Novel Techniques to Enhance the Security of Smartphone Applications

https://doi.org/10.3991/ijim.v10i4.5869

Muneer Ahmad Dar[1] and Javed Parvez[2]
[1] National Institute of Electronics and Information Technology (NIELIT), Srinagar, India
[2] University of Kashmir, Srinagar

Abstract—**Smart phones have already become an important part of our lives. Smartphone is in the hands of millions of novice users who are unaware of the security concerns they may face. In order to address the security concerns of the millions of users, we propose and implement three novel techniques to enhance the security of these Smartphone's. The first Technique is Need based Security (NBS) wherein we take away the flexibility from the programmer and give the control to the users who can decide whether to permit the application to access any of the resources or not. In our second Technique we used the security API which will take care of everything in background and in our third technique we locked all the applications so that they can be restricted from doing any malicious activity. We tried to find out the security loopholes in one of the leading Smartphone operating system i.e. Android with the intension to apply the novel techniques proposed and implemented in this paper.**

Index Terms—**Android, Application, Malicious Activity, NBS, Security API, Smartphone**

I. INTRODUCTION

The vision of Google's Director Andy Robin who once said, we should completely replace our personal computer with the smart phones i.e. our smart phones should be capable of doing everything that our PC can do. With this vision in mind the smart phones have almost completely replaced the desktops. Millions of users worldwide are using their smart phones to do wide range of activities ranging from social networking to internet banking. The popularity of these hand held devices is because of the vast range of applications also called as apps. The applications of these apps from medical consultation to online chatting and wide range of applications which is beyond the scope of this paper [1-7]

From the security perspective it is pertinent to mention that the inbuilt security provided by Apple and Google to their respective operating systems is adequate to secure their users. But the openness of the smart phone operating system, particularly Android has given additional flexibility to their programmers to develop a malicious code which can compromise the security of the novice user. The Antivirus software's are also not very effective to check the malicious behavior of the app. It is because of the platform limitations that the app can't check the file system and any other directory of any other app. Thus the security provided by the Android to their end users is no adequate and needs to be upgraded. The operating system of Android being open source can't be modified as that may lead to some usability issues, as such we propose and implement three novel techniques which can be used to enhance the security of Android based smart phone operating systems.

This research mainly focuses on the additional security that can be given to the novice users so that their security and privacy is protected. The next section elaborates the existing security framework of Android to come up with its advantages and disadvantages. In section 3 we try to find out the existing research done in this field. In section 4 we propose three novel techniques to enhance the security of Android users. In Section 5 we implement the proposed techniques and finally we draw our conclusion in section 6.

II. EXISTING SECURITY OF ANDROID

Android is a Linux based mobile operating system having its build in security features to safeguard its users. The security architecture of Android is given in Figure 1.

Being the open source mobile platform, the Android security framework has some advantages and disadvantage listed in the subsection below.

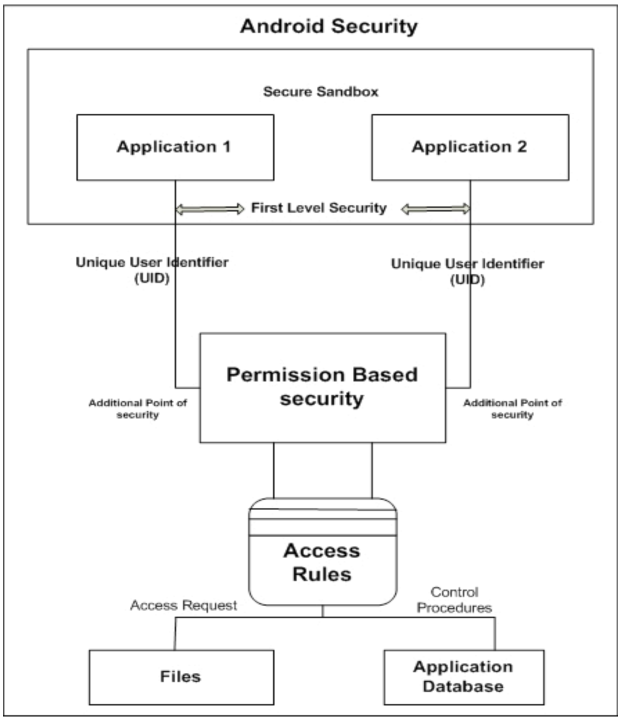


Figure 1. Existing Security Framework of Android

A. Advantages of Existing Security of Android

The Android smart phone operating system being an open source platform is the most popular and leading smart phone operating system. The major advantages of the existing security framework of Android are as under [8-13].

