Security Assessment for Personal Health Data Management System

Pingping Wu, Won-Jae Yi and Jafar Saniie

Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, Illinois, United States

***Abstract*—In this paper, we explore and discuss the data security issues in a real-time remote patient health monitoring system, called the Personal Health Data Management System (PHDMS). The health monitoring system examined for this study consists of multiple wireless sensors, central data coordinator called the Wireless Intelligent Personal Communication Node (W-iPCN), an Android smartphone, and a remotely located central medical server. The PHDMS transmits raw sensor data from the W-iPCN to the Android smartphone using the Bluetooth connection. In addition, the system transmits raw sensor data using the Wi-Fi or cellular broadband connection between the Android smartphone and the central medical server. As a case study, we present a security protection scheme to encrypt and decrypt raw sensor data using multiple cryptologic algorithms. We applied this security protection scheme on the W-iPCN, Android smartphone and the central medical server to test their ability to process and transmit sensor data in real-time. This security assessment aims to observe the performance of data encryption and decryption, and real-time feasibility to remotely monitor patients' health.**

I. Introduction

In every hospital or health institution, patients' health records are confidential data, which have to be stored in a secure location and should only be accessible to the authorized personnel. In order to ensure the security and privacy of the patients' health records, certain mechanism must be implemented into the remote patient health monitoring system [1]. Security and privacy become more critical in remote systems where patients' data needs to be shared among the medical authorities and doctors. Also, the data acquired from the remote patient must be protected while being transmitted over the network. Since raw sensor data transmission is prone to data security breaches and privacy intrusions [2], we explore the potential security issues on the utilized wireless protocols, and provide a feasible solution. One such solution that provides real-time operations for the current Wireless Body Sensor Network (WBSN) utilizes cryptologic algorithms applied to raw sensor data in order to protect against any potential data intruders and/or imposters. In addition, the solution must be a cost-effective, secure and reliable algorithm, executed on the core system components in the system design. For example, one of the widely used encryption algorithms known as Advanced Encryption Standard (AES) can be used to encrypt and decrypt the raw sensor data. Another example can be the Ron Rivest, Adi Shamir, Leonard Adleman (RSA) algorithm that uses public and private keys for encryption and decryption or a cryptographic hash function such as a Secure Hash Algorithm (SHA) to conceal the relationship between the original and encrypted data [3].

In the WBSN architecture, we have designed a remote health monitoring system using an Android smartphone, a dedicated central node for data fusion and a central medical server for logging records into the database [4], [5], [6], [7].

Body and environment sensors are connected to the dedicated central node called as Wireless Intelligent Personal Communication Node (W-iPCN) using different types of wireless mediums. Amongst the various wireless protocols, the Bluetooth is a practical way to transmit sensor data to the W-iPCN, and from the W-iPCN to the Android smartphone.

The Wi-Fi or 3G/4G cellular broadband connection is used to transmit data from an Android smartphone to the central medical server to transmit data over the Internet. In this paper, we investigate the possible security issues and solutions that can improve the security and privacy of the patients' sensor data. We demonstrate that our system design can cope with real-time sensor data transmission, as well as securing patients' personal data.

II. Personal Health Data Management System

The Personal Health Data Management System (PHDMS) expands upon the WBSN structure [4], [5], [6], [7]. The PHDMS is designed to track personal or family health history, and to provide an environment for medical authorities and doctors to diagnose patients' health remotely. As shown in Figure 1, the system consists of various sensors important for the patient health monitoring and diagnosis, a dedicated central node on the user-end, an Android smartphone, and a central medical server. Sensors include accelerometers, gyroscopes, temperature and humidity sensors, and others to observe and detect anomalies and activities of the patient. The central node, known as the Wireless Intelligent Personal Communication Node (W-iPCN) [5], is designed to handle wireless protocols such as the Bluetooth. Also, the W-iPCN is responsible for combining and executing sensor data analysis algorithms. The outcome of the processed sensor data from the W-iPCN are transmitted to the

978-1-4673-9985-2/16/$31.00 ©2016 IEEE

Android smartphone via the Bluetooth protocol. The Internet connection on the Android smartphone is established using Wi-Fi to communicate with the central medical server. The central medical server holds patient's information in the database, which can be accessed remotely by medical authorities and doctors.

