Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 10, Issue. 3, March 2021, pg.72 – 82

Security of Sensitive Data on Android Smartphones Using Cloud Storage with Reference to Gravitational Search Algorithm

Solomon Babatunde Olaleye
olaleye.solomon1115@fcesoyo.edu.ng
Department of Computer Science, Federal College of Education (Special), Oyo, Nigeria
DOI: 10.47760/ijcsmc.2021.v10i03.009

Abstract

The focus of this paper is to secure sensitive data on smartphones using cloud infrastructure. Nevertheless, the biggest challenge is data security. This work, therefore, addresses the issues of security of user data by using cloud storage with the development of Solo App v2.1 android application which adopted an Advanced Encryption Standard (AES) for maintaining the confidentiality of data. This will enable smartphones data to remain secure either at rest or in transit. Since numerous threats to mobile phones nowadays have made the issue of security of data on smartphones very critical, hence, the need for the development of such application. Three Android devices were used for the experiments to implement the application. The results are presented using time performance metrics that is based on encryption time and decryption time for each device. The results showed that the work can effectively protect user sensitive data with speed using cloud and when compared with existing works, the results are satisfactory.

Keywords: *GSA, Android smartphones, security, cloud storage, sensitive data*

1. Introduction

Security of data against attacks in mobile devices is an issue of indispensable global concern. To ensure the confidentiality, integrity and availability of data to genuine users, continuous research must be carried out, since they are always connected to the Internet. In recent times, smartphones have become an essential part of everyday lives of people. Securing sensitive data on smartphones have become more important, because of increasing usage of mobile devices with Internet (Donald et al., 2013). Android smartphone is not fully secured as it

appears (Tiwari et al., 2013), security is one of the main concerns for smartphones users today with their susceptibility for attacks by malware, viruses, security threats, loss etc. (Mandeep & Kanwalvir, 2013).

Similarly, there is no robust facility to save data in the cloud with appropriate extraction in android phones. Memory cards have reduced data security and reliability. Security is regarded as an essential concern for any smartphone which comprises sensitive information and accesses the Internet. Due to inherent nature of these appliances such as portability and mobility they face additional security problems contrast to conventional devices of computing. The world-wide adoption of smartphones has surpassed the use of personal computers and laptops computers as a leading computing platform due to mobility, constant connectivity and many suitable or interested applications (Altuwaijri & Ghouzali, 2018).

Insecure data storage is one of the leading top 10 security issues in smartphones since sensitive information can be revealed if it is not protected carefully. An attacker who accesses smartphone physically can attach the phone to a computer and retrieve sensitive personal information, (OWASP Mobile Security Project-OWASP, owasp.org, 2016).

The authors (Ahvanooey, Li, Rabbani & Rajput, 2017) provided a comprehensive overview of mobile threats, vulnerabilities and countermeasures by reviewing published papers during the period 2011-2017. Also, they demonstrated vital points to ensure mobile security based on restricting malicious activities at several levels: application's store level and operating system level. Further, Altuwaijri & Ghouzali (2018) stated that securing data stored on smartphones has become a key issue. The authors investigated the security of Android storage model between 2009-2019. Several threats were found in the literature that can be categorized as physical or software threats. Although, Android provides valuable encryption systems including full disk encryption and keychain to enhance the data storage security, the encryption key, which is stored in the device, is still vulnerable to physical threats. Confidential data may exist in memory on mobile devices for a long time after being used. Thus, on stolen device, the retrieval of sensitive information is possible and becomes a growing concern.

In addition, Android Enterprise Security White Paper (2020) stated that Android security by design uses hardware and software protections to achieve strong guards. At hardware level, the user is authenticated with lock screen credentials. At the software layer, built-in protection is essential to helping Android devices stay safe. Application sandboxing isolates and guards Android apps, preventing malicious apps from accessing private information. Android also protects access to internal operating system components that are exploitable. Android applications are usually downloaded from Google Play Store. The Google Play Store provides threat detection services, actively scanning over 50 billion apps on devices every day to monitor harmful behaviour. At the moment, Android supports the following security features: Biometrics, Fingerprints authentication, Face authentication, etc.

