Exploiting Wireless Channel Randomness to Generate Keys for Automotive Cyber-Physical System Security

Jiang Wan, Anthony Bahadir Lopez, Mohammad Abdullah Al Faruque
{jiangwan, anthl10, alfaruqu}@uci.edu
Department of Electrical Engineering and Computer Science
University of California, Irvine, Irvine, California, USA

ABSTRACT

Modern automotive Cyber-Physical Systems (CPSs) are increasingly adopting wireless communications for Intra-Vehicular, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) protocols as a promising solution for challenges such as the wire harnessing problem, collision detection, and collision avoidance. Regrettably, this new trend results in new security vulnerabilities that can put the safety and privacy of the automotive CPS and passengers at great risk. In addition, automotive wireless communication security is constrained by strict energy and performance limitations of Electronic Controller Units (ECUs) and sensor nodes. As a result, the key generation and management for secure automotive CPS wireless communication is an open research challenge. This paper aims to help solve these security challenges by presenting a practical key generation technique based on the reciprocity and high spatial and temporal variation properties of the automotive wireless communication channel. To validate the practicality and effectiveness of our approach, we have conducted separate real-world experiments with automobiles and with RC cars. Lastly, we demonstrate through simulations that we can generate keys with high security strength (keys with 67% min-entropy) with up to 10X improvement in performance and 20X reduction in code size overhead in comparison to the state-of-the-art security techniques.

CCS Concepts

•Security and privacy → Key management; Mobile and wireless security; •Networks → Cyber-physical networks; •Computer systems organization → Embedded software;

Keywords

Security; Automotive Cyber-Physical Systems; Wireless Communication; Key Generation; Key Exchange; Symmetric Cryptographic Algorithm;

ICCPS '16 April 11–14, 2016, Vienna, Austria

DOI: 10.1145/1235

1. INTRODUCTION AND RELATED WORK

Wireless technologies are widely implemented in automotive Cyber Physical Systems (CPSs) for infotainment applications such as navigation schemes, hands-free calling, and satellite radio. However, due to the wire harnessing problem [18], recent intra-vehicular sensor networks are also adopting wireless technology to greatly reduce the total weight of the vehicle and the complexity of adding newer features during the design time. As a result, using wireless technology may greatly enhance the functionality and efficiency of the automotive CPS [7,9]. As an example, the Tire Pressure Monitoring Systems (TPMS) use wireless sensors to inform both the automotive system and the passengers about valuable information such as temperature and tire pressure. Applying wireless technology to detect collisions is a promising solution to increasing traffic efficiency and reducing the number of accidents, where more than 80% are caused by drivers [37]. For this reason, national agencies such as the U.S. Department of Transportation are developing Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications [12] to provide entertainment, road condition information, collision detection and avoidance measures (all of which can enable the realistic use of autonomous driving). Figure 1 provides an illustration for such a scenario.

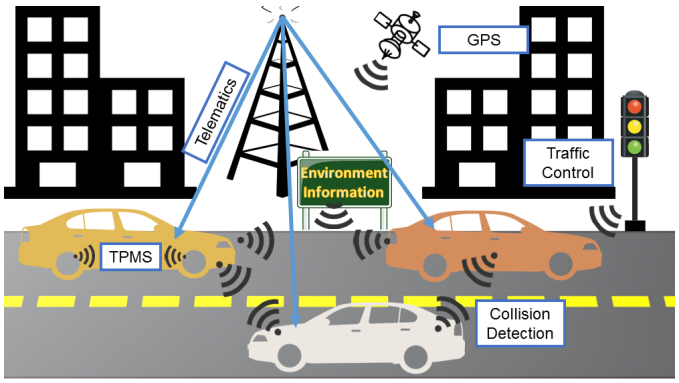

Figure 1: Examples of V2V and V2I Applications.

This new paradigm dealing with the connection between traditionally isolated automotive systems and the outside world over insecure wireless channels introduces several security concerns such as leakage of private information of passengers and direct influence over the automotive system's behavior [6,21,38]. In fact, this type of security concern began in the mid-1990s when many automotive systems used

Remote Key-less Entry (RKE) [5]. Afterward, in 2005, a successful Texas Instrument transponder hack indicated a potential security threat to millions of automobiles [5]. And more recently in 2010, researchers demonstrated the possibility of an attack that captures and reads TPMS communication packets from an automobile up to 40 meters away. They also presented the capability of injecting fake packets to trigger the target automobile's TPMS warning signal. Since incoming safety-critical V2V/V2I applications will also inherit these aforementioned security challenges and concerns, researchers from the European Telecommunications Standards Institute (ETSI) proposed the following security objectives for these applications: confidentiality, integrity, availability, accountability, and authenticity (for more details, please see the technical report [10]).

We summarize that for wireless communication in automotive CPS, messages will need to be authenticated and sometimes encrypted depending on the confidentiality requirements of applications [30]. As a simple example, account information will need to be encrypted in applications like Electronic Toll Collection (ETC) [27]. It is important to note that these security objectives apply to resource-limited (in terms of computational power, energy consumption and memory size) time-critical embedded devices (such as microcontrollers, sensor-nodes, etc.) and resource-limited non-time-critical devices (infotainment systems). For the purposes of our paper and its importance in keeping passengers safe, we focus on the former of these two device types: resource-limited and time-critical.

A typical automotive design needs to provide security for about 20 years or more [30, 35], implying the necessity of a reliable cryptographic design to achieve the aforementioned security objectives. Cryptographic algorithms fall under two categories: Asymmetric and symmetric. As seen in Table 1, symmetric algorithms (like AES) have very high performance and smaller energy overhead [25] in comparison to asymmetric algorithms (like RSA and Elliptic Curve Cryptography). However, the major problem of using symmetric algorithms is that both communicating parties must share a secret key before any secure communication [30]. **Therefore, secret key exchange is considered as a challenging problem in automotive wireless applications.** Although asymmetric algorithms do not require a shared secret key for secure communication, they are too slow for most of the time-critical automotive CPS applications [30], and they also consume more computational power and more memory space in comparison to other algorithms [25]. Thus, in the state-of-the-art approach, research groups and government organizations are proposing the use of hybrid solutions [30, 31], where a symmetric key is generated from a random number generator or a Key Encapsulation Mechanism (KEM) [13] and exchanged through an asymmetric algorithm. As a result, higher performance can be achieved with symmetric encryption of both small and large data.