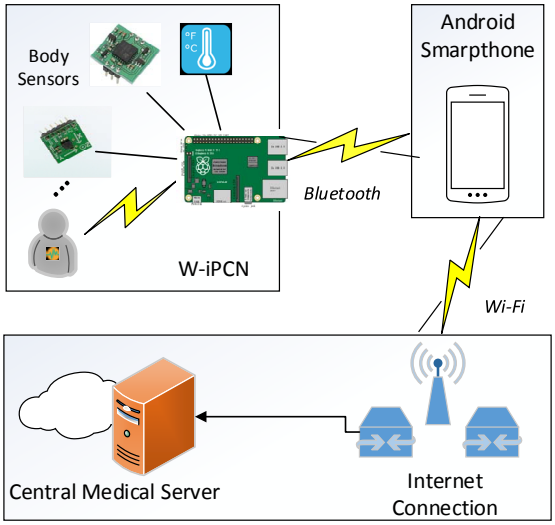

Figure 1. Overview of Personal Health Data Management System

The current design of the PHDMS acquires raw sensor data as well as processed patient data from the W-iPCN and transmits to the central medical server through the Android smartphone. Raw sensor data includes temperature, humidity sensor data, accelerometer and gyroscope information, and Electrocardiography (ECG) sensor data. Processed patient data includes body orientation and fall detection information, real-time heart beat rate, and patient's current location information [8]. In this paper, we investigate strong, efficient and cost-effective methods to protect the original sensor data and analyzed information from any unauthorized intrusions, data hijackers, and data corruption. In the next section, we discuss potential security issues that the WBSN may encounter.

III. Potential Security Issues

Privacy and security issues for personal health sensor data are important as addressed by HIPAA (Health Insurance Portability and Accountability Act) Privacy Rule [9]. In Figure 1, Bluetooth is the link between W-iPCN and the Android smartphone. Bluetooth is a low-cost, low-power technology that provides a mechanism to create a wireless Personal Area Network (PAN) [10]. But Bluetooth-enabled devices are prone to attacks against data confidentiality, integrity and availability [11]. Also, Bluetooth technology and associated devices are vulnerable to general wireless networking threats, such as denial-of-service (DoS), eavesdropping, man-in-the-middle (MITM) attacks, message modification, and resource misappropriation. In specific, there are threats associated especially with the Bluetooth technology such as bluesnarfing, bluejacking, and bluebugging [11]. To avoid security risks in our implementation and to leverage the security policy of Defense in Depth (DiD) [12], we apply higher-level protection solutions over the security features included in the Bluetooth protocol.

Wi-Fi has received the widest market acceptance within the wireless LAN technologies in the past decade [13]. According to Mobidia's analytics on mobile network usage, 62% of mobile devices use Wi-Fi and 38% of them use cellular network [14]. Many literatures have been published to uncover security vulnerabilities in Wi-Fi networks [15]. The common risks of the public Wi-Fi include sniffers, evil twin, MITM attacks and side jacking [16]. This determines the necessity for strong encryption of sensitive data in the PHDMS.

The W-iPCN utilizes a single-board computer known as the Raspberry Pi 2 [17]. The W-iPCN runs on a GNU/Linux-based operating system. In addition, the central medical server in the PHDMS also runs on the Linux-based operating system. The Linux operating system is recognized to be safer than the Microsoft Windows [18]. However, it is inevitable that at some point, the Linux-based system might be compromised by malicious activities such as a virus, worm, Trojan horse, or hacking [19], [20]. For example, Shellshock, designated as CVE-2014-7169 [21], is a bash bug that has the potential to compromise any computer running Linux/Unix/Mac operating systems.

Android, the most popular mobile device operating system is based on Linux kernel. Thus, certain software flaws that infect Linux are also threads to Android devices. According to F-Secure Lab's 2014 Mobile Thread Report, *275* different threats appeared on Android, while only *1* threat was identified on iOS and Symbian devices [22]. The four major security problems in Android that have been identified are operating system security updates, OEMs' effect on the security architecture of Android, Android permission model, and the Google Play market [23].