Consequently, Hayajney, Bhuiyan & McAndrew (2020) reported that new cloud computing with its numerous resources can help in securing and storing smartphones data. Also, can overcome the limited battery, processing and storage of mobile devices. It can be deduced that with the integration of cloud computing with mobile devices (mobile cloud computing) has the capability to allow mobile application developers to develop mobile application that can provide security on smartphones by encryption and further send such encrypted data to

the cloud which can be decrypted at anytime and anywhere by the user by supplying appropriate credentials with Internet connectivity.

The main aim of this paper is to address the security of data on Android smartphone using cloud storage and at the same time making use of Gravitational Search Algorithm (GSA) technique for finding the optimal location of the cloud servers for storing and retrieving of stored data. Based on this study, the following objectives are set to guide this work; to identify existing security techniques for securing data on smartphones and to implement a proven security algorithm for enhancing security of user sensitive data on Android smartphones with linkage of Amazon EC2 cloud storage. A mobile application named Solo App is developed and implemented using advanced encryption standard (AES) to provide security of data on smartphones. It is placed on Google Play store for public use and appropriate feedbacks. The Solo App can successfully encrypt and decrypt user data on Android smartphones.

2. Android Sandboxing

According to Singh (2014), a sandbox is a security mechanism for limiting mobile application to mobile device resources. An application must not be allowed to access some device resources for security reasons. It is usually used to run unverified applications from unreliable developers and insecure websites. The security of mobile devices is enhanced by sandboxing technique. For instance, a malicious code is denied entry to a mobile device with the help of sandbox except permission is granted.

Android provides two security mechanisms, one is a sandbox mechanism which is positioned in the kernel space (Kandalkar & Bartere, 2016) and the second is a permission mechanism (Fang et al., 2014; Yang et al., 2015). For the sandbox, Android allocates a unique User ID (UID) to individual application and executes the application in a distinct process. On the permission mechanism an application is only allowed to gain access to constrained mobile phone resources. Whenever a smartphone user wants to install an application, Android will always request permission from user. If permission is granted by the user, the application is installed else it will not be installed.

However, Android authenticates the legitimacy of an application through the UID of the application. Thus, the two security mechanisms rely on the UID of an application to provide security. Moreover, Android confines application interaction to its particular Application Programming Interfaces (APIs) by executing each application using its own user identity. From experience, developing a secure application is not easy. Likewise, the use of effortless permission restricts access to mobile phone resources and other applications (Enck et al., 2009; Kawabata et al., 2013). Nevertheless, some users do not always pay full attention to the permission request of an application because an application may request to access more than the resources required to function (Borde & Nigrel, 2016).

3. Types of Security Attacks in Smartphones

The remarkable increase in smartphones usage has led to uprising in the number of users as well as increase in Internet connectivity. This escalating number of smartphones users and connectivity provide hackers with the chance to write various malicious applications. Compared to other Operating Systems (OS) on smartphones, Android is the most regularly attacked, because it is an open source OS (Narudin et al., 2016). There are a number of existing malware and vulnerabilities which are discovered and reported by different security

providers that attack smartphones. Some known threats from telecommunication networks to smartphones and from smartphones to telecommunication networks are Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, Peer-to-peer connections or connectivity via Bluetooth or infrared.

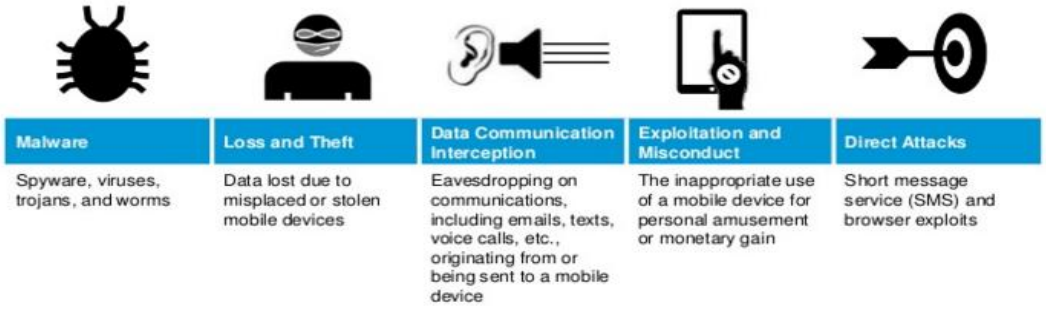

Figure 1. Types of security attacks in smartphones (Hp, 2012)