However, there are still three major limitations in the current hybrid approach. **First**, this solution requires a key exchange session using an asymmetric algorithm before the data transmission session. This overhead is estimated to be up to several seconds [30] and is generally not acceptable for safety related applications which require a reaction time of 50 to 200 milliseconds [30]. **Second**, the hybrid solution requires an implementation of asymmetric algorithm in the embedded devices, thus causing non-negligible memory space overhead. **Third**, similar to symmetric algorithms, the hybrid solution generally needs a random number generator that produces symmetric keys with high entropy. Traditionally, the generation of random bits rely on a software-based pseudo random number generator or user given inputs. This approach, however, cannot provide enough entropy[1] due to its high level of predictability and determinism [23].

Table 1: Comparison of Existing Cryptographic Algorithms

	Symmetric	Asymmetric	Hybrid
Authentication	Message Authentication Code (MAC)	Digital signature	Digital signature on keys MAC on data
Confidentiality	Encryption of data	Encryption of small data	Encrypt keys with Asym. Encrypt of data with Sym.
Performance	Very fast	Slow	Medium
Code size	Thousands of bytes	Thousands of bytes	Thousands of bytes
Key size	32-256 bits	ECC: 256-384 bits RSA: 1024-3072 bits	512-3072 bits for Asym. 32-256 bits for Sym.
Key management	Random key generation Pre-shared secret key	None	Random key generation

To solve this problem, researchers have been looking toward physical randomness as a high entropy source for random number generation. One of the products of their ideas is the Physical Unclonable Function (PUF), a function based on physical characteristics that are practically impossible to be duplicated by any attackers. Recently, researchers have proposed to use PUFs that can generate secret keys by extracting randomness from the physical environment [29, 33]. Similarly, it is possible to use the wireless communication channel as a source of physical randomness to generate secret keys. Most of the state-of-the-art theories and practical methods for generating secret keys using physical characteristics of the wireless channel have been proposed within the last decade [4, 15, 19, 24, 28, 36, 40, 41]. The success of generating dynamic keys from the wireless communication depends on three properties: 1) reciprocity of the radio wave propagation, 2) temporal variations, and 3) spatial variations in the wireless channel (see details in Section 2.1). Besides most of the theoretical works [4, 19], some practical implementations have been demonstrated in sensor network applications [1, 15, 28], and they rely on the Multiple-Input and Multiple-Output (MIMO) approach or collaborations among multiple wireless nodes to create secret keys with higher entropy. Work in [40] has provided an implementation on V2V/V2I applications. However, it mainly focuses on the comparison between different key generation algorithms and adequately model the spatial and temporal variations of the automotive wireless channel. Moreover, the authors do not consider practical challenges such as the real-time requirements for safety-critical V2V applications.

1.1 Problem and Research Challenges

In summary, solving the limitations of the above-mentioned state-of-the-art approaches poses the following key challenges:

1. **Finding a reliable high entropy source** to generate secret keys for symmetric cryptographic algorithms, for ensured secure wireless communication in automotive CPS.

[1]Entropy is the quantified value of the randomness for a set of bits.

2. **Finding a low cost solution** in terms of performance and memory size for the exchange of symmetric keys in automotive CPS.

1.2 Our Contributions and Concept Overview

To address the above-mentioned challenges, we propose a novel technique to generate symmetric keys from the physical randomness of automotive wireless communication under tight memory and performance budgets. **To the best of our knowledge we are the first to demonstrate, through realistic automotive modeling, simulation and experiments, that higher level of entropy may be obtained from the moving and changing environment to generate symmetric secret keys for automotive CPS wireless communication practically.** The contributions of this paper are as follows:

1. **Wireless communication system models (Section 2)** including the channel, device, and attack models from the security perspective.

2. **A physical layer key generation technique (Section 3)** for automotive wireless communication between an automotive sender and an automotive receiver.

3. **Real world experiments to demonstrate the practicality of our proposed key generation technique (Section 4)**.

2. SYSTEM MODELING

2.1 Wireless Communication Model

We provide a sender-to-receiver model of an automotive wireless communication system, where an ECU or sensor-node inside an automobile A is communicating with another automobile or infrastructure B in the presence of an eavesdropper from automobile E. In this model, the sending signal S_A from A over the wireless channel will be received by B and E as follows:

$$\begin{aligned} R_{A\to B}(t) &= H_{A\to B}(t) \times S_A(t) + N_{A\to B}(t); \\ R_{A\to E}(t) &= H_{A\to E}(t) \times S_A(t) + N_{A\to E}(t); \end{aligned} \quad (1)$$

where H is the channel gain and N is the zero mean additive Gaussian noise [39]. If B responses with a signal R_B to A, then the received signals by A and E may be modeled as follows:

$$\begin{aligned} R_{B\to A}(t) &= H_{B\to A}(t) \times S_B(t) + N_{B\to A}(t); \\ R_{B\to E}(t) &= H_{B\to E}(t) \times S_B(t) + N_{B\to E}(t); \end{aligned} \quad (2)$$

Suppose, $S_A(t)$ and $S_B(t)$ are two probe signals, known to A, B, and E. Based on the received signal $R_{A\to B}(t)$, $R_{B\to A}(t)$, $R_{A\to E}(t)$, and $R_{B\to E}(t)$, the channel gain can be estimated as $H'_{A\to B}(t)$, $H'_{B\to A}(t)$, $H'_{A\to E}(t)$, and $H'_{B\to E}(t)$, respectively. Due to the **reciprocity** [39] of the wireless channel, if A and B send the probe signals to each other within the coherence time[2] of the wireless channel, we may assume the estimated channel gain as: $H'_{A\to B}(t) \approx H'_{B\to A}(t)$. However, from the eavesdropper's side, the estimated channel gain $H'_{A\to E}(t)$ and $H'_{B\to E}(t)$ will be independent of $H'_{A\to B}(t)$ and $H'_{B\to A}(t)$, if the eavesdropper is a few wavelengths [39] away from the legitimate wireless channel. Utilizing this concept, the channel gain ($H'_{A\to B}(t)$ and $H'_{B\to A}(t)$) may be used to extract secret keys (our proposed physics layer key generation, for details see Section 3) for standard symmetric cryptographic algorithms.

The wireless communication channel gain varies over time due to temporal or spatial variations in the environment. Typically, the channel may be modeled with a fast fading model or a slow fading model depending on the changing speed of the environment [32]. For automotive CPS, if there exists a velocity difference between two communicating automotive wireless nodes, we use a **fast fading model (temporal variation)**, otherwise, we use a **slow fading model (spatial variation)**.