- Unlike Desktop operating system where all the apps share the same UID, Android apps are partitioned from each other and they work under unique UIDs. This partitioning of these apps make them more secure as no other app can access the data or files of any other app and it has to be done by the programmer explicitly.
- The Android operating system provides a two layer security. First by restricting them in their unique user identifiers and second by the permission based security. The permission based security is a core mechanism in Android to restrict the access of the resources. The user is given a set of permissions at the time of installation. If the user is technically aware he/she can judge the appropriateness of these permissions and this makes the users aware of the privacy issues that he/she may face.

B. Disadvantages of Android Security Framework

The openness of Android makes it more popular, but at the same time makes it more vulnerable to the malware attacks as the programmers are free to write anything and a novice users security can get compromised. Some of the major security concerns in Android are as under.

- The user while installing the Android app is forced to accept all the permissions in order to successfully install the application. The user can't accept one and reject other permission as this can abort the installation.
- Programmers are given freedom to write any type of code in Android. They can ask for n number of permissions to achieve a particular functionality which could have been achieved by little permission.
- The novice users are not able to understand the appropriateness of the permissions they are asked and they hardly care about these permissions. With the result the novice users are in constant threat of getting his privacy and security compromised.

III. EARLIER RESEARCH

Pridgen & Wallach [14] examined over one lac Android applications and they came to the conclusion that the number of permissions in each app is much higher. With the result the privacy of users is compromised as their more unnecessary permissions which can make the privacy of users vulnerable. In order to earn from the apps with the help of advertisements, the programmers are increasing the number of permissions so that they get the resources from the app. This was researched by Dietz, Shekhar & Wallach [15]. Felt et al [16] and Kelley et al [17] tried to find out the users behavior while installing the apps and they concluded that the novice users are unaware of the permissions they are accepting at the time of installation. Kern & Sametinger [18] gave more freedom to the users by giving them the freedom to select or reject the permissions at the time of installation. The proposal to use the monitor app and the target app was given by Berthome et al [19].

Our research focuses on the incorporation of novel techniques which not only enhances the existing security of the Android users but also does not alter the existing security framework of Android, thereby providing a second layer security to the users of these smart phones.

IV. PROPOSED NOVEL TECHNIQUES

The existing security of Android is not capable of handling the privacy issues of users particularly the novice users who are unaware of the permissions they are granting at the time of installation. With this intension to give extra freedom to our users so that they can feel more secure, we propose three novel techniques to safeguard these users.

A. Technique 1: Need Based Security (NBS)

In this method we use the reverse engineering to extract all the files from the .APK file called the Android Application Package. The process involves the Extraction or decompilation, modification of the source code and recompilation. The objective is to remove the unnecessary permissions. For example if we extract the APK file of one of the Android app called Prince of Persia, which is a gamming app available on Google Market. After extraction we will get a file called AndroidMenifest.xml, which lists all the permissions given by the programmer while creating the app. After going through the permissions we may encounter some extra or unnecessary permissions for example if this app asks for MANAGE_ACCOUNTS permission, which is not in its domain and can be dangerous for our security, we remove such type of permissions from the file.

Once we remove the extra permissions, we give extra freedom to our users by giving them the permission screen at the time of execution which tells the user that this app is trying to access the resource. The user may accept or reject that request at the time of execution.

