireless access technology. The main WLAN standards are IEEE 802.11 a/b/g/n operating in 2.4 GHz and 5 GHz bands. Due to its wide availability, many practical key generation implementations in WLAN have been reported. WLAN is primarily designed for indoor environments, where there is limited mobility. Therefore, in order to guarantee the randomness of the key sequence, the probe rate should be relatively large, as the channel can remain essentially static over long periods, which results in a low KGR.

RSS is available in all the WLAN standards and can be obtained in the commercial NICs. The research emphases are mainly on the improvement of KGR and decrease of KDR. For example, KGR is increased with the aid of multi-antenna [22] or adaptive channel probing [79], and KDR can be decreased by using a level crossing algorithm [20].

CSI-based systems are also feasible as IEEE 802.11 a/g/n use OFDM modulation and channel estimate can provide detailed channel information. Practical systems have been implemented using Intel WiFi Link 5300 wireless NIC and the KGR is much higher than RSS-based systems [30], [31]. The channel responses of individual OFDM subcarriers have also been leveraged for key generation [65] and an optimal probing rate can be tuned based on its theoretic model [64].

B. Wireless Sensor Networks (WSNs)

WSNs are widely used in environment monitoring, health care, or military [92], where there is a clear need to protect the data exchanged. The sensor nodes in WSNs are equipped with 802.15.4 transceivers operating in the 2.4 GHz to 2.8 GHz industrial, scientific and medical (ISM) band. RSS information is usually available in these transceivers and can be used to establish the keys in WSNs. However, the sensor nodes are static or with little movement, battery powered, and with low computational capacity, which places special requirements on the implementation. A key generation architecture for resource-constrained devices is proposed in [93].

In order to address the issue of the static nature of channel in WSN, randomness in the frequency domain is exploited [27]. The key generation system is designed to probe
TABLE IV
COMPARISON OF KEY GENERATION SYSTEMS

<table>
<thead>
<tr>
<th>Representative Work</th>
<th>Technique</th>
<th>Testbed</th>
<th>Parameter</th>
<th>KGR</th>
<th>KDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu et al. [30]</td>
<td>Fine-grained channel information; Multi-bit quantization (3-bit); MIMO (2 × 2)</td>
<td>Laptop with Intel WiFi Link 5300 NIC</td>
<td>CSI</td>
<td>360 bit/pkt</td>
<td>8%</td>
</tr>
<tr>
<td>Zeng et al. [22]</td>
<td>Multiple antenna diversity</td>
<td>Laptop with Intel WiFi Link 5300 NIC</td>
<td>RSS</td>
<td>10 bit/s</td>
<td>10%</td>
</tr>
<tr>
<td>Wei et al. [79]</td>
<td>Adaptive channel probing</td>
<td>Laptop with Atheros NIC</td>
<td>RSS</td>
<td>100 bit/s</td>
<td>N/A</td>
</tr>
<tr>
<td>Patwari et al. [24]</td>
<td>Multi-bit adaptive quantization</td>
<td>TelosB sensor mote</td>
<td>RSS</td>
<td>10 ~ 22 bit/s</td>
<td>0.54% ~ 2.2%</td>
</tr>
<tr>
<td>Mathur et al. [20]</td>
<td>Level crossing algorithm (^2)</td>
<td>Customized platform</td>
<td>CIR</td>
<td>1.17 bit/s ((m = 4))</td>
<td>15.85% (\sim 10^{-5}) ((m = 2 \sim 11, \text{SNR} = 30 \text{ dB}))</td>
</tr>
<tr>
<td>Ali et al. [25]</td>
<td>Channel sampling using regular data transmission; Employing Savitzky-Golay filter to mitigate noise effect.</td>
<td>MICAz sensor mote</td>
<td>RSS</td>
<td>0.037 ~ 0.205 bit/s</td>
<td>0 ~ 1.6%</td>
</tr>
</tbody>
</table>

\(^1\) KGR of a multiple antenna system \((3 \times 3 \text{ antenna pairs})\) is 4.5 higher than the KGR of a single antenna system

\(^2\) Level crossing algorithm requires a parameter \(m\), which is the number of the same consecutive bits in an excursion.

on different channels in order to extract the randomness from the frequency-selective fading. Signals with different carrier frequencies experience varied fading and thus the RSSs are different. However, this method requires a frequency-selective channel and the randomness is rather limited. After the initial generation from the randomness introduced by frequency selectivity, the refresh of the key becomes impossible if there is no further randomness caused by the movement or other changes to the wireless channel.