For the fast fading model, we use a *Rayleigh fading channel* [32] which provides a general-case wireless communication model suitable for automobile [32] in an urban driving profile. The *Rayleigh fading channel* models the *Doppler shift effect* [32] due to the different speeds between two communicating wireless nodes. In this model, the channel gain H should follow the following Probability Distribution Function (PDF):

$$PDF_H(H,\sigma) = \frac{H}{\sigma^2} e^{-x^2/(2\sigma^2)} \quad (3)$$

where σ is an environment-related parameter. Due to the *Doppler shift effect*, H only remains constant within the coherence time [32] T_c (see the following Equation).

$$T_c \approx \frac{0.423}{f_d} \quad (4)$$

here, f_d is the maximum *Doppler frequency* during the communication process. In an automotive wireless communication between A and B, f_d may be decided by the speed difference of the two communicating automobiles ΔV_A as shown below:

$$\begin{aligned} f_d &= \frac{\Delta V}{c} f_0 \\ \Delta V &= |V_A - V_B| \end{aligned} \quad (5)$$

where c is the speed of light and f_0 is the communication frequency.

Therefore, the model reflects that the channel changes roughly every time interval of T_c. In other words, the higher the ΔV is, the more frequently the channel is changing and the quicker channel-based key may be generated. However, in order to extract information from channel gain H to generate the key, the generation of 1-bit key must be constrained within a given time period, T_c. Otherwise, the channel will change and it will result in a mismatch between the generated keys from both the communicating automotive wireless nodes.

When the relative speed between the communicating automotive wireless nodes is low, $\Delta V \approx 0$, the fast fading model will not work. Therefore, we use a general slow fading model for the wireless communication. In a slow fading channel, the channel gain remains correlated in time, if the channel does not move over a certain distance. This distance is defined as the coherence length L_{cor}. On the other hand, the model assumes that if the channel moves further than L_{cor}, the channel gain will become independent of time.

[2]In wireless communication system, coherence time is the time duration over which the channel impulse response is considered to be not varying.

Therefore, considering the velocity of the automobile V, we may calculate the coherence time for a slow fading channel as follows:

$$T_c \approx \frac{L_{cor}}{V} \tag{6}$$

Similar to the fast fading channel model, the slow fading channel also changes roughly every time interval, T_c. In Equation 6, L_{cor} is decided by the environment. Therefore, the higher the V is, the more frequently the channel is changing. In this paper, the time varying channel gain for a slow fading model follows the log-normal distribution as shown below:

$$PDF_H(H,\sigma) = \frac{1}{H\sigma\sqrt{2\pi}} e^{-\frac{ln(H)}{2\sigma^2}} \tag{7}$$

2.2 Security Strength Model

Security strength indicates the amount of work that an attacker needs to break the cryptographic algorithm. Since most if not all cryptographic algorithms require a secret key. According to the National Institute of Standards and Technology (NIST) standard [2], an algorithm is defined to have "X-bits security strength" if it takes to try "X" number of symmetric keys that has no short cut attack (only brute-force attack).

To directly measure the randomness and security strength of the key, we use the concept of min-entropy [14]. The min-entropy is a worst case entropy estimation, and provides the lower bound of a cryptographic key's randomness. Let K be the set of all possible keys generated randomly, the min-entropy is defined as follows:

$$H_\infty = H_{min} = -log(\max_{k \in K} Pr[K = k]) \tag{8}$$

where, $Pr[K = k]$ is the probability of generating key $k \in K$.

Thus, we model the security strength $Security_{str}$ of a cryptographic algorithm using the average min-entropy on each bit of the key as follows:

$$Security_{str} = H_{min}/Key_{size} \tag{9}$$

where, Key_{size} is the size of the key and $Security_{str}$ is a value ranged from 0 *to* 1 in the unit of bits. For example, a 128-bit key with $Security_{str} = 0.5$ *bit* will have 64 *bits* of min-entropy.

2.3 Attack Model

In this paper, we consider **a non-intrusive wireless attack model** where the attacker tries to decipher the message by sniffing the legitimate wireless channel through a third wireless channel. We assume that the attacker can capture all the wireless packets sent through the wireless channel and the attacker knows all the information about the communication system including modulation/coding techniques and cryptographic algorithms. Therefore, in such a scenario, if the attacker can get the related key, the system will be broken. As a result, we define attack strength $Attack_{str}$ as the number of bits the attacker can decipher with a given amount of computing hardware resources.

We note that intrusive attack models are not considered in this paper since they typically requires the use of highly expensive and impractical devices and are challenging to implement for attacks on specific automobiles in real scenarios.

3. WIRELESS CHANNEL-BASED GENERATION

We present our wireless channel-based key generation algorithm with a V2V wireless communication example shown in Figure 2. In this example, both Alice and Bob are driving, where Alice's automobile (A) is communicating with Bob's automobile (B). Assume the driving speed for A and B is V_A and V_B, respectively and the speed difference between these two moving automobiles is ΔV. The coherence time T_c of the communication channel between A and B may be estimated using Equation 4 and Equation 5. Now, if A and B want to generate a key with size of K_{size}, they need to exchange a set of pre-defined probe signals (can be any kind of signals) to evaluate the randomness of the channel gain H using Equation 3. In order to have low key mismatch rate, as presented in the previous section, they must exchange each probe signal within the T_c time interval. Meanwhile, in order to keep bits of the generated key uncorrelated to each other, the time interval defined as τ_{step} between exchanging each probe signal should be no less than T_c. A predefined group

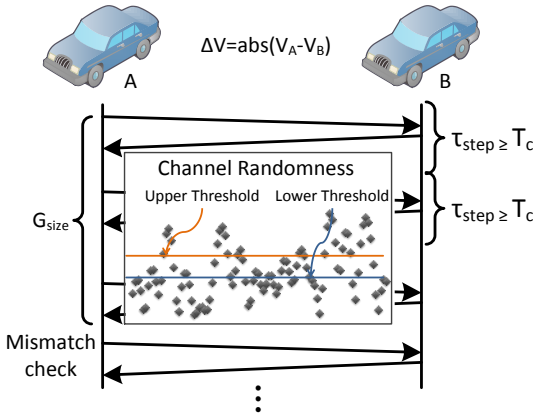

Figure 2: An Example of the Physical Layer Key Generation for a V2V Scenario.

of probe signals with a group size G_{size} is sent for evaluating the channel randomness. After the group of probe signals is exchanged, a set of measured Received Signal Strength (RSS) is used to generate bits. We then implement a mismatch check step to remove the mismatching bits. Once a set of bits is generated, the set's size must be more than or equal to the required key size, K_{size}. If the response is negative, the whole process iterates again until the key is sufficiently long. The details of the wireless channel-based key generation algorithm is presented in Algorithm 1.

