IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 7, 2019, accepted December 27, 2019, date of publication January 28, 2020, date of current version February 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970147

Ephemeral Secrets: Multi-Party Secret Key Acquisition for Secure IEEE 802.11 Mobile Ad Hoc Communication

DAVID SAMUEL BHATTI[1,2] **AND SHAHZAD SALEEM**[1]
[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Department of Computer Science and Information Technology, The University of Lahore, Lahore 54000, Pakistan

Corresponding author: David Samuel Bhatti (david.bhatti@seecs.edu.pk)

This work was supported by the National University of Sciences and Technology (NUST), Islamabad, Pakistan.

ABSTRACT Mobile ad hoc networks consist of wireless nodes and can be established quickly with minimal configuration and cost, because, they do not require any infrastructure in advance. Civil and military applications are using them extensively in emergency and mission-oriented scenarios respectively as multi-party communication systems. Whereas, the multi-party secret key acquisition is one of the acute issues in these low resource wireless ad hoc networks, especially, which are based on IEEE 802.11ah and IEEE 802.11ba (low power WiFi), IEEE 802.15.4(Zigbee), BLE, IEEE 802.15.6 (body-worn or wearable) devices. In this study, a novel low cost and robust approach has been proposed and tested to establish an identical secret key in a multi-user mobile ad hoc environment. We believe, it is one of the groundbreaking contributions toward establishing a cost-effective secret key acquisition solution with respect to memory, computation, and bandwidth. We have used Bloom-Filters to cope with these resource limitations of such wireless setups. The proposed approach has been tested using IEEE 802.11 adapters in a real environment, and we found it to be highly suitable for wireless resource-limited applications.

INDEX TERMS Dynamic secret, ephemeral secrets, shared secret, wireless networks, bloom filter, mobile ad hoc networks.

I. INTRODUCTION

Wireless networks have become an indispensable part of the social, military and healthcare applications [1], [2], and greatly being used in IOTs, emergencies, mobile computing and VANETs at the same time [3]. Novel paradigms in smart devices (mobiles and smartphones) are arising in different dimensions of human life in the form of IOT, wearable, and body-worn devices. These paradigms are assuring improvement in the quality of human life, which is not possible for these devices to achieve alone. WiFi-enabled smartwatches, body-worn cameras, helmets, glasses, and hand carries (bags) are a few examples of such body-worn or wearable [4]. One of the promising mode in which wireless devices can operate is an ad hoc mode in addition to infrastructure mode, in which they can also work as well. WSNs, VANETs, MANETs, and FLANETs are the most practical examples of ad hoc networks [5], [6]. Commonly, they have been observed in the scenarios like, 1) connecting GearVR or Google Day-Dream(virtual reality head wears for audio/video streaming) with mobile or smartphone, 2) only one of the WiFi-enabled device has internet and other WiFi devices are required to share it, 3) group of friends sitting in a cafe or library wants to create temporary hotspot WiFi quickly for data sharing, 4) passengers wants to share information or internet in a train, etc. Similarly, military troops extensively make use of ad hoc networks in fields or when they are on missions, where these networks seize to exist once the mission has been completed. They highly suit the medical emergencies in disaster areas, where rest of the infrastructure has been collapsed(Tsunami, earth quick etc.). One of such motivational scenarios has been shown in Fig.1.

Because, wireless ad hoc networks carry nation integrity-related information in case of military and defence applications; in emergency and disasters they collect and send life critical information, that is why, reliability, security, and throughput is of prime concerns in such situations [7]. For this purpose, IEEE 802.11(WiFi) is one of the most adaptable

The associate editor coordinating the review of this manuscript and approving it for publication was Emrecan Demirors.

FIGURE 1. WiFi adhoc motivation scenario [8].

wireless technologies, which is extensively being used in this context, in addition to others such as Bluetooth, Zibgee, and NFC. Ad hoc mode of networking is very useful as it connects two or more than two WiFi devices, and allow them to send and receive data without using any additional devices like access point (WiFi router). But, due to decentralized and broadcast nature of this technology, information security emerges as a critical issue. Whereas, the most primitive type of concern in information security is confidentiality which assures the prevention of data leakage to an unauthorized person. Cryptographic operations such as encryption/decryption are used to achieve this with the help of shared secret key, which must be highly random and ephemeral in nature. In this study, we have devised a low cost multi-party secret key acquisition solution for limited resource ad hoc networks, assuring randomness and freshness of this shared secret in a timely manner.

Contributions of this study are as follows:

1) Current work is a new contribution in multi-party secret key acquisition because most of the previous proposed schemes were two-party.
2) This acquisition model is interlinked with compact key reconciliation using Bloom filters.
3) It is a low cost solution with respect to time and space, highly suits resource constrained scenarios at the same time.
4) Working of the proposed technique has been tested and verified using real 802.11 WiFi adapters.

Rest of the article has been organized as II. wireless ad hoc networks, III. secret key issues, IV. our goals, V. principles behind SKG, VI. SKG background, VII. related work, VIII. our approach, IX. experimentation and result analysis, X. evaluation, XI. conclusion and future work, and XI. acknowledgment.

II. WIRELESS AD HOC NETWORKS

All the nodes in ad hoc networks can send and receive data to one another, playing the role of relay or router as well. They are self-configured, dynamic, and can be quickly deployed where network infrastructure has been collapsed. For the last couple of years, a continually increasing interest in the acquisition of these networks has been seen in different defence and civil domains due to above mentioned characteristics. But, due to certain unique attributes, for example, dynamically joining and leaving network by the wireless nodes, the design and management of these networks have become seriously challenged as compared to other contemporary and coeval networks. These challenges have motivated research community to play an active role in the innovated advancements and improvements of these networks.

Since, they are distributed type of networks having no centralized control, that is why, they are highly vulnerable to different attacks such as eavesdropping, jamming, MITM, DOS, replay, and resource consumption [9], [10]. We, believe IEEE 802.11 is the most commonly adopted technology in this context, because of its wide provisioning in mobiles, smartphones, tablets, PCs, laptops, wearable and body-worn devices. Moreover, 802.11-based networks have commonly been observed in homes, marts, hospitals, and banks. Thus, it can be concluded, WiFi-enabled devices are easy to integrate with networks in the vicinity. Furthermore, in addition to low transmission power and energy optimization capabilities, WiFi is the only technology which provides higher bit rate and can support multimedia application like live audio/video streaming and monitoring, not possible with other wireless protocols mentioned above [11], [12]. All these reasons motivated us to make use of 802.11 WiFi adapters in our experimentation setups for the proof of concept.

Challenges being faced by ad hoc networks are power, computation, and storage limitations, lacking of centralized control, nodes have to be collaborative inherently, frequent topology change, and security issues due to decentralization and broadcast nature such setups.

III. SECURITY: SECRET KEY ISSUES

A rapid rise in the usage of mobile ad hoc networks requires scientists and researchers to design and develop secure communication protocols for the prevention of data leakage through eavesdropping, masquerading, hacking or ransomware attacks [13]. Moreover, broadcast nature makes ad hoc networks more susceptible to security threats such as eavesdropping, traffic analysis, replay, jamming, spoofing, DOS, DDOS and modifications etc [14], [15]. If we talk about WiFi 802.11 security, among WEP, WPA, WPA2 and WPA3, WEP and WPA has been obsoleted; WPA2 serves only those devices which were approved after 2006. The 4-way handshake protocol used to establish a new session key in WPA2 still not secure and is vulnerable to different attacks such as key re-installation [16], [17]. So, if the secret key happens to be leaked then whole of the communication system become compromised. This shows that acquiring an ephemeral secret key in secure fashion is a basic necessity for these systems. Present secret key establishment schemes are just computationally secure, that is, with sufficient computing resources adversary can crack the secret key, hence they are based on the complexity of large prime numbers [18] and reverse computation problem of discrete logarithm [19]. Unfortunately, key exchange process is also vulnerable to ''Man In The Middle'' attack. These schemes

are computationally heavy such as PKI, that is why, they are not suitable for low resource systems like MANETs, WSNs, wearable and body-worn communication scenarios. So, there is a need to look for means of identical secret key acquisition which an adversary should not be able to breach even it is equipped with unlimited processing capability.

IV. OUR GOAL: COST EFFECTIVE SECRET KEY

In this study, we have focused to establish a shared secret key among the resource considering wireless nodes operating at 2.4GHz. Our work is based on two well established facts, i) all nodes in wireless network can overhear the transmission of any node due to broadcast behavior of the medium. ii) But, the probability of any two nodes to tap exactly the same frames of bits becomes very low, almost approaches to zero [20], [21]. Why these nodes cannot do this is due to erroneous nature of medium and incapability of wireless interface to capture and process all frames flying in the air. It is a natural wireless phenomena occurring at physical and link layer which can be harvested to fix secret key sharing problems in low-resource wireless scenarios discussed in Sec.V and Sec. VI. Furthermore, in our experimentation wireless nodes even within the range of $\lambda/2$ have not been observed to hear exactly the same frames of bits in 2.4GHz wireless environment. A very good example to illustrate above assumptions has been given in [22], that is, its very unlikely for two persons to hear exactly the same parts of one's speech in a noisy room when they are standing at two different locations. Such conceptualization gives birth to implementation scenarios of information-theoretic security an unconditionally secure crypto-system, that is, even having unlimited computing resources adversary cannot break it [23], [24]. Background of secret key generation in context to information-theoretic security perspectives has been discussed in coming Sec. VI.

V. PRINCIPLES BEHIND KEY GENERATION

The broadcast nature of wireless communication has been researched and found ideal for agreeing upon a common secret by transmitter and receiver. This is based upon three wireless communication principles, temporal variation, spatial correlation and channel reciprocity found at wireless physical layer. Reciprocity is responsible for observing symmetric channel path (common randomness) at transmitter and receiver for a very short time known as coherence-time. Temporal and spatial variation causes uncertainty or de-correlation of common randomness, due to which an adversary away by $\lambda/2$ from the receiver will have different observation of same channel path established between transmitter and receiver [14], [25], [26]. At physical layer these principles have been made as a base to propose symmetric peer-peer and group key in different scenarios of wireless networks [20], [22], [27].