Figure 1 shows the types of security attacks in smartphones. These infectious applications can affect smartphone security to a great extent (Poonguzhali et al., 2016). They are briefly discussed below along with others as found in literature.

i. Malware: Attackers exploit just a single vulnerability to attack different kinds of devices by causing major security outbreaks (Buschkes et al., 1998). The majority of the attacks in smartphones use social engineering tactics to convince the user to install or subscribe attacker-controlled service. There are various types of malware attacks on smartphones, such as Spyware Attacks, Surveillance Attacks, Financial Malware attacks, Worm-based attacks, Botnets etc. Hackers try to monitor remotely by a set of Zombie devices that are infected by malware. Today, many new malwares which are unidentified are constantly created to attack smartphones. And smartphones are dependent on anti-virus applications for detection and prevention due to resources limitation (Nissim et al., 2016).
ii. Keystroke Logging: Keystroke logging is a kind of malware that registers keystrokes on mobile devices to capture sensitive data such as credit card numbers or personal details.
iii. Loss and Theft: This is due to data lost by misplaced or stolen smartphones. Smartphones are simple to be stolen because of their smaller sizes or misplaced than notebooks and laptop PC (Donald et al., 2013). With physical control, it is simple for attackers to acquire access to data which the hand held appliances store or are capable to access remotely (Diksha et al., 2014). Likewise, services of electronic tracking permit the location of registered cell phones to be supervised and monitored. This tracking can be performed openly for legal purposes but it may also be performed surreptitiously.
iv. Data Interception: Data interception is a security attack which exists when an attacker is eavesdropping on communications originating from or being sent to a mobile device (Tekade & Shelke, 2014). Electronic eavesdropping is feasible through different techniques such as wireless fidelity sniffing, man in middle attacks, electronic mails, text messages from one smartphone to another and so on.
v. Spam: It uses pop up messages or electronic mail to deceive people into disclosing sensitive data which is known as phishing (GAO, 2012). Spam is also a security threat to smartphone which is an unsolicited commercial electronic mail ad for services, websites and products. Spam can be used as a delivery process for malicious software. Spam can be seen in e-mail as well as text messages. This can happen by using Short Message Service (SMS) and also vulnerability through browser exploits.

vi. DOS: A Denial of Service (DoS) attack is another type of security attack in smartphones which have been around for a long time and are not specific or new to mobile appliances. DoS attacks render a device or service unusable for its legal users avoiding availability. The issues with DoS attacks against smartphones are related much towards reduced capabilities and strong connectivity. To be specific DOS attacks against smartphones could use the fact that these appliances function on batteries. Therefore, the aim of the attack is to rapidly drain targeted device batteries. Successful attacks will dramatically restrict or shut down the target's operation time (Erich & Cliff, 2013; Janal, 2014).

vii. Infrastructure Based Attacks: Infrastructure based attacks are another type of security attack in smartphones. Since the services offered by infrastructure are the basis for important functionalities of smartphone such as receiving or placing calls, electronic mail services and SMS, the social and economic influence of these attacks may be huge. The security influence of interface of SMS on feasibility of mobile phone network is evaluated. If an attacker is capable of sending messages through numerous available portals simultaneously through numerous feasible portals into the network of SMS, the resulting aggregate load can saturate the channels of control and therefore block SMS communication and legal voice (La Polla et al., 2013; Zameshkumar & Vijay, 2014).

4. Cloud Storage

Cloud storage is accessed by Internet connectivity. One can use 3G/4G/5G link or Wireless Fidelity (Wi-Fi) to access cloud services. Cloud computing has really changed the way individuals and organizations are conducting services today. This is achieved because of the numerous benefits that are provided by the cloud. Modern mobile devices are designed to access cloud services. Cloud provides a lot of benefits on one side and security risk on the other (Cimler et al., 2015). Security is a major concern when considering storing part or all of your data in the cloud. Privacy considerations are very important before a user or organization opts for cloud. As a cloud user, once you store your data, how will you be sure that your data is safe and secure (Anthony et al., 2009)? The truth is that you are not really sure. Before using the cloud for sensitive data, one must be certain that the cloud provider has a very strong mechanism for protecting sensitive data in its custody. Thus, this work developed a mobile application that encrypts user data on user smartphone before sending to cloud for storage.