Lines 3-5 takes $(G_{size} \times \tau_{step})$ time to collect all Received Signal Strength (RSS) values from the wireless channel. Line 6 filters the low frequency parts of the collected RSS values with high pass filter defined by its impulse frequency response $H_{highpass(t)}$. The filtered signal values $RSS_{filtered}$ contains all the information that we need to extract the keys. Lines 7-10 calculates the thresholds used for generating bits from the received RSS values. As proposed by [26], we use two thresholds for key generation. Every RSS value greater than the upper threshold Th_{up} is considered as 1 and every RSS value less than the lower threshold Th_{lo} is considered as 0. Any value in between Th_{up} and Th_{lo} is discarded. The thresholds Th_{up} and Th_{lo} are calculated by the equation in

Algorithm 1: Algorithm for Physical Layer Key Generation of an Wireless Automotive CPS.

Input: Measured Signal Strength: RSS
Input: Sample Time Step: τ_{step}
Input: Group Size: G_{size}
Input: Threshold Parameter: α
Input: Required Key Length: L_{key}
Output: Generated Key: Key

```
1  L = 0; Key = 0; RSS_set=∅; RSS_filtered=∅; Key_idx=∅;
2  while L < L_key do
3      for i=1 to G_size do
4          RSS_set = RSS_set ∪ RSS;
5          Wait(τ_step);
6      RSS_filtered = RSS_set * H_highpass(t);
7      MeanValue = Average Value of RSS_filtered;
8      Var = Variation Value of RSS_filtered;
9      Th_up = MeanValue + α * Var;
10     Th_lo = MeanValue − α * Var;
11     foreach RSS_j ∈ RSS_filtered do
12         if RSS_j > Th_up then
13             Key = (Key << 1) + 0;
14             L = L + 1;
15             Record j in Key_idx;
16         else if RSS_j < Th_lo then
17             Key = (Key << 1) + 1;
18             L = L + 1;
19             Record j in Key_idx;
20     Exchange Key_idx;
21     Remove mismatch bits from Key;
22 return Key;
```

Line 9 and Line 10, respectively, based on the mean and variation value of the collected RSSs. Lines 11-19 check all the collected RSSs and generate a key Key with length L. Notice that, Line 15 and Line 19 also record the index of the suitable RSSs for generating keys. The indexes defined by Key_{idx} from the two communicating automotive wireless nodes are exchanged in Line 20 and in Line 21, Key_{idx} is used to remove all mismatching bits. A shared key is generated among both communicating parties but algorithm will iterate if $L < L_{length}$.

4. RESULTS AND EVALUATION

4.1 Key Generation Simulation

For evaluation purposes, we have developed an automotive wireless channel model together with our wireless channel-based key generation algorithm in MATLAB [20]. The parameters for average driving speed is set to 50 miles per hour (MPH), and the coherence length for slow fading is set to 20 meters for urban environment. The simulation evaluates the key generation time with respect to the relative speed between two communicating nodes (0 to 75 MPH in our setup). Moreover, the simulation is conducted with respect to 6 different key sizes (56, 112, 128, 168, 192, 256 bits) proposed by the security standards from NIST [3]. The summarized simulation setup is presented in Table 2.

As presented in Figure 3, our key generation algorithm has negligible performance (10 to 100 milliseconds) overhead when the relative speed is high due to the fast fading of the wireless channel. This implies that our key generation algorithm may work well for V2V and V2I applications. On the other hand, for intra-vehicle communications where the relative speed between two nodes is around zero, our simulation results show a longer key generation time (around 1 to 2 minutes). Compared to the time that the generated key will be effective, which is typically several hours to even months, several minutes can also be considered as negligible. Although in some cases, several seconds of overhead for key generation is not acceptable (e.g. safety related applications), our wireless channel-based key generation algorithm can be applicable in most of the V2V or V2I-related CPS applications.

Table 2: Experimental Setup For Our Key Generation Algorithm.

Tested Key Length (bits)	56, 112, 128, 168, 192, 256
Relative Speed Range (km/h)	0 to 120
Average Speed (km/h)	60
Signal to Noise Ration (dB)	80 [37]
Coherence Length (m)	20 [37]
Group Size (bits)	10

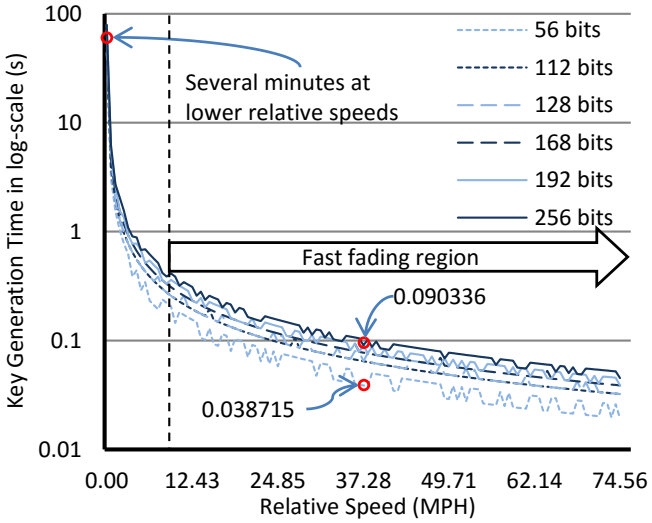

Figure 3: Simulation Results of Our Key Generation Overhead.

We additionally conducted simulations to confirm the independence of two generated keys from two different automotive wireless communication channels to demonstrate that the attacker cannot easily retrieve the key by eavesdropping. The simulation setup is presented in Figure 4. Three automobiles (with driving profiles) are modeled and connected using the developed wireless channel models. Two wireless channel models are instantiated in the simulation, where one connects the automobile models with *Drive Profile 1* and *Drive Profile 0* to each other, and the other connects the automobile models with *Drive Profile 1* and *Drive Profile 2* with each other.