These principles also validate data link layer principle that its not possible for the eavesdropper to listen transmission between two nodes correctly or without missing

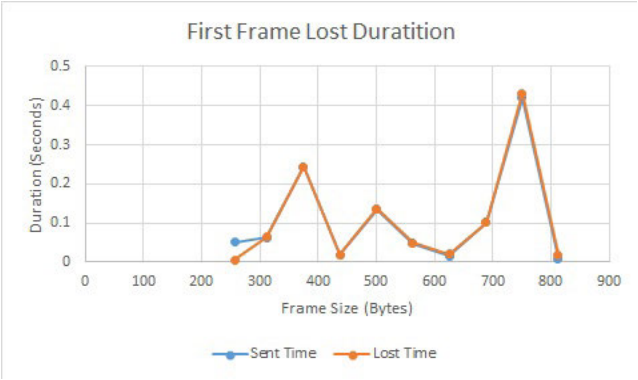

FIGURE 2. First frame lost.

a part of that. It has been observed that an eavesdropper drops its first packet within less than half a second due to quick de-correlation factor or erroneous nature of the wireless medium [20]. We have observed transmissions with different frame sizes, but, first IEEE 802.11 frame has always been lost within 0.5 second as shown in Fig.2.

At this point, a fortune is this that if two legal nodes exchange information of their shared frames without leakage then they can generate a symmetric key based on common frames received correctly by them. Fortunately, there exist probabilistic data structure named **Bloom filters** invented by Burton Bloom in 1970 which can accomplish this [28].

VI. SECRET KEY GENERATION: BACKGROUND

Randomness and uncertainty at wireless physical layer has been used to proof the concept of information-theoretic security in the paradigms of its practicality and implementation [29]–[31]. But, the credit of information-theoretic security back in history goes to Claud E. Shannon, a person who introduced the concept of ''perfectly secure'' system in which the secret key must be equal or larger than the length of a message, and be used only once [32]. In other words, entropy of secret key must be at least equal to the entropy of message being transmitted, that $H(Key) >= H(Message)$, which is also observed in Vernam One-Time-Pad [33]. If a system holds such a condition then according to information theory $I(Message; Cipher) = 0$, means mutual information between plain text and its cipher would be zero. Whereas, **I(.;.)** represents mutual information and **H** Entropy. In the sequel of ''perfectly secure'' system, A. Wyner rules out the assumption of Shannon noiseless channel and assumed a noisy wiretap channel [34]. Wyner's concept of wiretap channel can be attained with the help of additive white-Gaussian noisy channel proposed by Choeng and Hellman [35]. Maurer in his revolutionary work, supported the Shannon's ''perfectly secure'' system model and rejected the Wyner's model of wiretap channel arguing, Wyner's model is impractical in reality [36]. Maurer's model is based on common correlated randomness which can be harnessed to acquire symmetric secret key by the transmitter and receiver. Later, this study was used by the Ahlswede,

Csiszar and Narayan to compute the secret key capacity [37], [38]. The model used by Maurer in [39] and Csiszer in [37], [38] for acquisition of the shared secret is based on two legitimate wireless devices Alice, Bob and an adversary Eve. Alice and Bob observe a common randomness for a very short interval of time due to principal of channel reciprocity [40]. It de-correlates instantly because of temporal and spatial variations [41]. It can be envisioned in information-theoretic perspectives. For instance, If L, M, O are the signal observations made by Alice, Bob and Eve against random source of key material like channel state information, such that, $L = \{l_1, l_2, l_3, \ldots l_n\}$, $M = \{m_1, m_2, m_3, \ldots m_n\}$, and $O = \{o_1, o_2, o_3, \ldots o_n\}$. Then probabilistically, l_i and m_i exhibit a high dependency and correlation provided the channel is error free. But, it de-correlated instantly due to multi-path fading and scattering. Alice and Bob can secure a shared secret key K_i from alike observation l_i and m_i, by exchanging certain set of messages. But, Eve who is separated by a distance of $\lambda/2$ from Alice or Bob will have different observations due to quick de-correlation in channel parameters as a result of spatial and temporal variation. In this scenario l_i, m_i, and o_i are independent and identically distributed (i.i.d) realizations of random variables L, M, and O respectively, whose joint probability distribution is P_{LMO}. In current communication model V, Alice and Bob using functions $SKG_f(L, V)$ and $SKG_f(M, V)$ produce secret keys K_A and K_B over the key range K, respectively. Such a communication system must hold a property, $Pr(K_A = K_B) \geq 1 - \varepsilon$, which assures that two objects Alice(A) and Bob(B) can agree upon a common secret with a very high probability. Here, ε is infinitesimally small, which is not zero but less than any real number [42], [43]. The second property asserted by these systems is $H(K) \geq Log|K| - \varepsilon$, means Key K_i generated in each case is independent of the previous key and is uniformly distributed over the entire key range K. Another property claimed by these system is $H(K) \geq n(R - \varepsilon)$. It means Alice and Bob can generate at least $R - \varepsilon$ secret key bits per observation. Finally, such an information-theoretic system states $I(K; V) \leq \varepsilon$, means Eve knows nothing or too little about key K_i to guess the entire key length. Since, the communication processes at the lower layers (Physical and Link layer) of TCP/IP model are random and correlated, that is why, variables L, M, O used by Alice, Bob and Eave are treated as independent and identically distributed random variables. So, the maximum achievable rate or key capacity can be defined using Eq.(1) given below; whose outcome is such a key which can be used to encipher long messages [44], [45].

$$C_K = R(L; M||O) = min[I(L; O), I(L; M|O)] \quad (1)$$

So, these are the properties which make information-theoretically secure systems hard to break for an adversary who has no bound on computing power.

VII. RELATED WORK

Since, security is an important factor, that is why, a comprehensive related work from the area of ad hoc communication setups has been carried out. One of the earliest attempts to generate identical secret key between two nodes using the physical characteristics of the wireless communication was made by **Jana *et. al.* [25]**. They made use of the **received signal strength (RSS)** to prove common randomness exist at transmitter and receiver which can be used to generate alike key between them. They found that mobility and channel variation can lead to secret bit generation with high entropy. **Ensemble [46]** extends famous **Diffie-Hellman (DH) [19]** secret key exchange protocol and **Amigo [47]** for device authentication. Proximity-based variation in RSS has been leveraged to decide whether the pairing between two wireless devices is authentic or not. Authors used **Voting Systems** to declare an authenticity of pairing process with the help of additional nodes already equipped with pre-shared **Diffie-Hellman (DH) [19]** keys. **ProxiMate [48]** is a sequel of **Amigo [47]** and **Ensemble [46]**, but, with additional feature of extracting shared secret while exploiting the phase of a radio signal instead of RSS, denouncing DH protocol for key exchange at the same time. ProxiMate is computationally simple as its complexity is $O(n)$ compared with **Diffie-Hellman(DH) [19]** whose complexity is $O(n^3)$. Another attempt, to establish a secret key between body-worn devices was made by **Jeff and Tsouri [49]**, which is based on **RSS** measured from the packets being sent forward and backward like data and its ACK using the WBAN AP. WBAN AP sends a poll-request-packet against which the data packets is sent by the WBAN device on the reverse link. Similarly, **Taha and Sivaraman [50]** have harnessed the motion-based variation in RSS as a source of randomness to create a shared secret between two body-worn devices. **ASK-BAN [51]** also exploits the RSS feature of wireless channel for pair-wise and group key acquisition. They used static channels for device authentication and dynamic ones for secret key extraction. For this approach to work effectively, devices must be in line of sight, and they also must be multi-hop relay nodes working in a collaborative fashion. **iARC [52]** utilizes RSS to establish a pair-wise secret key between two WBAN devices. They used channel hopping to introduce the randomness artificially. Similarly, using the same channel parameter RSS, secret key generation for line-of-sight and non line-of-sight has been accomplished by the authors of **[53]** for body-worn wireless sensor devices. **Zhouzhou *et. al.* [54]** have extended the work presented in **[55]**. They used RSSI feature of the radio signal for establishing alike secret key between WBAN device and the control unit with the help of nodes which are already paired with it in a secure fashion. **GPAKE [56]** is a combination for two protocols, one for pairwise and other for group key acquisition. It is a generalized protocols which can be configured to make use any of the wireless signal characteristics such as amplitude, phase, strength of the radio signal transmitted in Coherence-Time. This protocol assumed the nodes to be arranged in a ring fashion which is not feasible in all scenarios. Secret key without using cryptography was accomplished by ***Safaka et. al. [22]*** between two or multiple nodes of 802.11 standard(2.4GHz). They

exploited the communication properties of data link layer for this purpose, where a large number of packet exchange and re-transmissions has been used to obtain sufficient number of identical packets at different nodes. This idea is software based, and brings no change in the existing architecture of the hardware. But, this approach severely suffers from a lot of packet exchange, re-transmission and storage issues, which are making it bandwidth and space hungry. In addition to this, linear combination and the use of error correcting codes demands a handsome amount of computing resources; this means it is less suitable for low resource scenarios. Secret key acquisition based on the similar principles as in [22] has been evaluated and proposed in [27], which is basically a slight variation of secret key generation technique presented in [57]. Creating secure key from wireless erasure environment was presented by Katrina *et. al.* [58]. In this technique, they used beam-forming for creating more noisy environment; further they used wiretap coding to transmit shared secret from one node to another node. They proposed that if the secret is made to be transmitted with SNR less than the receiving SNR of the adversary but greater than that of legal node, then, that transmission will be decoded by legal, but, not by the adversary. This scheme demands to have the prior knowledge of the attacker SNR and provision of directional antenna on each wireless node in the network.

VIII. OUR APPROACH FOR SECRET KEY SHARING

Before discussing the steps of secret key generation in this study, its better to discuss Bloom filter, on which, our proposed approach is based upon. Bloom filter is a space-efficient probabilistic data structure of type bit array. It is used to check the membership of an element in the set. A large number of elements can be represented in a very small space of Bloom filter. That is why, they have great applications in the area of computer science, software engineering, databases, network communication and information security [59], [60]. Due to space/time efficiency, they are highly suitable to carry a large number of frames information from one entity to another over a low resource networks such as WSN, WBAN, WiFi, Zigbee, Bluetooth, and NFC [61].

Working of Bloom filter is very simple; Bloom filter of size **m** is initialized with **0**. Then, an element **d** from input set **S** is passed to **k** different hash functions. Their output is the index values of the Bloom filter from **0** to **(m-1)**; these indices are set to **1**. All other elements are inserted in the same way. When the membership of an element need to be checked, it is passed to the same set of hash functions, and if all the resultant indices are **1**, the element is present, otherwise not. Bloom filter never generate **False-Negative**, that is, it never reply ''Does Not Exist'' to a membership query operation when an element $d \in S$. But, **False-Positive** can be observed, that is, it can reply to membership query operation Element Does Exist, even if an element $d \notin S$. So, the price to pay for using Bloom filter is the **false positive rate(FPR)** which grows with the rate of filling insertions. **false positive rate (FPR)** can be optimized by keeping reasonable size of Bloom filter and the number of hash functions. Moreover, by using different collision resistant hashing techniques, **FPR** can be minimized further.