All the aforementioned issues lead to potential risks in the PHDMS. To alleviate attacks, we implemented application-level cryptographic algorithms upon raw sensor data to provide secure and protected WBSN system. The following section describes how the cryptographic algorithms protect the PHDMS against potential security risks and maintain real-time capability requirement.

IV. Security Concept and Features/Algorithm

The Federal Information Security Management Act (FISMA) defines three security objectives for data and data handling systems. These security objectives are Confidentiality, Integrity and Availability (CIA), as shown in Figure 2 [24].

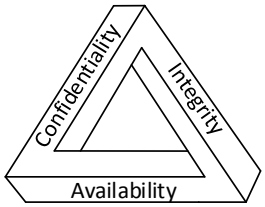

Figure 2. CIA Triad

To accomplish the above objectives, we guard the system during wireless data transmission by applying proper cryptography algorithms. The algorithms exploited in our proposed security scheme are public-known AES, RSA and SHA functions. AES is comparatively light-computation algorithm, commonly used to perform massive encrypting-decrypting user data at higher speed. In contrast, RSA is computationally expensive algorithm which is widely used to encrypt the shared key for symmetric key cryptography. SHA function is utilized in this system to perform data integrity verification.

A. Secured Channels

As depicted in Figure 1, wireless data transmission occurs in following channels: between the W-iPCN and Android smartphone through the Bluetooth connection, between Android smartphone and central medical server through the Wi-Fi connection. These channels, where we implement most of the secure strategies, are vulnerable to security threads as we discussed in Section III. Figure 3 outlines the security communication in this study to mitigate wireless cyber-attack and achieve the security goals. We apply the proposed secure scheme between W-iPCN and Android smartphone. The same scheme is utilized between Android smartphone and central medical server.

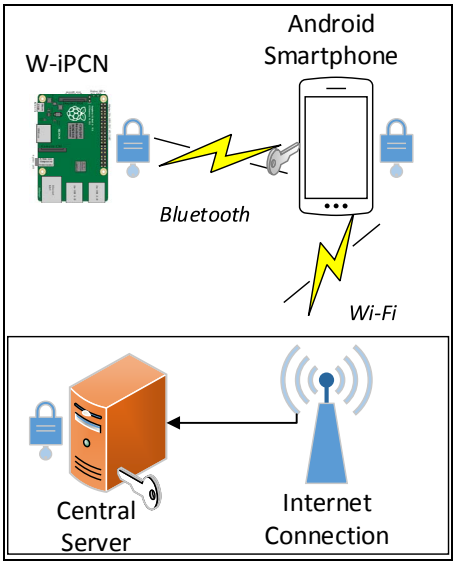

Figure 3. Secure Communication in PHDMS

Besides the proposed scheme, the Internet connection between Android smartphone and central medical server is running HTTPS protocol, which further strengthens the security level. This protocol will not result in real-time performance issues in the system. The critical point that may slow down the whole system throughput is between the central node and Android smartphone during cryptography, which incurs nontrivial computation and transmission overhead on either side.

To balance between real-time performance and security level, we propose a scheme to encapsulate raw sensor data to fulfill safety features, and simultaneously reduce the computation and transmission overhead. This is achieved by introducing and adopting appropriate cryptographic algorithms. Figure 4 describes the sender's encryption layer of the security scheme, and Figure 5 describes the receiver's decryption layer of the security scheme.

B. Proposed Security Scheme

Two parties involved in the security scheme are the sender (see Figure 4) and the receiver (see Figure 5). The data transmission cycle via air interface consists of two phases: Phase I and Phase II. Phase I is designed to refresh the shared key (Ks) used in AES between two parties; Phase II is the main process of transferring sensor data.

