BringOWN - AN AGILE BYOD SOLUTION WITH DEVICE FRIENDLY CIPHERING

Warnakulasuriya Krishan Tharaka Thisera, Dilan Bandara, Peshala Vithana, Sahan Shavindrika

Sri Lanka Institute of Information Technology

B263, Malabe 10115, Sri Lanka.

krishees@yahoo.com, dilanchathurangamlz@gmail.com, peshalawith@gmail.com

***Abstract*— Recently, BYOD or Bring Your Own Device has become one of the most popular model for enterprises to provide mobility and flexibility in workplaces. The emergence of new technologies and features of mobile devices makes them integral parts of every aspect of daily business activities. In BYOD, the personal devices can be used to increase employees' satisfaction and reduce an organization's device costs. However, due to attacks and vulnerabilities it is difficult to trust the personal devices coming into the workplace. There is security concern to protect sensitive corporate data, protecting sensitive corporate data is a major concern in the industry. This project proposal proposes a BYOD model which can be used to overcome the mention problems /vulnerabilities. The BYOD model is a combination of a light weight.**

***Keywords*—BYOD, Android, Software Security, VPN, FTP, Encryption.**

I. INTRODUCTION

As employees quietly bring their personal devices to work forward-thinking companies respond, this win-win workplace revolution presents some challenges. An integrated "bring your own device" concept helps maximize the benefits while minimize the risk. Motivations to adopt BYOD go deeper than just "everyone is doing it." Some of the most convincing reasons that companies choose to allow BYOD -- or full-on embrace it -- are that BYOD can save money and it helps companies evolve with the landscape of IT. And BYOD improves employee satisfaction. [1]

There is no unique approach to implement a BYOD concept so there are various technics and practices can be found in today's world. But how much of them implemented with proper security and how far it would be secured.

BYOD solutions vary depending on the specific needs, abilities, and security concerns of companies.

Unlike traditional corporate-issued laptops, which can be protected by full disk encryption techniques, mobile devices are more diverse in their individual platform capabilities. In addition, at most enterprises, most mobile devices that are brought forward for business purposes are personally owned, and this presents unique challenges. For personally owned enterprise-connected devices, it is important to preserve the user experience as it related to personal us while providing a containerized enterprise experience as well as the additional layers of authentication that the enterprise imposes.

Communication should be take place in a well secured manner. As well as the data which transfer to the mobile device, the main file server stays secure and also available 3to the valid user request. All the communication should be done very secure manner.

File and user authentication all the users registered, user should be conduct their communications with the main server based on session. Also it must to authenticate user and also his request by authentication servers.

Acceptable Wi-Fi range, in this implementation this will provide as a feature to Wi-Fi network. In above discussion the all the activities happened inside the company premises but when it comes to Wi-Fi network it is not a good practice to consider the whole organization Wi-Fi area as a company premises. So it's better to define a Wi-Fi range to perform the activities mentioned earlier due to that all the user activities are limited to defined Wi-Fi area (A virtual boundary).

A. OBJECTIVES

The main objective is to transform the organizations' network to support bring your own device (BYOD) and deliver exceptional user experiences which requires the correct enterprise and support throughout the development cycle by providing security, encryption within a smart phone application. The objectives associated with "any" BYOD implementation vary from cost savings, attracting employees, innovation, work-life balance to employee retention and many more. Objectives/Requirements that the proposed BYOD implementation.

II. METHODOLOGY

If it's all about the security of the cooperate data, the implementation should be addressed,

- what if it's required to grant access to the employees on the cooperate data vaults with very accountable manner enriched with the proper security, how it should be implemented?
- If there is a rouge employee who is waiting to take advantage of cooperate data by selling them or may be publishing them in public. how it should be implemented to keep those organizational files only available to the employee only if when they are inside the company?

Having the cooperate data with the employee in the employee device, obviously employee have to bear the burden of the responsibility, how to mitigate that burden of the responsibility by transferring it to a third party.

Based on those issues following barrios will be introduced,

1. How to securely communicate with company server?
2. How to Protect both company and user's personal data when they were downloaded in to the employee device?
3. How to provide accountability?
4. What happen if the device has stolen?
5. If it's an android application which user's interface to the fileserver, how to provide the integrity plus security on it?