An interesting feature of Bloom filter is this that there is a well-defined trade-off between size **m** of Bloom filter (space and bandwidth factor), the number of hash function **k**(computation factor), and FPR (error probability) [61]. **k** and **m** can be configured in such way that false positive rate fall within the tolerable limits. If we know the size of input **n** and error probability threshold, then the size **m** and the number of hashing functions **k** can be determined. Obviously, $P_{B[i]\neq 1}$ the probability of not setting a certain location after the addition of **n** elements) is $(1-1/m)^{kn} \approx e^{-kn/m}$. This approximation has been made from well-known formula for calculating e, that is, $e = \lim_{x\to\infty}(1-1/x)^{-x}$. Similarly, $P_{B[i]=1}$ the probability of setting certain location to 1 would be $1-(1-1/m)^{kn} \approx 1-e^{-kn/m}$. Provided the hash functions are independent, random, and uniformly distributed over the entire space of Bloom filter **B**, then the probability of false-positive can be determined using Eq. (2)

$$P_{FP} = \left(1-(1-1/m)^{kn}\right)^k \approx \left(1-e^{-kn/m}\right)^k \tag{2}$$

It can be seen that the values of k and m can lessen **FPR** to a great extent [62]. Thus, knowing the size of input **n** and P_{FP}, the number of hash functions k and size of bloom the filter **m** can be estimated using Eq.(3) and Eq.4 respectively.

$$k = \log 2 \times m/n \tag{3}$$

$$m = -n \log P_{FP}/(\log 2)^2 \tag{4}$$

The derivation procedure of these equations is beyond the scope of this article, but, can be reviewed at [59], [63]. These equations give approximate values, for desired accuracy, we highly recommend that user must configure the values of **m** and **k** against a certain value of error probability instead of fully relying on theoretical values of **m** and **k** derived from these equations. Reason for such recommendation has been discussed in sec. X-B2.

A. STEP1: WIRELESS TAPPING/SNIFFING

Sniffing is the technique in which wireless nodes can capture network packets not even destined to them. Wireless nodes can be set in three modes, network, promiscuous and monitor mode. Network mode is a normal communication mode where nodes can communicate with one another nodes in the network. Promiscuous mode allows the wireless interface to capture packets belonging only to its own network. The packets being captured in this mode have pseudo MAC header attached to it instead of original one. Moreover, these packets do not contain wireless MAC and radio information. Whereas, packets captured in monitor mode renders whole frame along with wireless MAC and radio information.The monitor mode is highly suitable for the analysis of layer2, which include data, ACKs, control and management frames

TABLE 1. Command and tools.

Sr.No	Command and Tools	Purpose
1	pkill Network Mananger	stops any process which hinders in setting the WiFi Adapter in Monitor Mode
2	service network-manager start	restore network services stopped due to setting the device in monitor mode
3	ifconfig	list the network interfaces
4	ifconfig wlanx down	shuts down the interface wlanx where x could be 0,1,2,3
5	iw dev	lists the physical interfaces attached with WiFi adapters
6	iw phy phy0 info	gives a lot of information about whether the physical interface phy0 supports monitor mode or not
7	iplink set wlanx down	shuts down wlanx
8	iwdev wlanx set type monitor	put the wlanx device in monitor mode
7	iplink set wlanx up	start wlanx for capturing, if was set monitor mode
8	iw dev wlanx channel y	set the channel number y on wlanx
9	iw dev wlanx link y	gives link information such as bit rate, ssignal strength, frequency etc
10	ip link show	show the current MTU size
11	ifconfig wlanx mtu y	set the current MTU size
12	iwlist scan	scans for WiFi channels in th environment
13	tcpdump	captures traffic
14	scapy, hashlib, Tskidmarks, SciPy	python libraries for raw frames send/receive, hash functions, and statistical testing
15	python -m cProfile program.py	finds execution time of script program.py

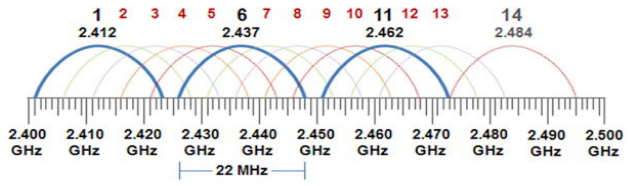

FIGURE 3. 802.11 2.4GHz channels overlapping.

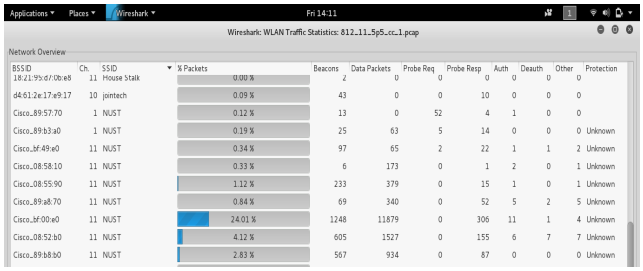

FIGURE 4. Frames from overlapping channels.

transmission from the specified channel. We have used ***tcpdump*** and ***Wireshark*** for 802.11 frames capture and analysis. From our practical and hands-on-working, we have found, Kali Linux a highly suitable operating system platform to carry out such sort of research activities because it is equipped with all required software, commands and tools. Windows operating systems provides no support to work in monitor mode with the exclusion of specialized paid software named ***"Acrylic WiFi Software"*** [64].

For the current study, we set nodes in monitor mode to sniff IEEE 802.11 frames from the wireless MAC layer for the purpose of multi-node secret key generation. For sniffing, nodes in the monitor have been observed with well-known non-overlapping channel 1, 6, and 11 as shown in Fig.3. Anomaly has been detected too, that is, node in the monitor mode also sniffs some packets from the adjacent or other channels due to channel overlapping [65]–[67]. During capturing frames from specified channel 11, some of the frames from 10 and 1 have also been observed to be captured by 802.11 nodes. The statistics of such as capture has been shown in Fig. 4. It is a Wireshark screenshot of statistics about WLAN traffic captured by node "CC" in Test1, which is discussed in Sec. IX-A. In addition to 98.68% frames from channel 11, about 0.31% and 0.33% frames from channel 1 and 10 have also been sniffed by this node. It is necessary to mention, the values shown in Fig. 4 are just the fractions of these percentages.

The set of linux commands used to carry out this research have been given in the Table. 1.

B. STEP2: POPULATING BLOOM FILTER

Once, the wireless nodes have been set in a monitor mode, they start capturing 802.11/2.4GHz frames of the specified channel from its surrounding, and save them in one of the packet capturing format, such as, .pcap or .pcapng. We have used .pcap file format for packet capturing because of its simplicity. When wireless nodes complete capturing to a certain level, then they populate their respective Bloom filters. The nodes $N_1, N_2, N_3, \ldots N_n$ populate their Bloom filters $\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n$ respectively using one of the group function SHA, MD5, and Murmer. We used Murmer hash functions in this study because they are fast to compute and generate relatively good hash codes [68]. Murmer with seed values ranging between 41 and 47 were selected to be used. Once, this step has been completed, reliably, these nodes exchange their Bloom filters with one another using feed back mechanism. Let, $f_1, f_2, f_3 \ldots f_n$ are 802.11/2.4GHz frames which a wireless nodes has sniffed, then they will be inserted in Bloom filter α of size $\mathbf{m}$ with $\mathbf{k}$ hash functions $H = \{h_1, h_2, h_3 \ldots h_k\}$ using Algo.1. It is a frames insertion algorithm, where the size of Bloom filter, number of hash functions and their identity is kept alike at all nodes.

C. STEP3: DETERMINING IDENTICAL FRAMES

Upon receiving the Bloom filters from other nodes successfully, each node determines the commonality of Bloom filters $\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n$ at their own sites. The Algo.2 has been used

Algorithm 1 Insert Frames Into Bloom filter

```
for i=1 to m STEP 1 do
    α_x[i] ← 0      //Set Bloom filter Locations to 0
end for
for i=1 to n STEP 1 do
    for j=1 to k STEP 1 do
        α_x[h_j(f_i) mod m] ← 1
    end for
end for
```

Algorithm 2 Computing Similarity

```
for i=1 to m Step 1 do
    if (α_1[i] = α_2[i] = α_3[i], ... = α_n[i] = 1) then
        β[i] ← 1
    else
        β[i] ← 0
    end if
end for
```

Algorithm 3 Extracting SKG Frames

```
index ← 1
for i=1 to n STEP 1 do
    for j=1 to k STEP 1 do
        if β[h_j(f_i)] = 0 then
            Return FALSE
        end if
    end for
    Return TRUE
    SKGFames[index ← index + 1] ← f_i
    index ← index + 1
end for
```

by each node to find the intersection of all the Bloom filters received from others.

D. STEP4: EXTRACTING SET OF COMMON FRAMES

Once the commonality among $\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n$ has been computed, then each of the nodes at own site is responsible for finding the set of common frames using check membership(query) algorithm Algo. 3. It is used to determine whether a particular frame exists in common Bloom filter β or not. The frames which has been mapped common Bloom filter β will be separated as ***SKGFames*** set. This set is probably identical at nodes $N_1, N_2, N_3, \ldots N_n$. The probability of un-alikeness comes from the False Positive Rate(FPR) which is manageable when the Bloom filter of reasonable length is populated with optimum number of hash functions, as discussed in Sec..

E. STEP5: SECRET KEY GENERATION

When the set of common SKG frames has been computed by each node, then they can generate a fixed length symmetric key using any well-known pre-shared non-key cryptographic hash function such as SHA, SHA256, SHA512, BLAKE, and MD5 etc. Eve does not participate in the secret key generation process because it does not share its blooms filter. It only captures the Bloom filters of others, who share the state of their Bloom filters. That is why, nodes other than Eve can generate a secret key, but, Eve cannot do so. This incapability of Eve is a consequence of some missing frames, which were correctly captured by others nodes. Moreover, Eve cannot share its Bloom filter; executing such an action it can be detected easily because it is not the part of a network to whom legal nodes belong. Each node at its site will pass its set of common SKG frames to the pre-shared hash function. This function which will generate a fixed length secret key using a simple process shown in Algo.4. In this algorithm, XORing of computed hashes of SKG frames is carried out to generate a symmetric key at all legal nodes. Reason to choose one of the existing hash function is, they are designed, tested and released after several years effort of world's top cryptographers. So, it is not a good idea trying to design a better hash function within a few days or months.

Algorithm 4 SKG From Common Frames

```
for i=1 to KeyLength STEP 1 do
    SK[i] ← 0
end for
for j=1 to n STEP 1 do
    h_j = H(f_j)
    SK = SK ⊕ h_j
end for
return SK
```

F. STEP6: KEY VALIDATION AND VERIFICATION

Secret key generated by two nodes can be validated through simple challenge-response protocol, HMAC (Hash Based Message Authentication Code) or CRC (Cyclic Redundancy Check). CRC is one of the simplest protocol, which is easy to implement in software and hardware with low processing overhead. That is why, we recommend CRC-based challenge-response protocol for this key validation process using Algo.5. In this algorithm sender sends an encrypted CRC of its own key, receiver decrypts it and sends back encrypted CRC+1. Sender decrypts and matches it with its own CRC+1, if matched, the key is valid, otherwise SKG process is started again.