As listed in Figure 3, per each key size we conduct the simulation with different relative speeds. For each relative

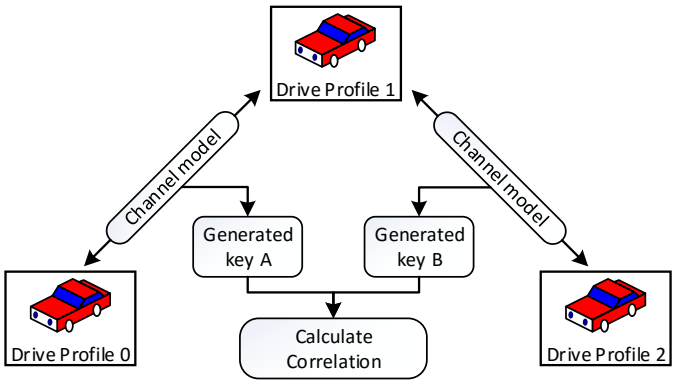

Figure 4: Simulation of Generating Two Secret Keys at the Same Time.

speed and key size, we run the simulation 100 times to generate two vectors of keys from two wireless channels at the same time. Then, we calculate the Pearson's correlation coefficient [17] between these two vectors. The calculated correlation results are presented in Table 3. From the simulation results, we can observe that all the correlation results are close to zero (the highest correlation value is just 0.0392). These results demonstrate that the keys generated from two automobiles connecting to the same target through wireless communication highly independent, which implies that the attacker cannot retrieve the key generated from the legitimate wireless channel.

Table 3: Correlations Results of the Keys Generated from Two Communication Channels.

Key size / Relative speed	56 bits	112 bits	128 bits	168 bits	192 bits	256 bits
0 km/h	0.0102	0.0121	0.0132	0.0207	0.0305	0.0233
20 km/h	0.0271	0.0053	0.0361	0.0221	0.0337	0.0125
40 km/h	0.0264	0.0132	0.0026	0.0125	0.0177	0.0283
60 km/h	0.0176	0.0177	0.0056	0.0293	0.0334	0.0268
80 km/h	0.0039	0.0236	0.0167	0.0392	0.0147	0.0244

4.2 Experiments with RC cars

Going further than simulation, we wanted to conduct real world experiments to validate the proposed physical layer key generation technique. In our first experiment, we used three systems made up of RC cars and Raspberry Pis and connected them through Bluetooth. As presented in Figure 5, we mounted the Raspberry Pi systems on top of the RC cars. On each Raspberry Pi board, we use USB Bluetooth dongles to establish the wireless communication. In this experiment, one of our objectives has been to confirm nearly zero correlation between generated keys from different wireless communication channels within a short distance, but longer than a few wavelengths. Therefore, we have mounted two Bluetooth dongles on *Car 1* (as shown in Figure 5) to establish two wireless communication channels between *Car 1* and *Car 0*, and *Car 1* and *Car 2*. During runtime, all the Received Signal Strength Indicator (RSSI) values from each Bluetooth dongle are collected by a computer through a separate WiFi connection (as shown in Figure 5). For each Bluetooth communication channel, we collected RSSI values from both communication nodes. Thus, in total there had been four sets of RSSI values collected from all the Bluetooth dongles. Although for this experiment, we have used a computer to execute the key generation algorithm and have analyzed its results, we have also implemented the same key generation algorithm in the Raspberry Pis.

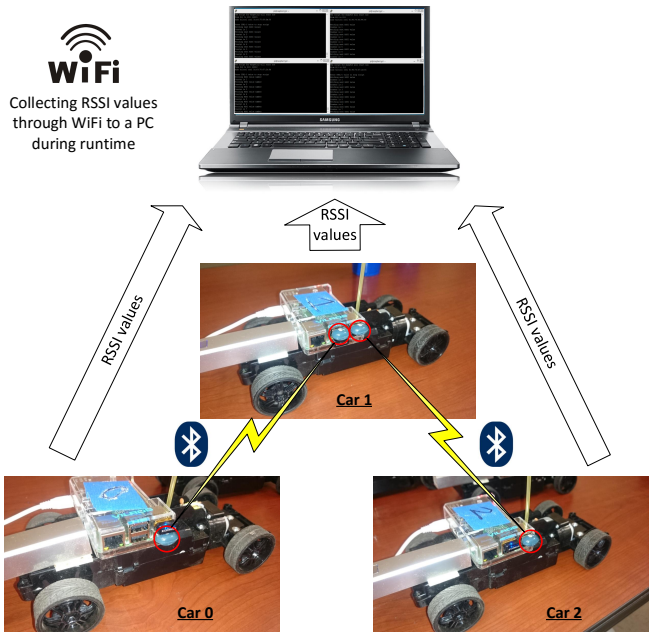

Figure 5: RC Car Experiments Setup

We consider the experimental environment with RC cars as a slow fading one because they move at low speeds (less than 5 MPH) and within a distance of 10 meters from each other in open areas with few moving objects around them.

200 samples of the collected RSSI values are presented in Figure 6. From the results, we can easily observe that the RSSI values collected at *Car 1* and *Car 0* for the wireless communication between *Car 1* and *Car 0* are highly correlated with each other (shown in red lines). The same results are also found for the wireless communication between *Car 1* and *Car 2* (shown in blue lines). These results clearly show the reciprocity characteristic in the wireless communication channel. Moreover, we have found that even with very short distances, the generated RSSI values from two different wireless communication channels have nearly zero correlation, thus supporting the assumption that "an attacker that is at a position of **several wavelengths distance** away will experience different wireless channel characteristics, and therefore cannot obtain or predict the secret keys" mentioned earlier in this paper is valid for automotive wireless communication systems. Table 4 shows the generated 64 bits of keys based on the collected 200 samples of data. Notice that, we use 50 as the probe signal group size for the key generation algorithm in this experiment.

4.3 Experiments with Real Automobiles

In order to further validate that our proposed key generation technique is practical, we have also performed ex-