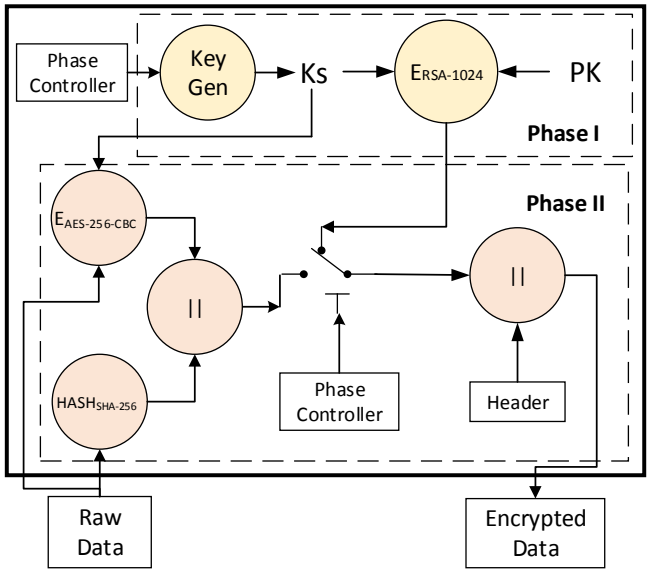

Figure 4. Sender's Encryption Layer of the Security Scheme

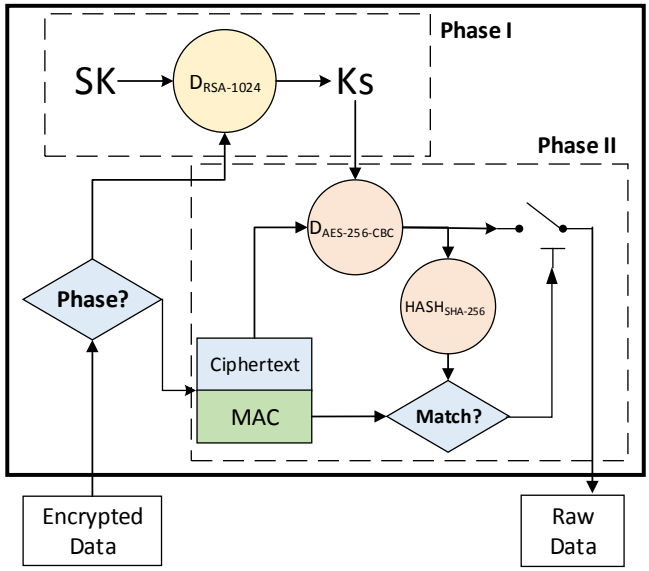

Figure 5. Receiver's Decryption Layer of the Security Scheme

During Phase I, a new key (*Ks*) is generated by the sender (noted as *Key Gen* in Figure 4) which is used for symmetric cryptography (i.e. AES) in Phase II. The key is then encrypted

by an asymmetric cryptography algorithm (RSA, noted as $E_{RSA-1024}$ in Figure 4) using the receiver's public key (*PK*). The sender transfers the unintelligible encrypted key together with a header to the receiver. The receiver shown in Figure 5 interprets the received package based on the header information. Once it detects a new-key header, it decrypts the information using RSA (noted as $D_{RSA-1024}$ in Figure 5) with its secret key (*SK*) and recover the key (*Ks*) for future AES decryption usage.

In Phase II, the sender transfers the sensor data to the receiver. At the sender's side shown in Figure 4, the raw information is handled by two functions respectively: hash and AES. The hash function (SHA-256, noted as $HASH_{SHA-256}$ in Figure 4) generates a Message Authentication Code (MAC) which is used to check data integrity at the receiver's side. AES algorithm uses the key (*Ks*) which is generated in Phase I to encrypt the raw sensor data and produce ciphertext (noted as $E_{AES-256-CBC}$ in Figure 4). This ciphertext is further concatenated with the MAC and a header before transferring to the receiver. As long as the sender is at active mode, the encryption layer is looped over Phase II. For the receiver in Figure 5, upon receiving a header denoting Phase II, the received information is split into a ciphertext and a MAC. The ciphertext is fed into the AES decryption function (noted as $D_{AES-256-CBC}$ in Figure 5) using the shared key (*Ks*) to recover the raw sensor data. A verification process is achieved by comparing the received MAC with the newly generated MAC by the hash function using the output of AES algorithm (noted as $HASH_{SHA-256}$ in Figure 5). The acceptance of sensor data at the receiver depends on the verification result.