5. AES

The encryption algorithm used is AES-128 with Cipher-Block Chaining (CBC) which was developed by Joan Daemen and Vincent Rijmen. Advanced Encryption Standard (AES) which was the best proven security algorithm as decided in autumn 2000. AES is a block cipher and it was designed as a substitution permutation network. It has a strong mathematical structure which operations were based on arithmetic in the finite fields. It could operate on block sizes 128, 192 or 256 bits while its final standard block size was fixed at 128 bits. AES supports keys of size 128, 192, or 256 bits and each key size supports 10,12 and 14 number of rounds respectively (Smart, 2016).

The AES has a set of four operations that operate as a round function. The four operations are SubBytes, ShiftRows, MixColumns and AddRoundKey.

Algorithm1: AES encryption pseudo-code

AddRoundKey (S, K_0).
for i=1 to 9 do
  SubBytes (S)
  ShiftRows (S).
  MixColumns (S).
  AddRoundKey (S, K_i).
Sub Bytes (S).
Shift Rows (S).
Add Round key (S, K_{10}).

The AES algorithm1 represents the encryption algorithm where S is the state matrix, which is the message block to be encrypted i.e. the input. $K_i=K_0 \ldots K_{10}$ are the round keys.

Algorithm 2: AES decryption pseudo-code

AddRoundKey(S, K_{10}).
InverseShiftRows(S).
InverseSubBytes(S).
for i=9 downto 1 do
AddRoundKey(S, K_i)
InverseMixColumns(S).
InverseShiftRows(S).
InverseSubBytes(S).
AddRoundKey(S, K_0).

The Algorithm 2 represents the AES decryption which is the inversed of algorithm1 i.e. AES key schedule computes the round keys from the main key to produce 11 round keys of $k_0,\ldots,k_{11}$.The AES key schedule uses a round constant represented as $RC_i \leftarrow x^i \pmod{x^8+x^4+x^3+x+1}$.

The round keys are labelled as $W_{4i}, W_{4i+1}, W_{4i+2}, W_{4i+3}$ where *i* is the round. The first key is divided into four 32-bit words(K_0, K_1, K_2, K_3).The round keys are then computed using Algorithm 3, where RotBytes serves as the function that rotates a word to the left by a single byte and SubBytes applies the AES encryption S-Box to every byte in a word.

Algorithm 3: AES key schedule pseudo-code

$W_0 \leftarrow k_0, W_1 \leftarrow K_1, W_2 \leftarrow K_2, W_3 \leftarrow K_3.$
for i←1 to 10 do
T←RotBytes (W_{4i-1}).
T←SubBytes (T).
T←TΦ RC_i.
$W_{4i} \leftarrow W_{4i-4}$ ΦT.
$W_{4i+1} \leftarrow W_{4i-3}$ ΦW_{4i}.
$W_{4i+2} \leftarrow W_{4i-2}$ Φ W_{4i+1}.
$W_{4i+3} \leftarrow W_{4i-1}$ Φ W_{4i+2}.

6. Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) is one of the applications of swarm intelligence that provided solutions to scientific problems by considering objects whose function is controlled by their masses. All these objects draw to each other by force that causes an overall movement of objects towards the objects with heavier masses. The position of an object relates to a resolution of a problem and inertial masses are determined by a fitness function

(Eldos & Qasim, 2013). GSA is a heuristic optimization algorithm which was introduced by Rashedi et al. (2009) and the approach is based on Newton's law of gravitation and motion (Wang & Song, 2016).

6.1 Finding the Location of Cloud Server

To establish the best location of a cloud server the GSA is used for establishing the optimal locations of the cloud servers. Whenever a user needs to access cloud server, user credentials are forwarded using mobile network services. The network provider in mobile provides services like verification, approval and accounting. By doing these functions, the demand is sent to the cloud by means of Internet. Each cloud owns a cluster of networks N= {N_1, N_2, N_3, N_4…N_n}. A gathering of application servers S= {S_1, S_2….S_s} with a gathering of resources R= {r_1, r_2….r_r} which can be used by cloud users on demand. Based on user Internet protocol, the network administrator connects with the cloud server.