G. STEP7: ENCRYPTED COMMUNICATION

After it has been verified that the key is valid, then that can be used for actual secure encrypted communication between two or more than two nodes using some encryption protocol such as AES. Moreover, if secret key does not refresh in timely fashion, it can be compromised. We recommend, use of new secret key for a new transmission session strengthens the security of a crypto-system which is hard to break,

Algorithm 5 SK Validation

@Sender
$N_1 \leftarrow f_{CRC}(SK_A)$
$C_1 = E(N1, SK_A)$
$A \xrightarrow{C} B$
@Receiver
$N1^{/} = D(C_1, SK_B)$
if $CRC_B = N1^{/}$ **then**
  $C_2 = E(N1^{/} + 1, SK_B)$
  $A \xleftarrow[send]{C_2} B$
end if
@Sender
$N2 = D(C_2, SK_A)$
if $N2 = N1 + 1$ **then**
  SK is Valid
  Start Secure Communication
else
  SK is Invalid
  Start SKG Process Again
end if

regardless how much computationally powerful an adversary is. Hence, a new key can be acquired quickly from a small number of common frames. This is what, we call ephemeral or dynamic keys, an excellent application of the unconditional or information-theoretic security.

IX. EXPERIMENTATION AND RESULT ANALYSIS

Experimentation has been carried out using 802.11 WiFi devices of alike and different vendors for keeping setup close to real scenarios. Three different types of tests has been conducted, which have been discussed in Sec.IX-A, IX-B and IX-C.

A. TEST1

In order to test how many frames can be overlapped between two nodes, we set up a simple scenario where two nodes were set in the monitor mode to capture the frames of IEEE 802.11/2.4GHz protocol flying the air. Devices labels shown in Fig.5 or subsequent figures of similar scenarios are the last Byte of their MAC addresses. We also deployed an ad hoc network of two nodes which could send and receive a simple file of any format. Nodes capture frames from the specified frequency at which two-node ad hoc network is operating, or the frames might belong to other 802.11 networks using the same frequency for transmission and reception. The nodes has been observed capturing frames of neighboring channels as well due to channel overlapping. But, their ratio compared to specified frequency is very low, moreover, it is not the concern of our discussion in this study. Still, a snap-shot from the Wireshark has been given in Fig.4 to show such an unexpected behavior. It can be seen from the Fig.4 that beside specified channel 11, current node "CC" has also

TABLE 2. .pcap sniffed file in monitored mode: A sample.

Sr. N0	Frame Type	Packet Count		
		Test1	Test2	Test3
1	802.11 Block Ack	15	2364	24
2	802.11 Block Ack Req	96	2068	79
3	Acknowledgement	17131	27522	17853
4	Action	23	99	3
5	Association Request	163	475	207
6	Association Response	2	9	6
7	Authentication	34	50	49
8	Beacon frame	4122	10204	4599
9	CF-End (Control-frame)		1	13
10	Clear-to-send	7526	102308	17300
11	Data	18160	25036	14604
12	Deauthentication	485	1302	556
13	Disassociate	328	924	417
14	Null function (No data)	352	1883	959
15	Power-Save poll		11	326
16	Probe Request	156	490	265
17	Probe Response	4162	1248	2020
18	QoS Data	847	14261	4648
19	QoS Null function (No data)	301	1843	1538
20	Reassociation Request	1	7	10
21	Reassociation Response	4	5	12
22	Request-to-send	791	12544	1861

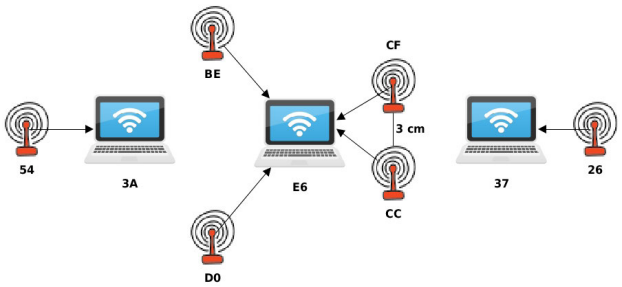

FIGURE 5. Network set-up for test1.

captured some packets from channel 1 and 10. This behavior has already been discussed in Sec.VIII-A

We set up devices in the Graduate Research Lab. of our university; and a 5.5Mb file was sent from device1 (54) to device2 (26). Nodes set in the monitor mode started and stopped manually to capture WiFi frames in the air and store them in .pcap format. We tested Wireshark and tcpdump; it was observed that both perform well in capturing process. But, the results in this article have been extracted from the files, captured using tcpdump. As a sample information, only one of the sniffed .pcap files from each test has been shown in the Table.2 because all other files also contain almost similar amount and type of information. For our current study, all discussion will revolve around the data packets. Experimentation setup used for Test1 has been shown in Fig.5. But, it is worth to mention here that RTS/CTS is disabled by default in wifi ad hoc mode because it does not perform well in this mode [69]. Thus the RTS/CTS frames seen in Table.2 do not belong to our two-node ad hoc set-up rather to other infrastructure based on 802.11 networks.

From Table. 3, it can be seen, there is quite reasonable overlapping of frames between every two devices. This table is showing the intersection of data frames as a sum of One

TABLE 3. Test1:Overlapped frames between 2 devices.

	3A	BE	CC	CF	D0	E6
3A	x	10109	10061	10060	9978	9526
OTF	x	6791	6750	6749	6692	6352
Retry	x	3318	3311	331	3286	3174
BE	x	x	10829	10751	10610	10096
OTF	x	x	7366	7312	7210	6828
Retry	x	x	3463	3439	3400	3268
CC	x	x	x	10693	10529	10025
OTF	x	x	x	7236	7127	6747
Retry	x	x	x	3462	3402	3278
CF	x	x	x	x	10502	9994
OTF	x	x	x	x	7121	6738
Retry	x	x	x	x	3381	3256
D0	x	x	x	x	x	9838
OTF	x	x	x	x	x	6635
Retry	x	x	x	x	x	3203

TABLE 4. Test1:Overlapped frames between 3 devices.

	BECC	BECF	BEDO	BEE6	CCCF	CCDO	CCE6	CFDO	CFE6	D0E6
3A	9959	9944	9888	9413	9898	9840	9526	9828	9375	9304
OTF	6660	6648	6611	6251	6609	6575	6353	6566	6222	6174
OTF	3299	3296	3277	3162	3289	3265	3173	3262	3153	3133
BE	x	x	x	x	10565	10450	9633	10424	9904	9831
OTF	x	x	x	x	7151	7075	6472	7090	6663	6610
Retry	x	x	x	x	3414	3375	3161	3334	3241	3221
CC	x	x	x	x	x	x	x	10353	9834	9747
OTF	x	x	x	x	x	x	x	6980	6585	6529
Retry	x	x	x	x	x	x	x	3373	3249	3218
CF	x	x	x	x	x	x	x	x	x	9737
OTF	x	x	x	x	x	x	x	x	x	6536
Retry	x	x	x	x	x	x	x	x	x	3201

TABLE 5. Test1:Overlapped frames between 4 devices.

	CCCF	CCDO	CCE6	CFDO	CFE6	DOE6
3ABE	9805	9764	9413	9746	9277	9228
OTF	6526	6505	6251	6491	6134	6104
Retry	3279	3259	3162	2922	3143	3124
3ACC	x	x	x	9703	9237	9185
OTF	x	x	x	6450	6103	6074
Retry	x	x	x	3253	3134	3111
3ACF	x	x	x	x	x	9183
OTF	x	x	x	x	x	6072
Retry	x	x	x	x	x	3111

Time Frames(OTF) and re-transmitted(Retry). The same convention has been followed in all three cases for representing intersection. Overlapping shown in the table includes frames of two-node ad hoc network and others operating at the same channel. Overlapping of frames excluding those belong to our own two-node network eliminates the requirement of data transmission and setup of this two-node network as well. It also causes, reduction in the overhead of bandwidth at the same time. Furthermore, if networks nodes are required to communicate in group fashion then we need to check intersection between more than two devices to observe the behavior of proposed scheme. That is why, we calculated overlapping of frames among 3, 4 and 5 nodes as shown in the Table.4, Table.5 and Table. 6 respectively. Among 5 devices frames have been overlapped up to a reasonable extent, such that, generating alike key from this information for these devices is highly possible. The overlapping of OTF between 6 nodes was found about 6000 frames.

TABLE 6. Test1:Overlapped frames between 5 devices.

	3ACCCFD0	3ACCCFE6	3ACCD0E6	3ACFD0E6
BE	9630	9156	9120	9113
OTF	6391	6030	6013	6007
Retry	3239	3126	3107	3106
CC	x	x	x	9073
OTF	x	x	x	5979
Retry	x	x	x	3094

Test1:Overlapped Frames Between 6 Devices

3ACCCFD0E6	
BE	9011
OTF	5921
Retry	3090

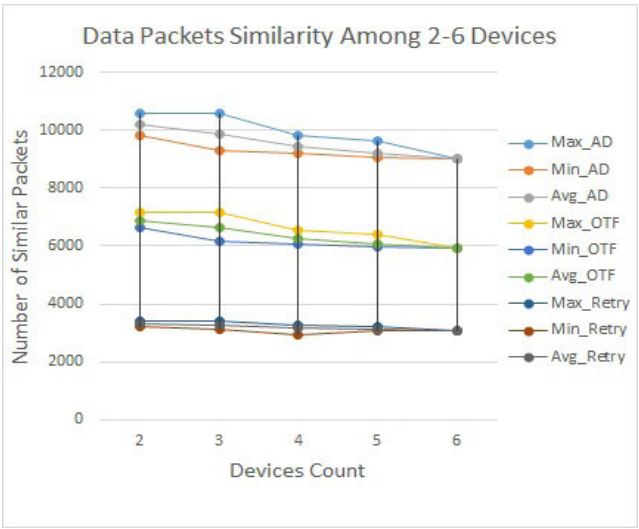

FIGURE 6. Test1: Identical data frames.

We determined, maximum, minimum and average of intersection against 2,3,4,5 and 6 nodes. Fig. 6 clearly shows that there is about 6000 to 7000 alike frames between 2 to 6 nodes, which concludes that this behavior is a better candidate for generating a shared secret in wireless networks.

B. TEST2

In Test2, we want to see behavior of more number of nodes in a grouped fashion. So, we set up two groups, each of five nodes as shown in the Fig.7.