C. Security Scheme Properties

1) Periodic Key Refresh for AES Algorithm

To sustain high level of security, we designed a random key generator to produce fresh key (Ks) for AES algorithm. In this way, we eliminate the potential risk of using the same key for all central nodes and mobile devices when applying symmetric AES-256 cryptography. In addition, the standard cryptography algorithm RSA-1024 guards the new key distribution process.

2) Data Transformation

Whenever sensor data or shared key (Ks) for AES is presented during wireless transmission, they are both transformed by certain cryptography in the form of unintelligible text. Even if the eavesdropper intercepts the data, the unreadable data format gives the adversary a slim chance to recover the original message.

3) Exhibiting Data Security Objectives

As shown in Figure 5, data integrity is verified by the receiver through the comparison of the two MACs: the received MAC and the generated MAC by hash function with its input from the output of AES decryption. If and only if both MACs match, we claim the data are original and have not been altered during transmission; at the same time the key (Ks) is also authenticated successfully by RSA and the integrity verification. Finally, data confidentiality is met by encrypting raw sensor data for wireless transmission.

4) Optimizing Real-time Performance

RSA is a relatively computation-intensive algorithm, which restricts its utilization to directly encrypt the user data. Often times, RSA is used to encrypt the shared key for symmetric key cryptography, which in turn can perform bulk encryption-decryption operations at much higher speed [25]. In our security implementation scheme, we carefully adopt the favorable side of RSA-1024 to help delivery of the secret key for AES-256 on Phase I and fully deploy light-computation feature of symmetric cryptography on Phase II. As sensor data transmission takes place exclusively on Phase II, we reduce the computational load by avoiding RSA-1024 on every transmission.

5) Flexibility of the Security Scheme

Various applications can adopt our proposed security scheme due to its simplicity and flexibility. The security scheme can further ease the computational load by adopting more lightweight cryptography algorithms when needed. The simplified design of input and output interface of the security scheme enables other users to address the security issues in their systems.

V. Performance Results

The implementation of the proposed security scheme is achieved to investigate the feasibility of the PHDMS as a secure real-time patient monitoring system by applying the cryptologic algorithms discussed above. In this paper, a data set of ECG sensor data at a sampling rate of 250 Hz (approximately 60 Kbytes per minute), is used to test individual devices' performance. Table I shows the specifications of the devices used in the PHDMS.

TABLE I. Specifications of PHDMS Components

	Hardware Specification		
	CPU	**RAM**	**Operating System**
W-iPCN [17] **(Rapsberry Pi 2)**	ARM Cortex-A7 Quad-core 900 MHz	1 GB	Raspberian (ARM GNU/Linux Distribution)
Android Smartphone [26] **(Samsung Galaxy Note 5)**	Samsung Exynos 7402 Octa (ARM Cortex-A53 Quad-core 1.5 GHz & ARM Cortex-A57 Quad-core 2.1 GHz)	4 GB	Android 5.1.1
Server [27] [28] **(Intel i7-2600)**	Intel i7-2600 Quad-core 3.8 GHz	12 GB	Ubuntu 14.04

A. Test Set Data Characteristics

The criterion for choosing test dataset is to pick up the sensor data that has the heaviest payload. The ECG signal sampled at 250 Hz with each sample being 4 Bytes requires bandwidth of approximately 1000 Bytes/s. Other sensor data,

like ambient temperature or gyroscope, has much less stringent bandwidth requirement. Thus, the ECG sample is an appropriate test case for evaluating the overhead caused by the encryption and decryption.

B. Computational Costs

The computational devices shown in Table I have multi-core processors. This implies that computational results may vary for each experiment. Therefore, the results presented in Table II are the average computation load of 10 trials and the related computational load for each device for Phase I and Phase II. This study involves 15,000 samples of ECG data, which is equivalent to 60 seconds of ECG recording.