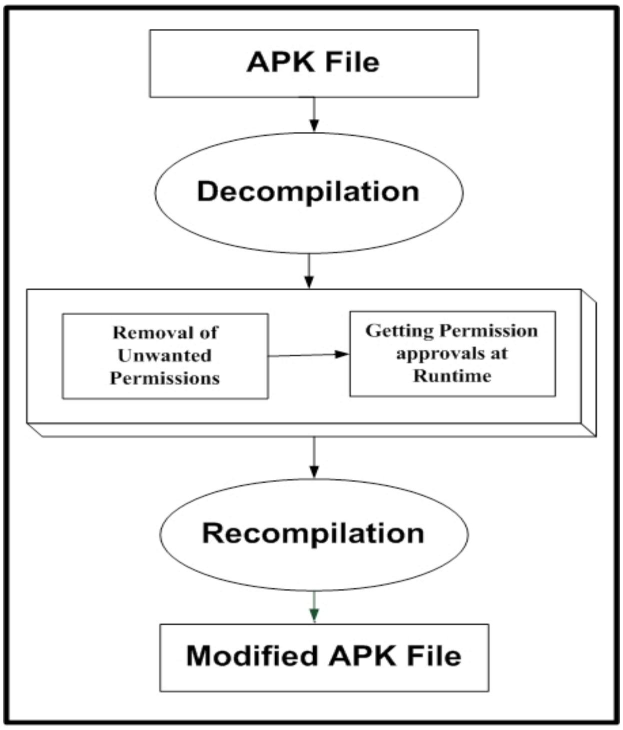

APK File
Decompilation
Removal of Unwanted Permissions
Getting Permission approvals at Runtime
Recompilation
Modified APK File

Figure 2. Architecture of NBS System

B. *Technique 2: Security check using API*

The objective of this technique is to give a user the additional freedom to accept all the permissions while installing the app. The user is not disturbed in any noticeable way and the security API takes care of all the privacy issues of the user at the back end.

C. *Technique 3: RD&P System*

In this technique we created an app called the controller app that will take care of the run time activates of all the apps . The apps are controlled by our app called the controller app. If the app tries to access any of the resources in the background without informing the user, the controller app immediately informs the user about this malicious activity.

V. IMPLEMENTATION

A. *Technique 1: Need Based Security (NBS)*

The technique is incorporated using the following three steps.

- Reverse Engineering
- Removal of extra/unnecessary permissions
- Runtime approval

The objective of reverse engineering is to extract all the files in the .APK file which includes the class files and one of the important file called AndroidMenifest.xml. This file holds all the permissions the app is asking at the time of installation.

The extraction of various components from the APK file is done using the apktool , classes_dextojar.src and other files which we can download from the internet. The sequence of commands like “apktool if file_Name.apk” and “apktool d File_Name.apk path_output” are to be executed. After successfully executing the commands, the different files from the APK file can be extracted.

After extracting the various components of .APK file we can check the applicability of the app and whether the permissions are relevant or not. For example if the app is gaming app let’s say CHESS app and if this app is trying to access the EXTERNAL_STORAGE which is not in its domain so we can remove that permission and any other such type of permission.

The final step that we incorporated is to thoroughly check the application code and enclose the block of the code which is giving access to the resource with a condition that the resources should only get accessed if the user allows the permission.

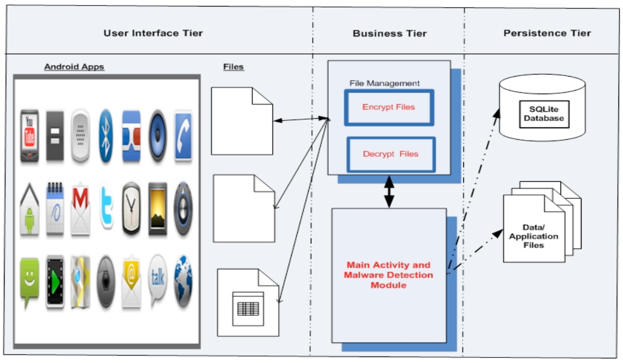


Figure 3. Architecture of Security API

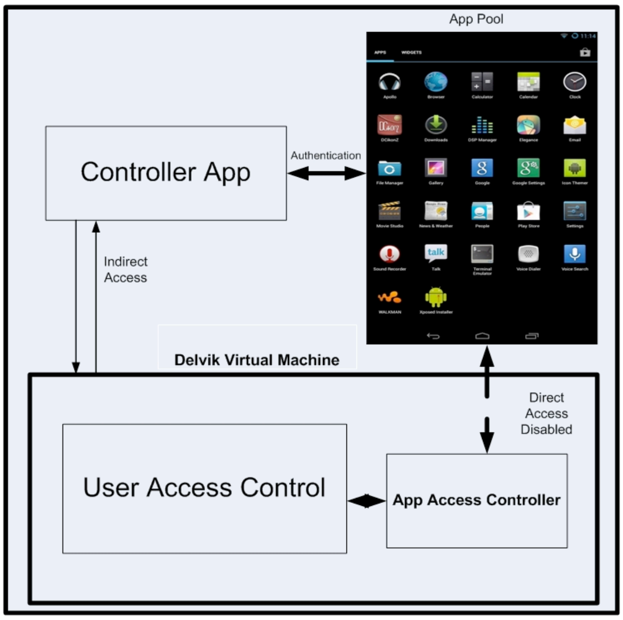
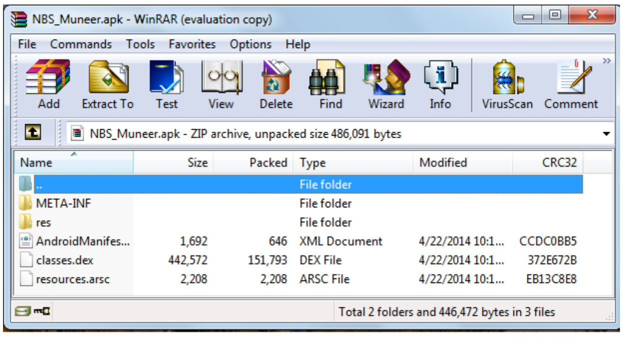