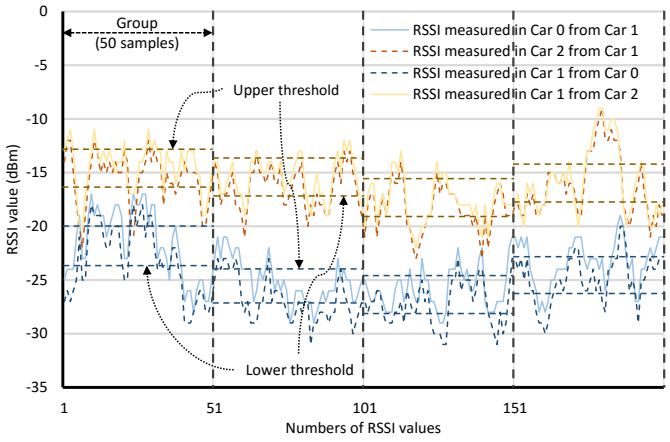

Figure 6: Collected 200 Samples of RSSI Values from the RC Car Experiments

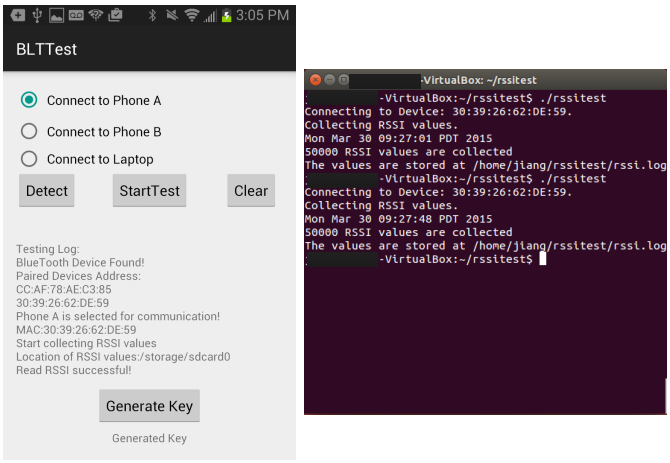

Figure 7: Our Developed Applications for Measuring Bluetooth RSSIs in Real Time.

Table 4: Generated 64-bit Keys from the RSSI Values

	Generated 64-bit Keys
Car 1 from Car 0	1100000110000000_0000000100000110_ 0000000010000000_0000011111111111
Car 0 from Car 1	1100000110000000_0000000100000110_ 0000000010000000_0000011111111111
Car 1 from Car 2	0000001111111111_1111000000000000_ 0000011111100000_0000011110000011
Car 2 from Car 1	0000001111111111_1111000000000000_ 0000011111100000_0000011110000011

periments in real driving scenarios. For the experimental purpose, we have used the Bluetooth from our laptops and android phones as the wireless channel for testing. We have developed applications in both Android phones and laptops (demonstrated in Figure 7) to measure the RSSI of the Bluetooth connection between two devices in real time.

As presented in Figure 8, we have placed the mobile devices in two automobiles, and checked the RSSI values from both automobiles in real time during the driving. Moreover, we use the proposed key generation algorithm to generate keys from the collected RSSI values. We demonstrate the RSSI values received from both sides of the mobile devices during a period in Figure 9. We may observe that there exists several mismatched signals in Figure 9, this is primarily because Bluetooth communication is not stable between the two fast automobiles resulting in some loss of RSSI data. However, our Algorithm 1 already considers these mismatches and handles them well. In this experiment, the RSSI value sampling rate is 10 milliseconds due to the limitation of the Bluetooth devices (mobile phone and laptop in this experiment).

The experiments are conducted on three relative speeds of 20, 10, and 2 MPH while driving in the same direction to collect the RSSI values and to generate 6 different sizes of keys (see Figure 10). We want to note that sampling takes the majority of time and our algorithm's execution time is negligible (constant).

4.4 Comparison to the State-of-the-Art

In this section, we compare our works with the state-of-

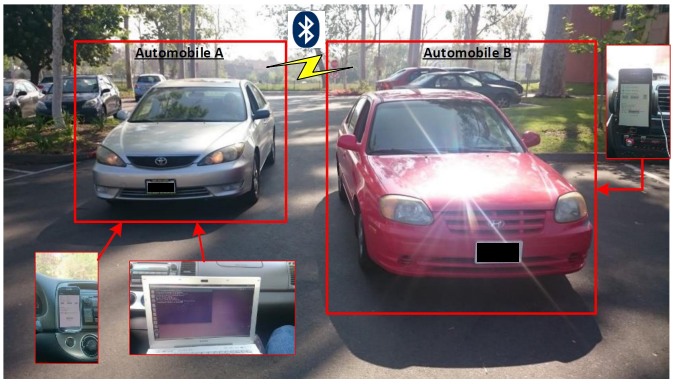

Figure 8: The Real World Experiments Using Phones and Laptops.

the-art hybrid cryptographic algorithms [22, 30] to evaluate the security strength, performance and code size overhead for automotive wireless communications.

1) **Security comparison:** We compare the security strength of our algorithm's generated keys to those produced by the state-of-the-art. We evaluate and compare the security strength using the proposed average min-entropy as the Key Performance Indicator (KPI).

Traditional wireless sensor communication uses pre-distributed keys [23] for their practicality (in terms of the simplicity of the key management scheme) in achieving real-time communication. However, since the pre-distributed keys and associated algorithms are predictable, the pre-distributed key approaches have little to no entropy [23]. In comparison to the traditional approach, state-of-the-art PUF-based approaches, such as the SRAM-PUF [14], can generate keys with high average min-entropy.

To estimate the average min-entropy of our key generation algorithm, we run our simulation 12800 times to generate $100 * 2^8 = 12800$ number of 8-bit keys. Based on the collected keys, we calculate the probability Pr_{max} of the key with the highest likelihood and apply this Pr_{max} to Equation 8. The results in Figure 11 show the average min-entropy of our technique and compares it with other

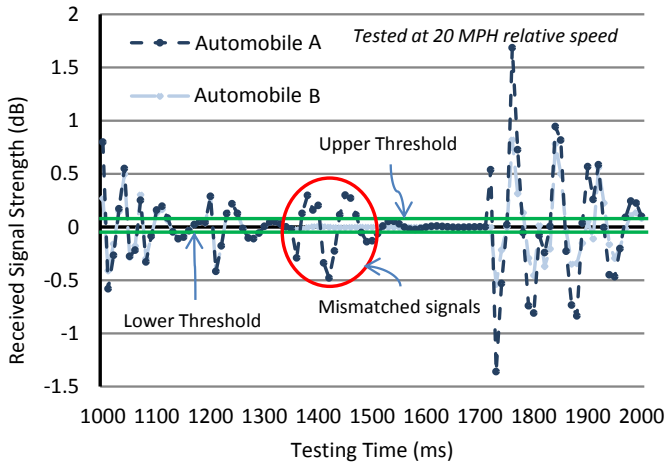

Figure 9: The Collected RSSIs from Both the Sender and the Receiver.