TABLE II. COMPUTATIONAL COST OF CRYPTOGRAPHIC ALGORITHMS

		W-iPCN	Android Smartphone	Server
Phase I	$E_{RSA-1024}$	6.183 ms	0.195 ms	--
	$D_{RSA-1024}$	--	2.623 ms	6.23 ms
Phase II	$E_{AES-256-CBC}$	641 ms	113 ms	--
	$D_{AES-256-CBC}$	--	95 ms	58 ms
	$HASH_{SHA-256}$	312 ms	36 ms	29 ms

C. Performance Assessment

Two phases are defined in our proposed security scheme according to their functionality lead to a certain computation load for our system. Phase II is the main process for transmitting sensor data (e.g., the ECG). The Android smartphone and server both deliver less computation cost compared to the W-iPCN. Thus, the real-time feature of the platform encompassing security consideration depends on the W-iPCN's computation ability, which consumes 641 ms for AES encrypting process and another 312 ms for hash function. The total computation cost is close to 1 second for encrypting 15,000 samples of ECG data. The computation overhead for encryption is less than 2%.

Phase I generates a key for AES algorithm using Java programming. Here we measure the computational load of RSA encryption and decryption process. Each device consumes less than 10 ms to complete this phase. In addition, this process is achieved only in the initial stage of establishing data communication among the devices. We notice that the Android smartphone outperforms the server for RSA1024 decryption. This is due to the difference in Java execution environment between the Android smartphone and server. The Android smartphone executes Java codes under Android Runtime (ART), and the server executes Java codes under Java Virtual Machine (JVM) environment. They have different approaches to execute Java code, where the ART has more efficient processing mechanism than the JVM [29]. On the other hand, Phase II is implemented in C/C++ programming environment where the performance depends on the processor's computational power.

In Phase II, one sample of ECG data is 4 Bytes and there are 250 samples per second. According to the nature of AES-256 and SHA-256, each set of raw ECG data (4 Bytes) will lead to 64 Bytes of encrypted and concatenated data, which means 16 Kbytes/s. This requires a minimum bandwidth of 16 Kbytes/s for the Bluetooth or Wi-Fi. The Bluetooth communication has maximum throughput of either 384 Kbytes/s or 3 Mbytes/s, depending on the Bluetooth versions. These specifications entail that real-time data transmission requirement is satisfied. Alternatively, Wi-Fi can provide higher bandwidth when compared to the Bluetooth protocol.

VI. CONCLUSION

In this paper, we discussed and investigated potential security issues of the remote health patient monitoring system. Using the PHDMS as a model, we explored types of cryptologic algorithms to protect raw sensor data when it is transmitted over the data communication network. With the chosen algorithms, we proposed a security scheme that maintains real-time sensor data transmission, as well as obtaining data security and privacy. Methods and algorithms were implemented on the devices in the PHDMS to prove that they are feasible for secure real-time patient monitoring applications. Using our proposed security scheme, the PHDMS was able to establish a secure data transmission mechanism in real-time.

REFERENCES

[1] M. Li, W. Lou and K. Ren, "Data Security and Privacy in Wireless Body Area Networks," *IEEE Wireless Communications,* vol. 17, no. 1, pp. 51-58, 2010.

[2] M. Meingast, T. Roosta and S. Sastry, "Security and Privacy Issues with Health Care Information Technology," *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* pp. 5453-5458, 2006.

[3] W. Stallings, Cryptography and Network Security - Principles and Practice, New York: Prentice Hall, 2011.

[4] W.-J. Yi, J. Saniie, "Smart Mobile System for Body Sensor Network," *2013 IEEE International Conference on Electro/Information Technology,* pp. 1-4, 2013.

[5] W.-J. Yi, S. Niu, T. Gonnot, J. Saniie, "System Architecture of Intelligent Personal Communication Node for Body Sensor Network," *2013 IEEE International Instrumentation and Measurement Technology Conference,* pp. 1717-1720, 2013.

[6] W.-J. Yi and J. Saniie, "System Architecture and Design Flow of Smart Mobile Sensing Systems," *Journal of Sensor Technology,* vol. 3, no. 3, pp. 47-56, 2013.

[7] W.-J. Yi, W. Jia and J. Saniie, "Mobile Sensor Data Collector using Android Smartphone," *IEEE 55th International Midwest Symposium on Circuits and Systems,* pp. 956-959, 2012.

[8] W.-J. Yi, O. Sarkar, S. Mathavan and J. Saniie, "Design Flow of Wearable Heart Monitoring and Fall Detection System using Wireless Intelligent Personal Communication Node," *2015 IEEE International Conference on Electro/Information Technology,* pp. 1-4, 2015.