The purpose of this experimentation was to see the applicability of proposed model for secret key establishment in scenarios where the wireless nodes are communicating in a grouped fashion such as cooperative nodes [70], non-chargeable or re-chargeable groups [71]. The purpose of referring these two articles is just to show, that, sometimes we need to establish secret key between group of nodes, such as military troupes, group of devices located and communicating on parts of the same body etc. We had a Graduate Research Lab. arrangement in such a way, that we have to put the nodes in a setup as shown in the Fig. 7. In this setup, nodes of group1 were slightly close to two-node ad hoc sender and receiver compared to the nodes od group2. It can be seen from

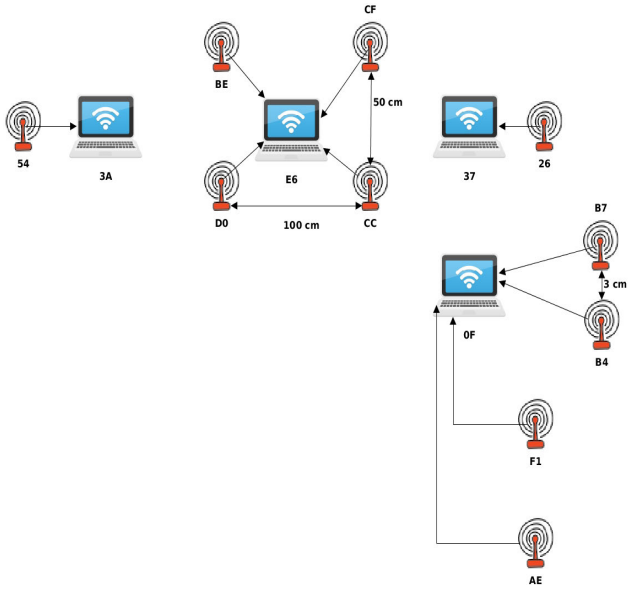

FIGURE 7. Network set-up for test2.

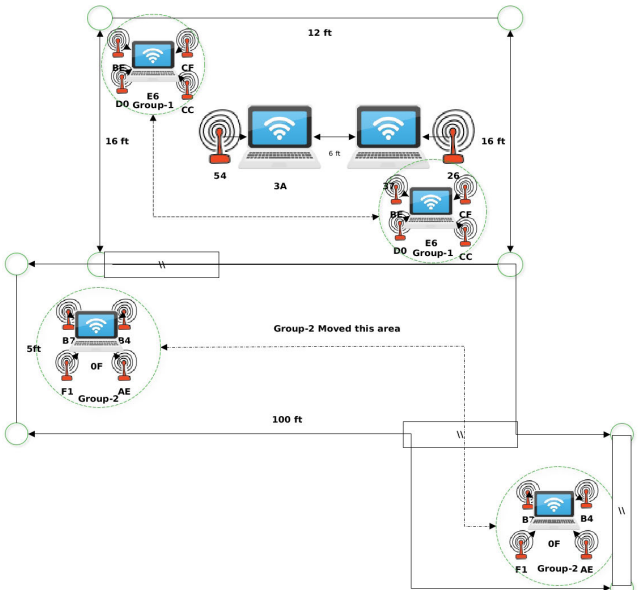

FIGURE 8. Network set-up for test3.

TABLE 7. Test2:Overlapped frames between 2 devices.

	Group1						Group2				
	{BE	CC	CF	D0	E6}		{AE	B4	B7	F1	ZF}
BE	x	8772	8688	8574	8626	AE	x	8953	9160	9038	8661
OTF	x	4625	4558	4445	4512	OTF	x	4804	4959	4859	4722
Retry	x	4147	4130	4129	4114	Retry	x	4149	4201	4179	3939
CC	x	x	9294	9186	9233	B4	x	x	8745	8571	8282
OTF	x	x	5069	4963	5028	OTF	x	x	4600	4447	4380
Retry	x	x	4225	4223	4205	Retry	x	x	4145	4124	3902
CF	x	x	x	9036	9171	B7	x	x	x	8906	8593
OTF	x	x	x	4836	4969	OTF	x	x	x	4722	4634
Retry	x	x	x	4200	4202	Retry	x	x	x	4184	3959
D0	x	x	x	x	8930	F1	x	x	x	x	8309
OTF	x	x	x	x	4764	OTF	x	x	x	x	4382
Retry	x	x	x	x	4166	Retry	x	x	x	x	3927

the Table.7, that there is a reasonable frames overlapping between each 2 nodes belonging to group1 and every 2 nodes of group2. Minimum intersection between every 2, 3, 4 and 5 nodes of group1 is 4445, 4125, 3944, 3837 and in group2, the values are 4380, 4107, 3723 and 3637 respectively for the same number of nodes. The results of intersection for 2, 3, 4 and 5 nodes have been drawn from Table.7, Table.8, Table.9, and Table.10 respectively. One of the good news is this, that there is a reasonable number of alike frames captured by two groups, that is about 3005 OTF frames. These results affirm that a secret key can be established between two groups of nodes operating at two different locations.

C. TEST3

The purpose of Test3 is to determine the overlapped frames when the nodes are moving, because there are different mobility models, grouped-mobility is one of them [72]. We took same 10 capturing devices of Test2 and divided them into two groups, group1 and group2. We moved one group of nodes close to two-node ad hoc network within the area of about 16ftx12ft and other group of nodes in the area of about 100ftx20ft in Graduate Research Lab. of our university. This typical setup has been shown in Fig.8.

We have observed that distance from the source and mobility of nodes create more uncertainty in capturing process, also affects frames overlapping to a considerable extent. But, it was still in the figure of 1000 frames, which is still a good number of common frames for the generation of symmetric secret key between two mobile groups. Minimum intersection between every 2 and 3 nodes of group1 is 3378, and 3201, whereas, between 4 and 5 is 3110. Similarly, in case of group2, the intersection between every 2 and 3 nodes is 2130, 1895 and every 4 and 5 is 1858, as shown in Table.11, Table.12, Table.13 and Table.14 respectively. This model is fully applicable where nodes are moving in group/groups because about 1242 alike frames exist between two groups as shown in Table.14.

X. EVALUATION

There are different parameters which are being used in literature to check whether the proposed secret key generation procedure is worth to be adopted or not. The main purpose of this study is not to reject the existing cryptographic technologies for secret key acquisition, but, just to contribute in devising low cost, salable, and robust secret key acquisition solution for limited resource wireless scenarios. We have selected some parameters such as Randomness, Key Generation Rate (KGR), Key Mismatch Rate (KDR), Key Refresh Rate (KRR), compatibility, and Cost (in terms of bandwidth, Power Consumption, processing, extra hardware) to evaluate proposed scheme.

A. RANDOMNESS

Encryption key must contain substantial amount of randomness, that is, all bits must be uniformly distributed over the entire key length. People uses NIST software suit for testing

TABLE 8. Test2:Overlapped frames between 3 devices.

	Group1							Group2					
	CCCF	CCDO	CCE6	CFDO	CFE6	D0E6		B4B7	B4F1	B4ZF	B7F1	B7ZF	F1ZF
BE	8444	8339	8347	8246	8296	8155	AE	8571	8438	8112	8608	8277	8101
OTF	4366	4271	4299	4198	4255	4125	OTF	4512	4399	4300	4521	4419	4283
Retry	4078	4068	4048	4048	4041	4030	Retry	4059	4039	3812	4087	3858	3818
CC	x	x	x	8800	8846	8705	B4	x	x	x	8212	9725	7748
OTF	x	x	x	4666	4719	4592	OTF	x	x	x	4178	4107	3962
Retry	x	x	x	4134	4127	4113	Retry	x	x	x	4034	5618	3786
CF	x	x	x	x	x	8616	B7	x	x	x	x	x	8011
OTF	x	x	x	x	x	4515	OTF	x	x	x	x	x	4172
Retry	x	x	x	x	x	4101	Retry	x	x	x	x	x	3839

TABLE 9. Test2:Overlapped frames between 4 devices.

	Group1				Group2		
	{CFD0	CFE6	D0E6}		{B7F1	B7ZF	F1ZF}
BECC	8073	8095	7963	AEB4	8131	7824	7669
OTF	4068	4101	3987	OTF	4170	4079	3959
Retry	4005	3994	3976	Retry	3961	3745	3710
BECF	x	x	7907	AEB7	x	x	7422
OTF	x	x	3944	OTF	x	x	3723
Retry	x	x	3963	Retry	x	x	3699
CCCF	x	x	8413	B4B7	x	x	7476
OTF	x	x	4371	OTF	x	x	3760
Retry	x	x	4042	Retry	x	x	3716

TABLE 10. Test2:Overlapped frames between 5 devices.

Group1 CCD0E6		Group2 B4B7F1ZF	
BECF	7762	AE	7434
OTF	3837	OTF	3637
Retry	3925	Retry	3651
Group1 BECCCFD0E6		Group2 AEB4B7F1ZF	
All Data		6413	
OTF		3005	
Retry		3408	

TABLE 11. Test3:Overlapped frames between 2 devices.

	Group1						Group2				
	BE	CC	CF	D0	E6		AE	B4	B7	F1	ZF
BE	x	6434	6514	6215	6434	AE	x	3642	4732	5085	3642
OTF	x	3481	3554	3378	3481	OTF	x	2130	2655	2930	2130
Retry	x	2953	2960	2837	2953	Retry	x	1512	2077	2155	1512
CC	x	x	8620	7950	9120	B4	x	x	3783	4233	4386
OTF	x	x	5337	4877	5682	OTF	x	x	2240	2589	2671
Retry	x	x	3283	3073	3438	Retry	x	x	1543	1644	1715
CF	x	x	x	7953	8628	B7	x	x	x	5465	3788
OTF	x	x	x	4906	5340	OTF	x	x	x	3163	2228
Retry	x	x	x	3047	3288	Retry	x	x	x	2302	1560
D0	x	x	x	x	7653	F1	x	x	x	x	4551
OTF	x	x	x	x	4622	OTF	x	x	x	x	2849
Retry	x	x	x	x	3031	Retry	x	x	x	x	1702

randomness or its guidelines to measure this parameter [73]. But, NIST software suit requires a very large string of 0s and 1s, sometimes it crashes as well. That is why, we used Python libraries **Tskidmarks** [74] and **SciPy** [75] to test the randomness of shared secret generated through procedure already discussed in VIII-E. These libraries are being used extensively in scientific and research industry for statistical tests.