The proposed work process is described as follows:

Supposing that M agents (servers) exist in the cloud, every server S_i is chosen as an agent that represents a unique solution (servers) which are assigned to each client's subjobs SJ_i. Let SJ_i= {SJ_1, SJ_2….SJ_i; i=1,…n} of n mobile users to be executed in a specified time. The agent (server) position is refreshed at every phase. The direction and velocity of an agent is used for anticipating the user's next position, if an agent's velocity is not negative, then it moves towards positive direction else vice versa. The GSA algorithm used for predicting the optimal positions of the cloud servers are;

1. Start
2. Initialize the number of agents (cloud servers)
3. Evaluate the fitness value using binomial distribution
   Fit $_j$ (t) = $_nC_x$* p^x* (1-p) $^{n-x}$
4. Determine the best and worst fitness value for each agent
   Best (t) = max fit $_j$ (t) j € (1…N), Worst (t) = min fit $_j$ (t) j € (1…N)
5. Calculate the mass of agents (CPU usage). This step determines the resource utilization $M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}$ where $m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$
6. Estimate the acceleration (memory usage) of agents $a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}$, $F_i^d(t) = \sum_{j \in Sbes, j \neq i} rand_j F_{ij}^d(t)$
7. Let t ← t + 1
8. The position of an agent is determined using the velocity of those agents $v_i^d(t+1) = rand_i * v_i^d(t) + a_i^d(t)$
9. If appropriate position of agent is found step 10 else step 2
10. Stop

7. Experimental Setup and Methodology

The hardware requirements for the experiment is as shown in Table 1. Three Android smartphones of different configurations were used based on convenience and availability (Infinix Hot 8, Techno Pouvour 3 and Infinix S5 Lite). The RAM range from 2GB to 4GB, internal storage with minimum of 32 GB and maximum of 64 GB and the battery capacity minimum of 4000mAh and maximum of 5000mAh. The three mobile devices were fully charged and background applications deleted.

The mobile application developed was named Solo App v2.1 and was downloaded from Google Play store into the three Android smartphones. Solo App v2.1 used AES algorithm and was evaluated using different file sizes of data blocks ranging from 14.44kb to 8340kb to encrypt and decrypt the data block. All implementations were exact to ensure that the results are relatively fair and accurate. This work uses secret key cryptography (symmetric cryptography) which is secure, faster, reliable and uses less storage on Android smartphones unlike the public key cryptography (asymmetric cryptography) which is computationally expensive, uses more storage space and high energy consumption which are already constraints in Android smartphones.

Table 1. Hardware requirements

	Mobile Devices		
	Infinix Hot 8	**Tecno Pouvoir 3**	**Infinix S5 Lite**
OS	Android TM 9.0 (Pie)	Android 8.1.0	Android 9.0 (Pie)
CPU	2.0 GHz * 4	1.28 GHz * 4	2.0 GHz * 8
RAM	2 GB	2 GB	4 GB
Internal Storage	32 GB	32 GB	64 GB
Battery	5000mAh	5000mAh	4000mAh

7.1 Precautions

The following precautions were taken during the experiment;

1. The experiments were repeated three times and average recorded.
2. The digital stop watch used was fully charged and believe to be accurate.
3. All background applications of the three Android smartphones used for the experiment were stopped from running.

8. Results and Discussion

A block of plaintext data is given as input to the Solo App v2.1 for encryption. The ciphertext which is the result of the encryption is then sent to the cloud for storage with secret key. A user can retrieve the ciphertext whenever needed with the secret key and Internet connectivity. If the user credentials are correct, then the ciphertext is converted back to plaintext on the user Android smartphone otherwise after three attempts the user will be denied. This research uses time performance evaluation metric. The input plaintext is in kilo bytes while the encryption time and decryption time are measured in seconds. For different ten inputs the corresponding encryption time and decryption time are tabulated. The results when compared with the results published by Reddy & Babu (2013), the are faster in times of encryption and decryption.

The results of the encryption and decryption of Solo App v2.1 on Infinix Hot 8 is as shown in Table 2.

Table 2. Infinix Hot 8

S/N	File Size (KB)	Encryption Time (ET) (sec)	Decryption Time (DT) (sec)
1	14.44	1.78	1.81
2	15.75	1.86	2.14
3	49.64	2.11	2.36
4	147.00	2.44	2.53
5	303.00	2.71	2.87
6	454.00	3.34	3.27
7	591.00	3.52	3.24
8	1860.00	4.56	4.79
9	3650.00	5.73	4.81
10	8340.00	9.33	8.16

The results of the encryption and decryption of Solo App v2.1 on Tecno Pouvoir 3 is as shown in Table 3.