Figure 4. Architecture of RD&P System

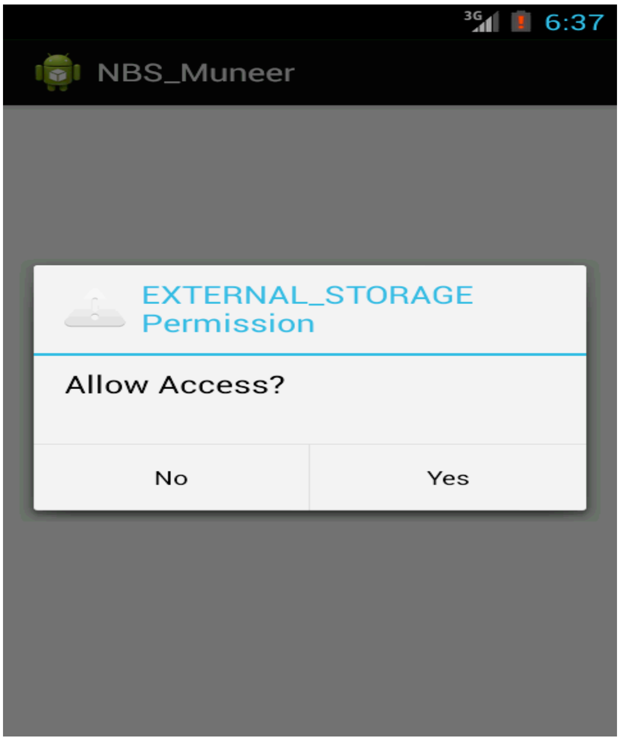


NBS_Muneer.apk - ZIP archive, unpacked size 486,091 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
META-INF			File folder		
res			File folder		
AndroidManifes...	1,692	646	XML Document	4/22/2014 10:1...	CCDC0BB5
classes.dex	442,572	151,793	DEX File	4/22/2014 10:1...	372E672B
resources.arsc	2,208	2,208	ARSC File	4/22/2014 10:1...	EB13C8E8

Total 2 folders and 446,472 bytes in 3 files

Figure 5. Contents of the APK file.



EXTERNAL_STORAGE Permission

Allow Access?

Figure 6. Permission Screen

B. *Technique 2: Security check using API*

The objective of this API is to encrypt the important file in our Smartphone If the app tries to access these file which are encrypted the user will be informed and the app can be checked for any malware. The access to these files can be blocked by the user at the time of execution. The user of this API is not informed in any noticeable way. User is informed only when the resource is getting accessed. The detailed flow of the events are described in the below figures.

C. *Technique 3: RD&P System*

The implementation of the RD&P system is done by locking all the apps which are critical as far as the security of the user is concerned. After locking the app, if the app tries to do any malicious activity it has to go through the user. i.e. the user will be informed and a password activity will be launched. The screen shot of the main activity is as under.

VI. CONCLUSION

Smart phones are going to outnumber the world total population by 2017. We use smart phones in everyday activities, ranging from social networking to GPS location searching and doing transitions using the internet banking. We tried to find out the security concerns of the novice users on one of the world's leading smart phone operating system, with the intension to incorporate three novel techniques proposed and implemented in this paper. Any one of the three techniques can be incorporated to the Android device and the extra level of security can be provided to the users. Without modifying the Android operating system which could have resulted some usability issues, we implemented all the techniques without interfering with the existing security framework of Android.