[9] U.S. Department of Health & Human Services, "Privacy | HHS.gov," 2016. [Online]. Available: http://www.hhs.gov/hipaa/for-professionals/privacy/. [Accessed 25 2 2016].

[10] S. Radack, "Security of Bluetooth Systems and Devices: Updated Guide Issued by the National Institute of Standards and Technology (NIST)," August 2012. [Online]. Available: http://csrc.nist.gov/publications/nistbul/august-2012_itl-bulletin.pdf.

[11] J. Padgette, K. Scarfone and L. Chen, "Guide to Bluetooth Security," June 2012. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-121-rev1/sp800-121_rev1.pdf.

[12] National Security Agency, "Defense in Depth," March 2010. [Online]. Available: https://www.nsa.gov/ia/_files/support/defenseindepth.pdf.

[13] V. Wekhande, "Wi-Fi Technology: Security Issues," *Rivier Academic Journal,* vol. 2, no. 2, pp. 1-16, 2006.

[14] mobidia, "Network Data | Analytics | Mobidia," 2016. [Online]. Available: http://www.mobidia.com/analytics/network-data.

[15] A. R. Al Tamimi, "Security in Wireless Dat Networks: A Survey Paper," 2006. [Online]. Available: http://www.cse.wustl.edu/~jain/cse574-06/ftp/wireless_security.pdf.

[16] Private WiFi, "The Hidden Dangers of Public WiFi," October 2014. [Online]. Available: http://www.privatewifi.com/wp-content/uploads/2015/01/PWF_whitepaper_v6.pdf.

[17] Raspberry Pi Foundation, "Raspberry Pi 2 Model B," Raspberry Pi Foundation, 2015. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. [Accessed 23 2 2016].

[18] N. Petreley, "Security Report: Windows vs Linux," October 2004. [Online]. Available: http://www.theregister.co.uk/2004/10/22/security_report_windows_vs_linux/.

[19] S. Prakasha, "Security in Linux," December 2009. [Online]. Available: http://www.linuxuser.co.uk/features/security-in-linux.

[20] C. Florian, "Most vulnerable operating systems and applications in 2014," February 2015. [Online]. Available: http://www.gfi.com/blog/most-vulnerable-operating-systems-and-applications-in-2014/.

[21] National Vulnerability Database, "NVD - Detail," May 2015. [Online]. Available: https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-7169.

[22] F-Secure Labs, "Mobile Threat Report Q1 2014," 2014. [Online]. Available: https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014.pdf.

[23] S. Liebergeld and M. Lange, "Android Security, Pitfalls and Lessons Learned," *Information Sciences and Systems,* vol. 264, pp. 409-417, 2013.

[24] U.S. Department of Commerce, "Standards for Security Categorization of Federal Information and Information Systems," February 2004. [Online]. Available: http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf.

[25] R. Mathur, S. Agarwal and V. Sharma, "Solving Security Issues in Mobile Computing using Cryptography Techniques - A Survey," *2015 International Conference on Computing, Communication & Automation,* no. http://dx.doi.org/10.1109/CCAA.2015.7148427, pp. 492-497, 2015.

[26] Samsung, "Samsung Galaxy Note 5 - The Offical Samsung Galaxy Site," Samsung, 2015. [Online]. Available: http://www.samsung.com/global/galaxy/galaxy-note5/. [Accessed 23 2 2016].

[27] Dell, "Dell XPS 8300," Dell, 2016. [Online]. Available: http://www.dell.com/us/dfh/p/xps-8300/pd. [Accessed 23 2 2016].

[28] Intel, "Intel® Core™ i7-2600 Processor (8M Cache, up to 3.80 GHz) Specifications," Intel, 2016. [Online]. Available: http://ark.intel.com/products/52213/Intel-Core-i7-2600-Processor-8M-Cache-up-to-3_80-GHz. [Accessed 23 2 2016].

[29] R. Yadav and R. Bhadoria, "Performance Analysis for Android Runtime Environment," *2015 Fifth International Conference on Communication Systems and Network Technologies,* pp. 1076-1079, 23015.