Table 3. Tecno Pouvoir 3

S/N	File Size (KB)	Encryption Time (ET) (sec)	Decryption Time (DT) (sec)
1	14.44	1.97	1.94
2	15.75	2.01	2.00
3	49.64	2.18	2.31
4	147.00	2.93	3.10
5	303.00	2.79	2.88
6	454.00	3.07	3.18
7	591.00	4.66	3.32
8	1860.00	6.18	4.93
9	3650.00	6.48	6.12
10	8340.00	10.59	8.11

The results of the encryption and decryption of Solo App v2.1 on Infinix S5 Lite is as shown in Table 4.

Table 4. Infinix S5 Lite

S/N	File Size (KB)	Encryption Time (ET) (sec)	Decryption Time (DT) (sec)
1	14.44	1.50	1.48
2	15.75	1.52	1.74
3	49.64	1.94	2.52
4	147.00	2.24	2.19
5	303.00	2.64	2.48
6	454.00	2.71	3.30
7	591.00	3.18	2.63
8	1860.00	3.91	3.49
9	3650.00	5.30	4.88
10	8340.00	7.62	6.44

9. Conclusion

Cloud computing provides a way out for constrained resources on smartphones due to its unlimited resources that can be accessed through the Internet. But the major challenge is security because user accesses cloud through insecure channel and in addition cloud usage has its own challenges. However, this work had implemented AES by the development of a secure app (Solo App v2.1) to provide security for user data on Android smartphones. The developed mobile application encrypts user data on user smartphone before sending to cloud for storage. The android application is now available on Google Play Store for general use and relevant feedbacks in form of online reviews that can further be used to improve the application. The application had been tested on three android devices and the results when compared with existing ones are found satisfactory.

Acknowledgement

This work is fully funded by Tertiary Education Trust Fund (TETFund), Nigeria [TETFUND/DESS/FCES/OYO/RG/2018].

References

[1]. Ahvanooey, M., Li,Q.,Rabbani, M. & Rajput, A. (2017). A Survey on smartphones security: Software vulnerabilities, malware and attacks. *Int. J. Adv. Computer Science Application*,8(10), 30-45.

[2]. Altuwaijri, H. & Ghouzali, S. (2018). Android data storage security: *A review Journal of King Saud University-Computer and Information Sciences*, 1-10. Doi.org/10.1016/j.jkscui. 2018.07.004.

[3]. Android Enterprise Security White Paper (2020). Android enterprise security white paper updated January 2020. https://static.googleusercontent.com/media/www.android.com

[4]. /en//static/2016/pdfs/enterprise/Android_Enterprise_Security_White_Paper_2019.pdf, 1-39.

[5]. Anthony, T. V., Toby, J. V. & Robert, E. (2009). *Cloud computing: A practical approach*. USA: The Mc Graw-Hill Companies.

[6]. Borde, B. & Nigrel, S. (2016). Understanding application security in Android. *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, 5(6), 1768-1772.

[7]. Buschkes, D., Kesdogan, R. & Reichl, P. (1998). How to increase security in mobile networks by anomaly detection. *Proceedings of the Computer Security Applications Conference*, Phoenix, 3-12.

[8]. Cimler, R., Matyska, J., Balík, L., Horalek, J. & Sobeslav, V. (2015). Security issues of mobile application using cloud computing. In: Abraham A., Krömer P., Snasel V. (eds). *Afro-European Conference for Industrial Advancement. Advances in Intelligent Systems and Computing*, 334. Springer, 347-357. doi:10.1007/978-3-319-13572-4_29.

[9]. Diksha, K., Vijay, B. & Sudhir, S. (2014). Security mechanisms for smartphones: survey. *International Journal of Applied Engineering Research and Development (IJAERD)*, 4(2), 49-56.

[10]. Donald, A. C., Oli, S. A. & Arockiam, L. (2013). Mobile cloud security issues and challenges: a perspective. *International Journal of Engineering and Innovative Technology (IJEIT)*, 3(1), 401-406.

[11]. Eldos, T. & Qasim, R. A. (2013). On the performance of the gravitational search algorithm. *International Journal of Advanced Computer Science and Applications*, 4(8), 74-78.

[12]. Enck, W., Ongtang, M. & McDaniel, P. (2009). Understanding android security, *IEEE Security & Privacy*, 1540-7993/09/$25.00, 50-57.