The most popular tests used to test randomness of a binary string are Runs-Test, $\chi^2 - test$, Shannon average entropy and correlation. In this evaluation, for simplicity **Hash** is the resultant hash value of five common frames (from Test3) which are XORed and hashed using SHA512. The binary output of $Hash = SHA512(5e173fc0456b09ef53f2872ff131e1a8a0c252ab1e16afe5\ldots)\oplus Hash(1b784e94b9f12f5615b08d8ab4b8ddc750f999d4e3ef2427\ldots)\oplus Hash(b477046129fbc9cbf72179aa65061b6b3193907bd4dc60d3\ldots)\oplus Hash(6117dbd184532a952bfebf5581fd4ca96fd4780ea5a3ffe8\ldots)\oplus Hash(309e5bab7ee2252f69051f84674b8677a2395288579ddd7e\ldots)\oplus Hash(5e173fc0456b09ef53f2872ff131e1a8a0c252ab1e16afe5\ldots)$ is 0b1000101010001100100110000011110010 0011101010010111000110111100010001110011 0100011011111101100110000110100011100000010 1000011100101001011000001111000000110111 0000110010110110011000000011000110001001 0101100111101011010010100011101100100 1010100011001111110000100000001100010001 0110001110111101001111101001010011011111 1 1100011001010101001100001100001100011100 1 0001101011111000111110011101010100001 1011111111111 0111011101110010010110000000 1 0000010001001010101 0100100110101000000000 0001011000100000011011011 0100. The randomness of this output has been tested using Runs, χ_2, entropy and correlation tests.

Runs test is one of the most practiced test in science and engineering to testify the randomness of information. For long string of binary symbols 0s and 1s, the runs test formulates two hypothesis. H_0: The given binary string is random; H_1: The given binary string is not random. Furthermore, in science and engineering significance level α is taken as 0.05 for statistical test. The statistics of Runs test [76] are

$$Z = (R - \mu)/\sigma \tag{5}$$

R is observed number of runs, μ is the expected number of runs and σ is the standard deviation which is square root of the variance, σ^2. μ, σ^2 can be computed using 6 and 7.

$$\mu = ((2N_0N_1)/N) + 1 \tag{6}$$

Where, $N = N_0 + N_1$ and N_0, N_1 are number of 0s and 1s in the binary string respectively.

$$\sigma^2 = 2N_0N_1(2N_0N_1 - N)/N^2(N - 1) \tag{7}$$

TABLE 12. Test3:Overlapped frames between 3 devices.

	Group1							Group2					
	{CCCF	CCDO	CCE6	CFDO	CFE6	D0E6}		{B4B7	B4F1	B4ZF	B7F1	B7ZF	F1ZF}
BE	6244	5965	6439	6009	6244	5965	AE	3314	3558	3642	4479	3314	3558
OTF	3363	3201	3381	3241	3360	3201	OTF	1895	2077	2130	2526	1895	2077
Retry	2881	2764	3058	2768	2884	2764	Retry	1419	1481	1512	1953	1419	1481
CC	x	x	x	7638	8620	7946	B4	x	x	x	3696	3783	4233
OTF	x	x	x	4670	5337	4877	OTF	x	x	x	2187	2240	2580
Retry	x	x	x	2968	3283	3069	Retry	x	x	x	1509	1543	1653
CF	x	x	x	x	x	7669	B7	x	x	x	x	x	3704
OTF	x	x	x	x	x	4692	OTF	x	x	x	x	x	2177
Retry	x	x	x	x	x	2977	Retry	x	x	x	x	x	1527

TABLE 13. Test3:Overlapped frames between 4 devices.

	Group1				Group2		
	{CFD0	CFE6	D0E6}		{B7F1	B7ZF	F1ZF}
BECC	5818	6244	5965	AEB4	3255	3314	3558
OTF	3110	3363	3201	OTF	1858	1895	2077
Retry	2708	2881	2764	Retry	1397	1419	1481
BECF	x	x	5818	AEB7	x	x	3255
OTF	x	x	3110	OTF	x	x	1858
Retry	x	x	2708	Retry	x	x	1397
CCCF	x	x	7638	B4B7	x	x	3696
OTF	x	x	4670	OTF	x	x	2187
Retry	x	x	2968	Retry	x	x	1509

TABLE 14. Test3:Overlapped frames between 5 devices.

Group1 CCD0E6		Group2 B4B7F1ZF	
BECF	5818	AE	3255
OTF	3110	OTF	1858
Retry	2708	Retry	1397
Group1 BECCCFD0E6		**Group2 AEB4B7F1ZF**	
ALL Data		2493	
OTF		1242	
Retry		1251	

When, we subjected the output of SHA512 to the Runs Test, we found $R = 254, N_0 = 239, N_1 = 273, \mu = 255.871, \sigma^2 = 11.252$, $z = -0.1666$, $p = 0.238$. Since the $Abs|Z| < Z_{1-\alpha/2}$, that is why, there is no reason to reject the null hypothesis, means the binary string has been generated through a random process [76].

Chi-Square Test is another used to testify the randomness using statistics given in Eq.(8) [77], [78].

$$\chi^2 = \sum_{k=1}^{n} (O_k - E_k)^2 / E_k \tag{8}$$

Passing, the output of SHA512 to χ^2 *Test*, we found $DOF = 1$, $O_k 239, 273$, $E_k 256, 256$, $\chi^2 = 2.25$. Since $P_{\alpha=0.10} < (\chi^2 = 2.25) \leq P_{\alpha=0.05}$, that is why we have no reason to reject null hypothesis. Thus, the output is truly a random sequences of bits [79].

Entropy is the measure of an uncertainty, thus more the bits of shared secret are random in nature more will be the uncertainty in guessing this shared secret. Entropy can be calculated using Eq.(9)

$$H(x) = \sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{9}$$

Entropy of the given bit sequence has been found to be 0.7 which means it is quite uncertain, or random.

Similarly, binary string with low auto-correlation indicates more randomness. We found a very low auto-correlation that is 0.005 of a given string, almost close to 0. At a confidence level $\alpha = 0.05$, the obtained value of auto-correlation indicates that the bit sequence is highly random in nature.

B. OVERHEADS

1) TRANSMISSION

Secret key can be acquired even without the setup of 2-node ad hoc sender and receiver. It means we do not need to make transmission for secret key generation. The nodes in monitor-mode can capture free frames flying in the air, whose commonality can be processed and can be used to acquire a shared secret key later. We randomly selected devices of Test1, Test2 and Test3 to determine the intersection of frames captured in monitor mode. We have selected nodes randomly and considered only those frames whose type was 'Data', excluding the frames of two-node network. Due to space limitation, we have represented $(CC \cap CF)$, $(D0 \cap E6)$, $(CC \cap CF \cap BE)$, $(CC \cap CF \cap D0 \cap E6)$, $(CC \cap CF \cap D0 \cap E6 \cap BE)$ by a, b, c, d, and e; $(B4 \cap B7)$, $(AE \cap ZF)$, $(B4 \cap B7 \cap F1)$, $(B4 \cap B7 \cap AE \cap ZF)$, $(B4 \cap B7 \cap AE \cap ZF \cap F1)$ by f, g, h, i and j respectively. The number of identical frames found were 367, 272, 221 in Test1, Test2 and Test3 respectively as shown in Table.15. These results clearly claim that we do not need to make any transmission for obtaining frames to use them as a key material, rather, free frames are available in the air for this purpose.

In addition to this, Bloom filters are small in size, that is why, their transmission requires minute bandwidth at the same time.

2) OPTIMIZATION OF THE BLOOM FILTER

Number of input frames **n**, size of Bloom filter **m** and number of hash functions **k** can be configured according to underlying wireless nodes resources. This can be seen from the Table16 that for certain specific size of overlapped frames(**n**) and

TABLE 15. Frames overlapping excluding own transmissions.

Test1									
a		b		c		d		e	
636		942		550		494		453	
$CC \cap CF \cap D0 \cap E6 \cap BE \cap 3A = 367$									
Test2									
	group1						group2		
{a	b	c	d	e}	{f	g	h	i	j}
2224	2807	1304	872	573	2844	3610	1789	1374	1066
$Group1 \cap Group2 = 272$									
Test3									
	group1						group2		
{a	b	c	d	e}	{f	g	h	i	j}
1956	1620	1241	1141	884	1629	1477	691	926	451
$Group1 \cap Group2 = 221$									

TABLE 16. Determining M and K for Bloom filter.

n	p	m	k
50000	$\{1^{-10}, 1^{-5}, 1^{-4}, 1^{-3}\}$	$\{293, 146, 117, 87\}$	$\{33, 17, 13, 10\}$
40000	$\{1^{-10}, 1^{-5}, 1^{-4}, 1^{-3}\}$	$\{235, 117, 94, 71\}$	$\{33, 17, 13, 10\}$
30000	$\{1^{-10}, 1^{-5}, 1^{-4}, 1^{-3}\}$	$\{175, 87, 71, 52\}$	$\{33, 17, 13, 10\}$
20000	$\{1^{-10}, 1^{-5}, 1^{-4}, 1^{-3}\}$	$\{117, 58, 47, 36\}$	$\{33, 17, 13, 10\}$
10000	$\{1^{-10}, 1^{-5}, 1^{-4}, 1^{-3}\}$	$\{58, 29, 23, 17\}$	$\{33, 17, 13, 10\}$
1000	$\{1^{-10}, 1^{-5}, 1^{-4}, 1^{-3}\}$	$\{6, 3, 2.5, 2\}$	$\{33, 17, 13, 10\}$

some acceptable value of **FPR** the size of Bloom filter(**m**) and the optimum number of hash functions(**k**) can be decided using equations 4 and 3. There are different online tools [80] to find the values of these variables. We have verified their results using a spread sheet software Open Office Calc installed on Linux Platform. Number of hash functions **k** grows linearly with bits per item **m/n**. Increasing the number of hash functions would incur more processing overhead. Similarly, increasing the size of Bloom filter would increase the storage overhead. No doubt, these two parameters play a great role in reducing **FPR**. But, at the same time, they must be chosen with great care because they directly affect the resources of wireless nodes. Beside all that, Bloom filters are efficient with respect to limitations of memory, processing and bandwidth. These features determine that small-sized Bloom filter can carry a lot of information over the network.

To verify the theoretical claims, for input size N=3500 frames at the rate of 0.0001false positive, the theoretical values of Bloom filter's size and optimum number of hash functions calculated were 67096 and 13 respectively. We took two .csv files exported from captured frames database of Test3 where the size of one file was 3497 and of other was 2177 frames. We found 0 false positives with m=60000 and k=7, after trying different values of **m** and **k**. The Bloom filters of both files were cross queried, and in both cases, the match found was 1615 data items. It was the number which was obtained when the query was executed in SQL to find common frames between these two files (SQL Tables). It shows, that, its the responsibility of security system designer to find the optimum values of **m** and **k** for system under consideration instead of totally relying on the theoretical claims. These claims may speed up the process of finding optimum values of **m** and **k** for certain value of input size **n** and **FPR**. They cannot provide us exact values because nature of the selected hashing algorithms also affects them.