Body area network is a special application of WSN with sensors mounted on the body [94]. An RSS-based key generation system is implemented in body area networks [25]. In order to save energy, channel is sampled in the course of a routine transmission rather than dedicated communications. A Savitzky-Golay low pass filter is employed to mitigate the noise component so the system can achieve a high key agreement rate around 98%, or even 100% with a specific setting. Thus there is no information reconciliation and specific communication in their system. This is at the cost of very low KGR. It takes 15 to 35 minutes to generate a 128-bit key.

C. Vehicular Communication

As discussed in Section V-A, when \(\left| t_{i,A} - t_{i,B} \right|\) is much smaller than the coherence time, Alice and Bob can get correlated measurements in a slow fading channel. However, in vehicular communication, this is not the case because vehicles can move fast and the coherence time can be as short as a few hundred \(\mu\text{s}\). In a 20 MHz channel spacing IEEE 802.11 OFDM system, a packet with a maximum rate and minimum length results in an over-the-air time of 34 \(\mu\text{s}\), which cannot be considered negligible compared to the coherence time.

There has been research effort applying key generation in vehicular communication [49], [95], [96]. An RSS-based key generation system has been implemented using off-the-shelf IEEE 802.11 radios [49]. RSS measurements are found to be swamped in the high noise level. A weighted sliding window smoothing algorithm is adopted, where Alice and Bob work cooperatively to maximize the correlation coefficient of the quantized bit sequences. Level crossing is used in their system but is improved by dynamic parameter adjusting in order to adapt to the dramatic channel changes. They achieve a secure system with a bit rate around 5 b/s.

A novel distance reciprocity-based key generation is designed in [96]. While the distance may be measured using infrared and ultrasound localization systems, a wireless radios system equipped with TelosB motes is used as an example. The distance is measured through the long time-averaged RSS values therefore the fluctuations due to fading and shadowing are eliminated. As the distance does not change much in a short time interval, the legitimate users can agree on the same keys.

VIII. CONCLUSION AND SUGGESTIONS FOR FUTURE RESEARCH

Key generation from the randomness of wireless communication channels is a promising technique to share cryptographic keys securely between legitimate users. It is relatively easy to implement using off-the-shelf wireless NICs and can achieve information-theoretic security. This paper focused on the techniques of key generation systems, specifically, we reviewed the key generation principles, metrics and procedure. We also discussed methods to optimize the key generation performance. Different application scenarios were surveyed in order to clarify the features and challenges of each environment.

There are still open questions to be resolved in order to make key generation more robust [18], [42]. Some future research scopes are summarized below.

- Key generation in static environments. Although researchers have tried to introduce randomness into static channels by employing random beamforming [97], virtual channels [98] and jamming [85], [99], these methods are not generic as they either require multi-antenna [97], [98], aid from other nodes [85] or OFDM modulation [99]. The ability to operate in a static environment will be essential for the application of key generation systems.

- Group key generation. There are already some group key generation protocols [23], [26], [47], [100]–[102],
but most key generation systems can still only extract keys in pairs. Group key generation has a wide range of applications. For example, in ad hoc networks, all the users will have to exchange secured information and the network is quite dynamic as there may be many users frequently joining and leaving.

- Attacks against key generation systems. This research topic currently receives limited research input. Key generation is vulnerable both to passive eavesdropping [44], [103] and active attacks [104], [105], or combined [106]. Research into how we can subvert or defend against such attacks is essential if we are to construct robust and secure key generation systems.

REFERENCES