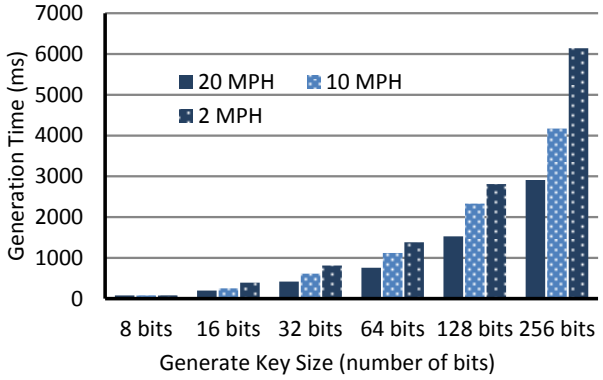

Figure 10: The Experimental Key Generation time at Different Speeds.

well-known techniques such as pre-distributed keys, Latch-PUFs, DFF-PUFs, and SRAM-PUFs. Note that our algorithm can generate keys with security strength close to that of some of the best PUF-based approaches (up to 67% average min-entropy for 8-bit keys[3]). Although some of the PUF-based approaches (e.g. SRAM-PUF) can generate keys with higher average min-entropy (since the number of 0 and 1 bits tend to be around the same), our algorithm has the advantage of generating keys by directly accessing the communication channel without needing a special physical process such as SRAM rebooting (for SRAM-PUFs). While the average min-entropy (67%) is not as high as some of the PUF-based approaches, it can be easily increased by adding hardware or algorithm improvements.

2) **Performance overhead comparison:** From the performance point of view, we know that wireless channel-based key generation algorithm has the advantage of not needing the time-consuming key exchange step of asymmetric and hybrid techniques. Thus, we compare our algorithm's key generation time to the execution time of two of the most popular asymmetric cryptographic algorithms (RSA

[3]according to [34], the average min-entropy increases with the respect to the size of the key.

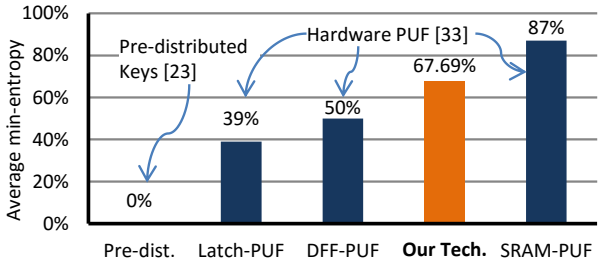

Figure 11: Estimated Average Min-Entropy Results Comparison.

and ECC [11]) used in hybrid solutions [30]. The comparison is conducted given two different NIST security strength (80 and 112 bits) requirements (please refer back to Section 2.2 for more details). We use the key generation time for two different relative speeds (2 MPH and 20 MPH) collected from our experiments (presented in Section 4.3) as our algorithm's performance. While, for the execution time of RSA and ECC algorithms, we refer to the implementation of these two algorithms on an 8-bit embedded processor (ATmega128) [11] which is widely used in modern wireless sensor networks. The results listed in Figure 5 show that our algorithm can generate/exchange keys 10X faster than the RSA algorithm and 1 to 2X faster than ECC algorithm. Notice that in this comparison, we only compare our algorithm, which solves both the key generate and exchange problems at the same time, to the key exchange process in the hybrid algorithm. A more fair comparison would also consider the key generation time in the hybrid algorithm but the current results clearly demonstrate the advantage of our technique.

3) **Code size overhead comparison:** In order to evaluate the overhead from the memory size point of view, we also compare the code size of our algorithm to sizes of implemented RSA and ECC algorithms. For a fair comparison, we cross-compiled the code of our proposed key generation algorithm to make it suitable for the same 8-bit processor and to get a valid code size. As shown in Figure 5, our algorithm in comparison is 10X smaller than the size of ECC code and is 20X smaller than the size of the RSA code [11].

Table 5: Performance and Code Size Overhead Comparisons on 8-bit Processor.

Security Strength	Performance Overhead (Seconds)				Code Size Overhead (Bytes)		
	RSA [11]	ECC [11]	**Our Alg. (2 MPH)**	**Our Alg. (20 MPH)**	RSA [11]	ECC [11]	**Our Alg.**
80 bits	11.42	1.62	1.725	0.95	6292	3682	331
112 bits	85.2	4.38	2.415	1.33	7736	4812	331

4.5 Discussion

Key generation time: From both the simulation and experiment results, we can see that the key generation time using our proposed key generation algorithm may vary from a few milliseconds to several seconds depending on the automobile speed and key size. However, some differences exist between the experimental and simulation results because the fast-fading and slow-fading models used for simulation cannot precisely model some realistic environments. For example, the results from simulations of different environments

might be the similar but for experiments in different environments, results will tend to not be the same. Another significant reason for this is that our current implementation has a limitation on the sampling rate of the RSSI signal due to the use of applications such as Bluetooth. If the coherence time is smaller than the sampling time (time interval between two RSSI samples), we cannot achieve the ideal key generation time which is computed in our simulations.

Nevertheless, the experimental results demonstrate a proof of concept that a physical layer key generation technique is practical for automotive CPS. For example, the non-safety critical applications such as traffic management generally will have a response time up to few minutes [30], while for safety related applications, the response time requirement may vary from seconds to hundreds of milliseconds. The experimental results have already shown that our technique can fulfill both of these timing requirements. Critical applications with stringent timing requirements (such as collision detection in V2V communication) may require a response time of around 50 milliseconds. However, this type of communication period is typically short and suggests that no large key size is required, which our algorithm can comfortably and quickly compute. Moreover, since the relative speed between automobiles is typically high in real case, our simulation results demonstrate that it is possible to quickly generate keys within a few milliseconds. Compared to our key generation approach, the state-of-the-art hybrid key generation approach is more costly and may require an expensive high-frequency processor or particular hardware accelerator to meet the real-time requirements of the safety critical applications [16].

Correlation of generated keys from different channels: The fundamental assumption of this paper is based on the theory that two wireless channels that are at least a few wavelengths apart are independent of each other. Although this is mostly a theoretical approach [42], researchers are recently performing experiments to prove that the two channels may not necessarily be completely independent [8]. In our work, our simulation results have shown that in automotive wireless communication systems, the correlation between two wireless channels is close to zero due to the high relative speeds of the automotive environment. More importantly, we have experimented with RC cars to demonstrate that in the real world, this assumption is valid for automotive wireless communication even when the two communication channels are close to each other (within 10 meters) and the relative speed between the automobiles is very low (less than 5 MPH). Arguably, an attacker can get around this by attaching devices (less than a few wavelengths) extremely close to the wireless nodes on the automobile in order to receive similar channel properties and information. Nonetheless, for automotive environments this can be tremendously difficult considering the required proximity to the wireless nodes and financial duress to be successful.