[13]. Erich, D. & Cliff, C. Z. (2013). Denial of convenience attack to smartphones using a fake Wi-Fi access point. *IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 11-14 January, 164-170.

[14]. Fang, Z., Han, W. & Li, Y. (2014). Permission based android security: issues and countermeasures. *ELSEVIER Computer & Security*, 1-14. http://dx.doi.org/ 10.1016/j.cose.2014.02.007.

[15]. GAO (2012). *General accounting office, information security: Better implementation of controls for mobile devices should be encouraged.* http://www. gao.gov/ assets/650/648519.pdf.

[16]. Hayajney, A. A., Bhuiyan, Z. A. & McAndrew, I. (2020). Security of broadcast authentication for cloud-enabled wireless medical sensor devices in 5G networks. *Computer and Information Science*, 13(2), 13-26.

[17]. Hp (2012) *Smartphone security reports on types of security attacks in smartphones*. http://www.hp.com.

[18].IDC (2016). Smartphone OS Market Share, www.idc.com, 2016. http://www.idc.com/ prodserver/smartphone-os-market-share.jsp

[19]. Janal, R. (2014). A survey of cyber-attack detection strategies. *International Journal of security and Its Applications,* 8(1), 247-256.

[20]. Kandalkar, B. R. & Bartere, M. M. (2016). Review paper on an android application sandbox system. *International Journal of Engineering Science and Computing,* 6(6), 6865-6866.

[21]. Kawabata, H., Isohara, T., Kani, J. & Agematsu, H. (2013). SanAdBox: sandboxing third party advertising libraries in a mobile application. *IEEE ICC 2013-Communication and Information Systems Security Symposium*, 978-1-4673-3122-7/13/$31.00, 2150-2154.

[22]. La Polla, M., Martinelli, F. & Sgandura, D. (2013). A survey on security for mobile devices. *IEEE Communications Surveys and Tutorials*, 15(1), 446-471.

[23]. Mandeep, S. & Kanwalvir, S. D. (2013). Securing RJSON data between middleware and smartphones through Java scripts based cryptographic algorithms. *International Journal of Soft Computing and Engineering*, 3(2), 189-194.

[24]. Narudin, F. A., Feizollah, A., Anuar, N. B. & Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. *Journal of Soft Computing, Springer*, 20, 343-357.

[25]. Nissim, N., Moskovitch, R., BarAd, O., Rokach, L. & Elovici, Y. (2016). ALDROID: efficient update of android anti-virus software using designated active learning methods. *Journal of Knowledge and Information Systems-Springer,* 1-39.

[26].OWASP (2016). Mobile security project-OWASP, OWASP.org, 2016. https://www.owasp.org /index.php/ mobile #tab=Top 10 mobile Risks.

[27]. Poonguzhali, P., Dhanokar, P., Chaithanya, M. K. & Patil, M. U. (2016). Secure storage of data on android based devices. *International Journal of Engineering and Technology*, 8(3), 177-182.

[28]. Reddy, M. S. & Babu, Y. A. (2013). Evaluation of microblaze and implementation of AES algorithm using Spartan-3E. *International of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(7), 3341-3347.

[29]. Singh, R. (2014). An overview of android operating system and its security features', *International Journal of Engineering and Applications*, 4(2), 519-521.

[30]. Smart, N. P. (2016). Information security and cryptography. Switzerland: *Springer International publishing.* doi 10.1007/978-3-319-21936-3.

[31]. Tekade, P. S. & Shelke, C. J. (2014). A survey on different attacks on mobile devices and its security. *International Journal of Application or Innovation in Engineering and Management*, 3(2), 247-251.

[32]. Tiwari, M., Srivastava, A. K. & Gupta, N. (2013). Review on android and smartphone security. *Research Journal of Computer and Information Technology Sciences,* 1(6), 12-19.

[33]. Wang, J. & Song, J. (2016). Function optimization and parameter performance analysis based on gravitational search algorithm. *Journal of Algorithms,* 9(3), 1-13.

[34]. Yang, T., Cui, H., Niu, S. & Zhang, P. (2015). An analysis on sensitive data passive leakage in android applications. *IEEE 16th International Conference on Communication Technology (ICCT)*.

[35]. Zameshkumar, J. B. & Vijay, S. G. (2014). A study on security for mobile devices. *International Journal of Research in Advent Technology,* 2(4), 196 - 203.