TABLE 17. Attack analysis using TestCase-3.

Intersection of Legal Nodes	Union of Attacker nodes	Dissimilarity
$b4 \cap b7 \cap ae \cap zf \cap f1x2n$=884	37=3394	884-628=256
	49=4096	884-732=152
	aa=2362	884-496=384
	$37 \cup aa$=5756	884-662=222
	$37 \cup 49$=5337	884-816=70
	$49 \cup aa$=6459	884-799=87
	$37 \cup 49 \cup aa$=7699	884-826=58
$cc \cap cf \cap d0 \cap e6 \cap bex2n$=451	37=3394	451-396=55
	49=4096	451-366=85
	aa=2362	451-340=111
	$37 \cup aa$=5756	451-409=42
	$37 \cup 49$=5337	451-436=15
	$49 \cup aa$=6459	451-425=26
	$37 \cup 49 \cup aa$=7699	451-440=11

C. RELIABILITY AND ROBUSTNESS: ATTACK MODEL

Robustness and reliability of the proposed model has been verified with three attackers deployed in a very close proximity of the group1, but, away from group2. We took two attacker nodes having double the antenna gain as compare to other nodes. We have observed that even three attackers in close cooperation cannot compromise this secret key acquisition process. Critics would say that it could be compromised with more number of attacker, but, we believe that an enemy deploying more number of attackers would increase the probability of their detection and isolation at the same time. Moreover, it is not practically economical for an enemy to deploy number of attacker equal or greater than the legal nodes for stealing the key material. Beside all that, attackers can be defeated easily on the cost of more number of frames captured in the capturing process because it will reduce the probability of tapping exactly the same frames by attackers as that of legal nodes. From analysis, we have come to the conclusion that one, two or three key stealers would never be able to compromise SKG process even having twice the antenna gain of legal node. Results give in Table.17 are showing the number of frames missed by eavesdropper in case of single, two and three attacker nodes. But, beside all that, even single frames missed by eavesdropper is enough to create a shared secret.

D. KGR

KGR depends upon the capturing speed of the devices. In worst case scenario, we have found about 1200 frames overlapped among 10 devices each having capture size of 9-10Mb with capturing duration of about 90 seconds. If we go back to Fig. 2, it has already been verified that a frame becomes lost by an eavesdropper within less than half a second. Whereas, it is a worst case, when a node is closer to two-node ad hoc sender and receiver. So, with distance and movement, time of first frame lost can be shifted toward zero from its mean. On a Haier Laptop model 7G-4H with 1.7GHz Core i3 4th generation processor,

4Gb DDR3 memory and 3 MB cache, the total time to fill Bloom filter, extracting common frames with parameters n(number of input frames)=3500, m(size of Bloom filter)=60000 (occupies 7.5KB on disk), k(number Murmer hash functions)=7, and FPR(false positive rate)=0, and generation of secret key using SHA512 was found to be 0.204 second. If exchange of Bloom filter(m=60000 bits) at minimum speed of 1Mbps takes 0.06 second then with 0.5 second capturing, the approximate time to generate 512 bits of the secret key would be 0=0.764 second.

E. KMR

In our case, evaluation of key mismatch rate is completely different from other proposed approaches. In this study, it is closely related to FPR of the Bloom filter. But, the interesting feature is this, unlike the other techniques, in our approach FPR can be tuned on the cost of minor number memory bytes.

F. KRR

Key Refreshing of secret key is very necessary for the robustness and reliability of the cryptographic system. Key refresh rate (KRR) also determines the probability of key compromise. Since, key generation process is not expensive, that is why, a new key can easily be acquired quickly for new communication session and the old key can be discarded. Such type of secret keys are termed as ephemeral or dynamic keys.

G. COMPATIBILITY

Since, it is software based scheme, that is why, it is fully compatible with existing TCP/IP network stack and does not require any of the layers to be changed.

H. SCALABILITY

Current technique is scalable to any extent of the wireless network density. A new node can acquire a group key, provided, any of the network nodes having group key captures frames along with this new node. Then both exchange their new Bloom filter. These nodes end-up with a alike pairwise key execute proposed scheme. This network node will encrypt its group key with newly acquired pairwise key and sends it to new node which can decrypt it with the same pairwise key, resulting in the acquisition group key.

XI. CONCLUSION AND FUTURE WORK

We are confident to state that the working of proposed secret key acquisition technique has been tested and verified using real IEEE 802.11 wireless adapters. Space efficient data structure make it suitable for low resource wireless devices. Its strength and robustness increases with the density of network. A strong resistant against single node, joint 2 and 3-node attack has been observed. Due to low overheads, proposed solution is highly suitable for ephemeral secrets in resource constrained wireless ad hoc networks.

As for the future work is concerned, we have planned to device a secret key acquisition for heterogeneous IOT devices using different wireless technologies which is one of the great challenge in current research at this time.

ACKNOWLEDGMENT

The authors are extremely grateful to Dr. T. Ali for valuable discussions which helped shape their final solution and for his constructive comments regarding the experimental setup.

REFERENCES

[1] Y. Xiao and Y. Pan, *Emerging Wireless LANs, Wireless PANs, and Wireless MANs: IEEE 802.11, IEEE 802.15, 802.16 Wireless Standard Family*, 1st ed. Hoboken, NJ, USA: Wiley, 2009.

[2] I. AlShourbaji, "An overview of wireless local area network (WLAN)," *Int. J. Comput. Sci. Inf. Secur.*, vol. 11, no. 2, pp. 46–53, Feb. 2013.

[3] P. A. Catherwood, D. Steele, M. Little, S. Mccomb, and J. Mclaughlin, "A community-based iot personalized wireless healthcare solution trial," *IEEE J. Transl. Eng. Health Med.*, vol. 6, pp. 372–380, May 2018.

[4] TredHunter. *WiFi Wearable*. Accessed: Jan. 20, 2019. [Online]. Available: https://www.trendhunter.com/protrends/wifi-wearable

[5] C. Prabha, D. Kumar, and D. Khanna, "Wireless multi-hop ad-hoc networks: A review," *J. Comput. Eng.*, vol. 16, pp. 54–62, Jan. 2014.

[6] A. Guillen-Perez and M.-D. Cano, "Flying ad hoc networks: A new domain for network communications," *Sensors*, vol. 18, no. 10, p. 3571, Oct. 2018.

[7] T. Hayajneh, G. Almashaqbeh, S. Ullah, and A. V. Vasilakos, "A survey of wireless technologies coexistence in WBAN: Analysis and open research issues," *Wireless Netw.*, vol. 20, no. 8, pp. 2165–2199, Nov. 2014.

[8] I. Cardei, Y. Wu, and J. Junco, "Backup Wi-Fi ad-hoc network for emergency response in scenarios with sporadic connectivity and primary users," in *Proc. 10th Int. Conf. Mobile Ad-hoc Sensor Netw.*, Dec. 2014, pp. 66–73.

[9] R. Meddeb, B. Triki, F. Jemili, and O. Korbaa, "A survey of attacks in mobile ad hoc networks," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, May 2017, pp. 1–7.

[10] S. K. Chowdhury and M. Sen, "Attacks and mitigation techniques on mobile ad hoc network—A survey," in *Proc. Int. Conf. Trends Electron. Inform. (ICEI)*, May 2017, pp. 11–18.

[11] D. Bankov, E. Khorov, A. Lyakhov, and E. Stepanova, "IEEE 802.11 ba—Extremely low power Wi-Fi for massive Internet of Things—Challenges, open issues, performance evaluation," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jun. 2019, pp. 1–5.

[12] M. T. Arefin, M. H. Ali, and A. F. Haque, "Wireless body area network: An overview and various applications," *J. Comput. Commun.*, vol. 5, no. 7, pp. 53–64, 2017.

[13] N. Sakovich. (Jun. 2018). *Wearable Technology in Healthcare: How Devices Will Influence Our Health*. Accessed: Oct. 3, 2018. [Online]. Available: https://www.sam-solutions.com/blog/wearable-technology-in-healthcare-ho%w-devices-will-influence-our-health/

[14] J. Zhang, T. Q. Duong, A. Marshall, and R. Woods, "Key generation from wireless channels: A review," *IEEE Access*, vol. 4, pp. 614–626, 2016.

[15] S. Baraković, E. Kurtović, O. Božanović, A. Mirojević, S. Ljevaković, A. Jokić, M. Peranović, and J. B. Husić, "Security issues in wireless networks: An overview," in *Proc. 11th Int. Symp. Telecommun. (BIHTEL)*, Oct. 2016, pp. 1–6.

[16] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2017, pp. 1313–1328.

[17] R. Singh and T. P. Sharma, "On the IEEE 802.11i security: A denial-of-service perspective," *Security Commun. Netw.*, vol. 8, no. 7, pp. 1378–1407, May 2015.

[18] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 26, no. 1, pp. 96–99, Jan. 1983.

[19] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[20] S. Xiao, W. Gong, and D. Towsley, "Secure wireless communication with dynamic secrets," in *Proc. 29th Conf. Inf. Commun. (INFOCOM)*, Piscataway, NJ, USA, 2010, pp. 1568–1576.

[21] L. Czap, V. M. Prabhakaran, C. Fragouli, and S. Diggavi, "Secret message capacity of erasure broadcast channels with feedback," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Paraty, Brazil, Oct. 2011, pp. 65–69.