REFERENCES

[1] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt, "Identity, location, disease and more: Inferring your secrets from android public resources," in Proceedings of 20th ACM Conference on Computer and Communications Security (CCS), Nov. 2013. http://dx.doi.org/10.1145/2508859.2516661

[2] M. A. Dar and J. Parvez, "Smartphone operating systems: Evaluation & enhancements," IEEE Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on, Kanyakumari, 2014, pp. 734-738.

[3] Janice C. Sipior , Burke T. Ward & Linda Volonino (2014) Privacy Concerns Associated with Smartphone Use, Journal of Internet Commerce, 13:3-4, 177-193, http://dx.doi.org/10.1080/15332861.2014.947902

[4] S. Grzonkowski, A. Mosquera, L. Aouad and D. Morss, "Smartphone Security: An overview of emerging threats.," in IEEE Consumer Electronics Magazine, vol. 3, no. 4, pp. 40-44, Oct. 2014. http://dx.doi.org/10.1109/MCE.2014.2340211

[5] Q. A. Chen, Z. Qian, and Z. M. Mao, "Peeking into your app without actually seeing it: Ui state inference and novel android attacks," in 23rd USENIX Security Symposium (USENIX Security 14). San Diego, CA: USENIX Association, Aug. 2014, pp. 1037–1052.

[6] R. Schlegel, K. Zhang, X. yong Zhou, M. Intwala, A. Kapadia, and X. Wang, "Soundcomber: A stealthy and context-aware sound trojan for smartphones." in NDSS. The Internet Society, 2011.

[7] L. Cai and H. Chen, "Touchlogger: inferring keystrokes on touch screen from smartphone motion," in Proceedings of the 6th USENIX conference on Hot topics in security, ser. HotSec'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 9–9.

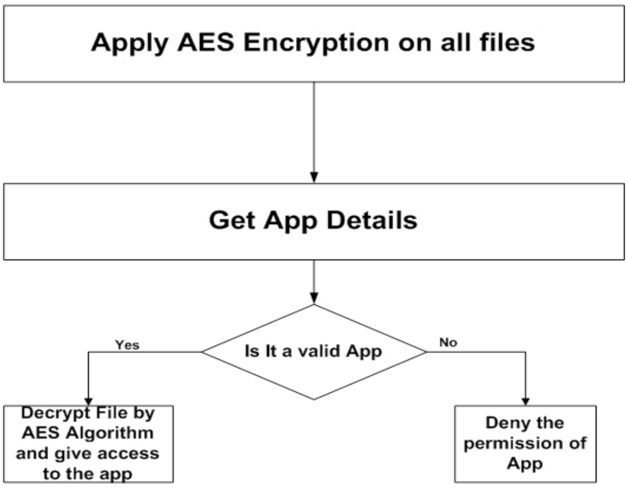

Apply AES Encryption on all files
Get App Details
Is It a valid App
Yes
No
Decrypt File by AES Algorithm and give access to the app
Deny the permission of App