5. CONCLUSION

We have presented a physical layer key generation technique that exploits the randomness of the wireless channel to generate secret keys to secure automotive wireless communication using symmetric cryptography. Moreover, our technique solves the challenging key exchange problem in automotive wireless communication with low costs in terms of performance and code size. As demonstrated by our results, the proposed algorithm can generate secret keys with 67% average min-entropy. Furthermore, our proposed technique can achieve up to 10X performance and 20X code size reduction in comparison to the state-of-the-art hybrid cryptographic algorithms. In summary, we propose a simple yet powerful proof of concept for a practical automotive CPS wireless communication-based key generation technique.

6. REFERENCES

[1] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener. Robust key generation from signal envelopes in wireless networks. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 401–410. ACM, 2007.

[2] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for key management-part 1: General (revised. In *NIST special publication*. Citeseer, 2006.

[3] E. Barker and A. Roginsky. Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. *NIST Special Publication*, page 131A, 2011.

[4] M. Bloch, J. Barros, M. R. Rodrigues, and S. W. McLaughlin. Wireless information-theoretic security. *IEEE Transactions on Information Theory*, pages 2515–2534, 2008.

[5] S. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo. Security analysis of a cryptographically-enabled rfid device. In *USENIX Security*, volume 5, pages 1–16, 2005.

[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces. *USENIX Security Symposium*, 2011.

[7] K. Dar, M. Bakhouya, J. Gaber, M. Wack, and P. Lorenz. Wireless communication technologies for its applications. *Communications Magazine, IEEE*, 48(5):156–162, 2010.

[8] M. Edman, A. Kiayias, and B. Yener. On passive inference attacks against physical-layer key extraction? In *Proceedings of the Fourth European Workshop on System Security*, page 8. ACM, 2011.

[9] T. ElBatt, C. Saraydar, M. Ames, and T. Talty. Potential for intra-vehicle wireless automotive sensor networks. In *Sarnoff Symposium, 2006 IEEE*, pages 1–4. IEEE, 2006.

[10] I. ETSI. Intelligent transport systems (its); security; threat, vulnerability and risk analysis (tvra). Technical report, ETSI TR 102 893, European Telecommunications Standards Institute, 2010.

[11] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *Cryptographic hardware and embedded systems-CHES 2004*, pages 119–132. Springer, 2004.

[12] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang. Vehicle-to-vehicle communications: Readiness of v2v technology for application. Technical report, 2014.

[13] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *Advances in*

Cryptology-CRYPTO 2007, pages 553–571. Springer, 2007.

[14] D. E. Holcomb, W. P. Burleson, and K. Fu. Power-up sram state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2009.

[15] 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 *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 321–332. ACM, 2009.

[16] F. Kargl, P. Papadimitratos, L. Buttyan, M. Muter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, et al. Secure vehicular communication systems: implementation, performance, and research challenges. *Communications Magazine, IEEE*, 46(11):110–118, 2008.

[17] J. Lee Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[18] C.-W. Lin, L. Rao, P. Giusto, J. D'Ambrosio, and A. Sangiovanni-Vincentelli. An efficient wire routing and wire sizing algorithm for weight minimization of automotive systems. *Proceedings of the 51st Annual Design Automation Conference (DAC'14)*, pages 1–6, 2014.

[19] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 128–139, 2008.

[20] MathWorks. Matlab, simulink. *www.mathwork.com*, 2014.

[21] C. Miller and C. Valasek. A survey of remote automotive attack surfaces. *Black Hat USA*, 2014.

[22] M. A. Moharrum and A. A. Al-Daraiseh. Toward secure vehicular ad-hoc networks: a survey. *IETE Technical Review*, 29(1):80–89, 2012.

[23] C. W. O'donnell, G. E. Suh, and S. Devadas. Puf-based random number generation. *In MIT CSAIL CSG Technical Memo*, 2004.

[24] 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*, 9(1):17–30, 2010.

[25] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *IEEE Transactions on Mobile Computing*, 5(2):128–143, 2006.

[26] S. N. Premnath, J. Croft, N. Patwari, and S. K. Kasera. Efficient high-rate secret key extraction in wireless sensor networks using collaboration. *ACM Transactions on Sensor Networks (TOSN)*, page 2, 2014.

[27] Y. Qian and N. Moayeri. Design of secure and application-oriented vanets. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2794–2799. IEEE, 2008.

[28] K. Ren, H. Su, and Q. Wang. Secret key generation exploiting channel characteristics in wireless communications. *Wireless Communications, IEEE*, 18(4):6–12, 2011.

[29] M. Rostami, J. B. Wendt, M. Potkonjak, and F. Koushanfar. Quo vadis, puf?: trends and challenges of emerging physical-disorder based security. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition 2014 (DATE'14)*, page 352, 2014.

[30] T. Schütze. Automotive security: Cryptography for car2x communication. In *Embedded World Conference*. Citeseer, 2011.

[31] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann. Car2x communication: securing the last meter-a cost-effective approach for ensuring trust in car2x applications using in-vehicle symmetric cryptography. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5. IEEE, 2011.

[32] M. K. Simon and M.-S. Alouini. Digital communication over fading channels. *John Wiley & Sons*, 2005.

[33] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. *Proceedings of the 44th annual Design Automation Conference (DAC'07)*, pages 9–14, 2007.

[34] R. van den Berg. *Entropy analysis of physical unclonable functions*. PhD thesis, MSc. thesis, Eindhoven University of Technology, 2012.

[35] J. Wan, A. Canedo, A. Faruque, and M. Abdullah. Functional model-based design methodology for automotive cyber-physical systems. *IEEE Systems Journal*, 2014.

[36] Q. Wang, H. Su, K. Ren, and K. Kim. Fast and scalable secret key generation exploiting channel phase randomness in wireless networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 1422–1430. IEEE, 2011.

[37] C. Weiß. V2x communication in europe–from research projects towards standardization and field testing of vehicle communication technology. *Computer Networks*, 55(14):3103–3119, 2011.

[38] D. Work, A. Bayen, and Q. Jacobson. Automotive cyber physical systems in the context of human mobility. In *National Workshop on high-confidence automotive cyber-physical systems*, pages 3–4, 2008.

[39] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam. Information-theoretically secret key generation for fading wireless channels. *IEEE Transactions on Information Forensics and Security*, pages 240–254, 2010.

[40] B. Zan, M. Gruteser, and F. Hu. Key agreement algorithms for vehicular communication networks based on reciprocity and diversity theorems. *IEEE Transactions on Vehicular Technology*, 62(8):4020–4027, 2013.

[41] K. Zeng, D. Wu, A. J. Chan, and P. Mohapatra. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[42] X. Zhou, L. Song, and Y. Zhang. *Physical Layer Security in Wireless Communications*. Crc Press, 2013.