[22] M. J. Siavoshani, U. Pulleti, E. Atsan, I. Safaka, C. Fragouli, K. J. Argyraki, and S. N. Diggavi, "Exchanging secrets without using cryptography," *CoRR*, vol. abs/1105.4991, 2011. [Online]. Available: http://arxiv.org/abs/1105.4991
[23] S. Liu, Y. Hong, and E. Viterbo, "On measures of information theoretic security," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2014, pp. 309–310.
[24] Y. Liang, H. V. Poor, and S. Shamai (Shitz), "Information theoretic security," *Found. Trends Commun. Inf. Theory*, vol. 5, pp. 355–580, Apr. 2009.
[25] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, New York, NY, USA, 2009, pp. 321–332.
[26] K. Ren, H. Su, and Q. Wang, "Secret key generation exploiting channel characteristics in wireless communications," *IEEE Wireless Commun.*, vol. 18, no. 4, pp. 6–12, Aug. 2011.
[27] I. Safaka, C. Fragouli, K. Argyraki, and S. Diggavi, "Creating shared secrets out of thin air," in *Proc. 11th ACM Workshop Hot Topics Netw. (HotNets)*, New York, NY, USA, 2012, pp. 73–78.
[28] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.
[29] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. Mclaughlin, "Wireless information-theoretic security," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2515–2534, Jun. 2008.
[30] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 240–254, Jun. 2010.
[31] A. Mukherjee, S. A. A. Fakoorian, J. Huang, and A. L. Swindlehurst, "Principles of physical layer security in multiuser wireless networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 6, vol. 3, pp. 1550–1573, Feb. 2014.
[32] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, Oct. 1949.
[33] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *Trans. Amer. Inst. Electr. Eng.*, vol. 45, pp. 295–301, Jan. 1926.
[34] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.
[35] S. Leung-Yan-Cheong and M. Hellman, "The Gaussian wire-tap channel," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 4, pp. 451–456, Jul. 1978.
[36] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733–742, May 1993.
[37] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography. I. Secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121–1132, Jul. 1993.
[38] I. Csiszar and P. Narayan, "Common randomness and secret key generation with a helper," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 344–366, Mar. 2000.
[39] U. M. Maurer, "Provably secure key distribution based on independent channels," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Veldhoven, The Netherlands, Jun. 1990.
[40] M. Guillaud, D. T. M. Slock, and R. Knopp, "A practical method for wireless channel reciprocity exploitation through relative calibration," in *Proc. Int. Symp. Signal Process. Appl. (ISSPA)*, Aug. 2005, pp. 403–406.
[41] G. Merline and R. C. Porselvi, "Addressing temporal correlation in wireless channel for secured communication," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Melmaruvathur, India, Apr. 2013, pp. 428–432.
[42] S. Abbott and P. Hoffman, "The man who loved only numbers: The story of Paul Erdos and the search for mathematical truth," *Math. Gazette*, vol. 82, no. 495, p. 532, Nov. 1998.
[43] D. T. Murphy, "What, if anything, is epsilon?" in *Proc. 8th Annu. Intercalary Workshop About Symp. Robot Dance Party Conf. Celebration Harry Bovik's 26th Birthday (SIGBOVIK)*, Apr. 2014, pp. 93–97. [Online]. Available: http://sigbovik.org/2014/proceedings.pdf
[44] D. Jost, U. Maurer, and J. Ribeiro, "Information-theoretic secret-key agreement: The asymptotically tight relation between the secret-key rate and the channel quality ratio," in *Proc. 16th Int. Conf. (TCC)*, Panaji, India, Nov. 2018, pp. 345–369.
[45] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 364–375, Sep. 2007.
[46] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, and A. LaMarca, "Ensemble: Cooperative proximity-based authentication," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, New York, NY, USA, 2010, pp. 331–344.
[47] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-based authentication of mobile devices," in *Proc. Ubiquitous Comput. (UbiComp)*. Berlin, Germany: Springer, 2007, pp. 253–270.
[48] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "Proximate: Proximity-based secure pairing using ambient wireless signals," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, New York, NY, USA, 2011, pp. 211–224.
[49] G. R. Tsouri and J. Wilczewski, "Reliable symmetric key generation for body area networks using wireless physical layer security in the presence of an on-body eavesdropper," in *Proc. 4th Int. Symp. Appl. Sci. Biomed. Commun. Technol. (ISABEL)*, New York, NY, USA, 2011, pp. 153:1–153:6.
[50] S. T. Ali, V. Sivaraman, and D. Ostry, "Secret key generation rate vs. reconciliation cost using wireless channel characteristics in body area networks," in *Proc. IEEE/IFIP Int. Conf. Embedded Ubiquitous Comput.*, Dec. 2011, pp. 644–650.
[51] L. Shi, J. Yuan, S. Yu, and M. Li, "ASK-BAN: Authenticated secret key extraction utilizing channel characteristics for body area networks," in *Proc. WISEC*, 2013, pp. 155–166.
[52] G. Revadigar, C. Javali, H. Asghar, K. B. Rasmussen, and S. Jha, "iARC: Secret key generation for resource constrained devices by inducing artificial randomness in the channel," presented at the 10th ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS), Apr. 2015.
[53] P. Van Torre, "Channel-based key generation for encrypted body-worn wireless sensor networks," *Sensors*, vol. 16, no. 9, p. 1453, Sep. 2016.
[54] Z. Li, H. Wang, and H. Fang, "Group-based cooperation on symmetric key generation for wireless body area networks," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1955–1963, Dec. 2017.
[55] Z. Li, H. Wang, M. Daneshmand, and H. Fang, "Secure and efficient key generation and agreement methods for wireless body area networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[56] Y. Zhang, Y. Xiang, and X. Huang, "Password-authenticated group key exchange: A cross-layer design," *ACM Trans. Internet Technol.*, vol. 16, pp. 24:1–24:20, Dec. 2016.
[57] M. J. Siavoshani, C. Fragouli, S. Diggavi, U. Pulleti, and K. Argyraki, "Group secret key generation over broadcast erasure channels," in *Proc. 44th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2010, pp. 719–723.
[58] K. Argyraki, S. Diggavi, M. Duarte, C. Fragouli, M. Gatzianas, and P. Kostopoulos, "Creating secrets out of erasures," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, New York, NY, USA, 2013, pp. 429–440.
[59] A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," *Internet Math.*, vol. 1, no. 4, pp. 485–509, Jan. 2004.
[60] S. Geravand and M. Ahmadi, "Bloom filter applications in network security: A state-of-the-art survey," *Comput. Netw.*, vol. 57, no. 18, pp. 4047–4064, Dec. 2013.
[61] D. Randall. *Cs 6550? Design and Analysis of Algorithms*. Accessed: Oct. 15, 2018. [Online]. Available: http://people.math.gatech.edu/~randall/AlgsF09/bloomfilters.pdf
[62] J. Lu, T. Yang, Y. Wang, H. Dai, X. Chen, L. Jin, H. Song, and B. Liu, "Low computational cost Bloom filters," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2254–2267, Oct. 2018.
[63] J. Honorof. (Mar. 2006). *An Examination of Bloom Filters and their Applications*. [Online]. Available: https://cs.unc.edu/~fabian/courses/CS600.624/slides/bloomslides.pdf
[64] Acrylic. *Compatible Monitor Mode Wi-Fi Cards Under Windows*. Accessed: Jan. 20, 2017. [Online]. Available: https://www.acrylicwifi.com/en/support-webinars-wifi-wireless-network-s%oftware-tools/compatible-hardware/
[65] E. G. Villegas, E. Lopez-Aguilera, R. Vidal, and J. Paradells, "Effect of adjacent-channel interference in IEEE 802.11 WLANs," in *Proc. 2nd Int. Conf. Cognit. Radio Oriented Wireless Netw. Commun.*, Aug. 2007, pp. 118–125.
[66] P. Miklavčić, "On the number of non-overlapping channels in the IEEE 802.11 WLANs operating in the 2.4 GHz band," *Elektrotehniski Vestnik/Electrotech. Rev.*, vol. 81, no. 3, pp. 148–152, 2013.
[67] P. Fuxjager, D. Valerio, and F. Ricciato, "The myth of non-overlapping channels: Interference measurements in IEEE 802.11," in *Proc. 4th Annu. Conf. Wireless Demand Netw. Syst. Services*, Jan. 2007, pp. 1–8.

[68] S. Richter, V. Alvarez, and J. Dittrich, "A seven-dimensional analysis of hashing methods and its implications on query processing," *Proc. VLDB Endowment*, vol. 9, pp. 96–107, Nov. 2015.
[69] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Ad Hoc Netw.*, vol. 1, no. 1, pp. 107–123, Jul. 2003.
[70] F. Kazemeyni, E. B. Johnsen, O. Owe, and I. Balasingham, "Grouping nodes in wireless sensor networks using coalitional game theory," in *Proc. FMOODS/FORTE*, in Lecture Notes in Computer Science, vol. 6117, J. Hatcliff and E. Zucca, Eds. Berlin, Germany: Springer-Verlag, Jun. 2010, pp. 95–109.
[71] X. Liu, Y. Guo, W. Li, M. Hua, and E. Ding, "A complete feasible and nodes-grouped scheduling algorithm for wireless rechargeable sensor networks in tunnels," *Sensors*, vol. 18, no. 10, p. 3410, Oct. 2018.
[72] M. Gerla, L.-J. Chen, Y.-Z. Lee, B. Zhou, J. Chen, G. Yang, and S. Das, "Dealing with node mobility in ad hoc wireless network," in *Proc. 5th Int. Conf. Formal Methods Design Comput., Commun., Softw. Syst., Mobile Comput. (SFM-Moby)*. Berlin, Germany: Springer-Verlag, 2005, pp. 69–106.
[73] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications, revision 1a," NIST, Gaithersburg, MD, USA, Tech. Rep. Sp 800-22, 2010. [Online]. Available: https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software
[74] Brentp. (Sep. 16, 2015). *Tskidmarks 0.0.6*. Accessed: Jun. 20, 2019. [Online]. Available: https://pypi.org/ project/skidmarks/
[75] S. Vermeulen. *Mathematical and Scientific Processing in Python*. Accessed: Jun. 20, 2019. [Online]. Available: http://vermeulen.ca/python-math.html
[76] *NIST/SEMATECH Engineering Handbook of Statistical Methods*, U.S. Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Apr. 2012. [Online]. Available: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35d.htm
[77] F. A. Feldman, "Fast spectral tests for measuring nonrandomness and the DES," in *Advances in Cryptology*, C. Pomerance, Ed. Berlin, Germany: Springer, 1988, pp. 243–254.
[78] D. Biebighauser. (2000). *Testing Random Number Generators*. [Online]. Available: http://www-users.math.umn.edu/~garrett/students/reu/pRNGs.pdf
[79] D. Eck and J. Ryan. (Apr. 2012). *The Chi Square Statistic*. Accessed: Jun. 30, 2019. [Online]. Available: http://math.hws.edu/javamath/ryan/ChiSquare.html
[80] T. Hurst. (Oct. 15, 2018). *Bloom Filters Calculator*. Accessed: Dec. 2018. [Online]. Available: https://hur.st/bloomfilter/

DAVID SAMUEL BHATTI is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST), Islamabad, Pakistan. His areas of interest are networks, mobiles and smartphones security, and secure routing protocol of ad hoc networks. He is also working on secret key acquisition for wireless networks in the paradigms of ad hoc networks, body-worn, wearable, and the IoT devices.

SHAHZAD SALEEM received the M.S. degree in information and communication systems security from the KTH Royal Institute of Technology, Sweden and the Ph.D. degree in digital forensics from the Department of Computer and Systems Sciences, Stockholm University, Sweden. He is currently an Assistant Professor with the School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST), Islamabad, Pakistan. He is also working with the Department of Computing. He is doing research in Information Security with an interest in digital forensics. He has been working with industry standard products in the domain of digital forensics e.g. i2 analyst notebook, EnCase, FTK, XWays, UFED, XRY, and device seizure.