Figure 7. System Flow

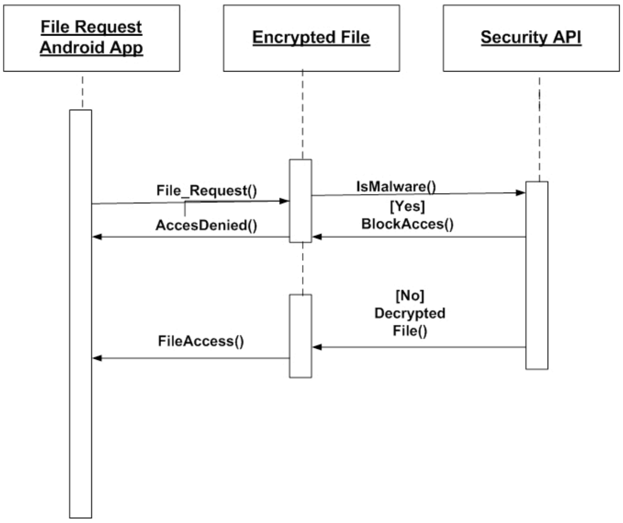


Figure 8. Sequence Diagram of the System

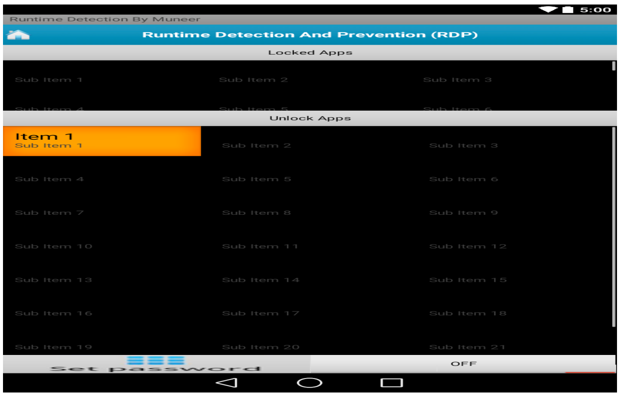


Figure 9. Snapshot of the Main Activity

[8] S. Jana and V. Shmatikov, "Memento: Learning secrets from process footprints," in Proceedings of the 2012 IEEE Symposium on Security and Privacy, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 143–157. http://dx.doi.org/10.1109/SP.2012.19

[9] M. Naveed, X. Zhou, S. Demetriou, X. Wang, and C. A. Gunter, "Inside job: Understanding and mitigating the threat of external device misbonding on android," 2014.

[10] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in 23rd USENIX Security

Symposium (USENIX Security 14). San Diego, CA: USENIX Association, Aug. 2014, pp. 1053–1067

[11] T. Book, A. Pridgen & DS. Wallach, "Longitudinal Analysis of Android Ad Library Permissions", arXiv preprint arXiv:1303.0857, 2013.

[12] S. Shekhar, M. Dietz & D.S. Wallach, "Adsplit:Separating smartphone advertising from application",CoRR, abs/1202.4030, 2013.

[13] A.P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin & D.Wagner, "Android permissions: User attention, comprehension, and behavior", SOUPS 2012, p. 3. http://dx.doi.org/10.1145/2335356.2335360

[14] Book T, Pridgen A, Wallach D S, "LongitudinalAnalysis of Android Ad Library Permissions", arXivpreprint arXiv:1303.0857, 2013.

[15] Shekhar S, Dietz M, Wallach D S, "Adsplit:Separating smartphone advertising from application",CoRR, abs/1202.4030, 2013

[16] Felt A P, Ha E, Egelman S, Haney A, Chin E, Wagner D, "Android permissions: User attention,comprehension, and behavior", SOUPS 2012, pp. 3.

[17] Kelley P G, Consolvo S, Cranor L F, Jung J, Sadeh N, Wetherall D, "A Conundrum of Permissions:Installing Applications on an Android Smartphone",Financial Cryptography and Data Security 2012, pp. 68-79.

[18] Kern MSametinger J, "Permission Tracking inAndroid", UBICOMM 2012, pp. 14855.

[19] Felt A P, Ha E, Egelman S, Haney A, Chin E, Wagner D. Android permissions: User attention, comprehension, and behavior. In SOUPS 2012.

AUTHORS

Muneer Ahmad Dar has completed his Bachelor's degree in science from university of Kashmir and M.C.A from the same university. He completed his M.Phil from Madurai Kamaraj University and is pursuing P.hD. from University of Kashmir. He has participated in more than 15 national and international conferences and has published various papers in some reputed journals. He is also the member of IETE. His areas of research are information Security, Smart phone Security, Data Mining and Algorithms He has served as Assistant Professor at different Government Colleges in Kashmir and currently is working as Scientist-B at National Institute of Electronics & Information Technology (NIELIT) J&K which is the department under Deity, Govt. of India (E-Mail: muneer@nielit.gov.in)

Dr. Javed Parvez has served as an Assistant Professor with P.G Department of Computer Science, University of Kashmir since 2002. He received B.E. (Electrical & Electronics Engg.) from BITS, Pilani (INDIA) and M.S. (Computer Science) from University of Oklahoma(USA). He also received Ph.D.(Computer Science) from the University of Kashmir(INDIA). His areas of research interest include the Security, Reliability and Performance of Computer and Mobile Communication Networks. Before joining our department he served in the R&D divisions of technology companies such as Epson, Synopsys, and Qualcomm & Ericsson. He has taught several subjects at the MCA level including C/C++ programming, Data Structures, Software Engineering, Data Communications, Computer Networks and has introduced and taught elective subjects such as Wireless/Mobile Communications. In addition he has taught and delivered lectures on interdisciplinary subjects such as Computer Viruses & Ecological Modeling. He is actively involved in guiding (M.Phil and Ph.D) research scholars & has published 32 research papers related to his fields of interest. He is a member of the IEEE, ACM and Computer Society of India(CSI).

Submitted 24 May 2016. Published as resubmitted by the authors 23 August 2016.