A Perform Secure Communication

There are plenty of technics and practices to achieve this requirement. As example it can be used

- ➢ HTTPS instead of HTTP.
- ➢ SFTP or FTPS instead of FTP
- ➢ SCP
- ➢ VPN

Communications which are taking place inside the encrypted tunnel is secure and more effective because inside the VPN tunnel it is possible to use most of the protocol.

1) HTTPS servers

HTTPS provide transport layer security with using SSL, further using SSL certificates issued to the both parties by valid certificate authority (CA) can encrypt the data traffic and ensure the security of the data. But to achieve that HTTPS server with the valid CA is required. The considerable disadvantage is that HTTPS users are bounded to one particular protocol (HTTPS).

2) SCP/SFTP/FTPS Protocols

WinSCP one of the most popular client application to transfer the data between SCP/SFTP enabled devices by using SCP or SFTP protocols. Unlike FTP SFTP/SCP traffic is encrypted along with user credentials due to that an untrusted third party is not accessible to the data. But like HTTPS, SCP/SFTP users are bounded to one particular protocol (SCP/SFTP).

3) VPN servers

It is more effective if it's possible to use all of the traffic which mentioned earlier inside a one single framework. Then encrypted tunneling comes in to play, Encrypt the all the traffic between Source and destination without bound to a one particular protocol (Both HTTP and FTP traffic can be accommodated by VPN server).

Let's have a look at most common VPN solutions,

a. PPTP VPN

Most common Point-To-Point tunneling protocol which also known as less secure but Most of the devices embedded with the PPTP client. But PPTP is old and vulnerable so above mentioned advantages are not worth.

b. L2TP/IPsec

The Layer 2 Tunneling Protocol (L2TP) was has its origins in PPTP. Since it does not provide security features such as encryption or strong authentication it is typically combined with IPsec. L2TP/IPsec is theoretically secure, but there are some concerns. It's easy to set up, but has trouble getting around firewalls and isn't as efficient as OpenVPN.

c. OpenVPN

OpenVPN based on open-source technologies like the OpenSSL encryption library and SSL v3/TLS v1 protocols. OpenVPN can be configured to any port unlike L2TP/IPsec. But only disadvantage is unlikely PPTP and L2TP/IPsec, OpenVPN users have to use third party client application to access the server. OpenVPN can configured to authenticate by using,

- key pairs and certificates
- username + password

Specially in TLS, VPN server has,

- Its own key
- An issued certificate
- The CA certificate

Further every client should have CA certificate and also client can be authenticated by using their own certificate, User credential or 2-factor authentication which both at once as a form.

B. File server/Transfer and Client functions

BringOWN uses samba as its main server aka BringOWN fileserver. Samba is a freely available server which provide interoperability between Unix/Linux server and windows base clients.

As all the transactions maintain under BringOWN VPN network all the transaction conducts through BringOWN backbone is safe. As both SCP and SFTP suffer from performance problems in the underlying openssh, BringOWN uses FTP/S. In FTP/S aka FTP over SSL, allows for the encryption of both the Control and Data Connections either concurrently or independently. as it files transfer protocol.

To connect with BringOWN samba server BringOWN android application has in built FTP/S client.

BringOWN end users have access to BringOWN file server only through BringOWN back bone itself. As an example BringOWN user can't use their credential, which use to access BringOWN backbone to access BringOWN file server alone.

C. Securing Downloaded Files

If user has successfully log in to the system using the given android application and assume that he/she requires to download some cooperate documents from the BringOWN file server, (note that all the data transferring and authentications happened under the mentioned VPN network) where to keep those particular files and how to secure those particular files.

1) Folder types in android

As BringOWN solutions are based on android platforms, it better to discuss some basic storage options available on android platforms. In here it'll discuss only related options to the implementation (Download and store files in runtime).

a. External Storage

Almost all the Android platforms supports with shared "external storage" that it can use to save data on the device, which can be either removable storage media or a device's internal storage aka non-removable media are stored in the external storage and also called world-readable. Data resides on these areas can be modified by a third party.

The files those are stored under this category bares the cost of low security because those files are world-readable.

b. App specific directory

The files store under this category, are not intended for other apps to use (such as graphic textures or sound effects used by only the particular app).

Storing files under this category is far more secure than External Storage category because files store under app specific category only access through the application and also when user uninstall the app those particular app specific files will be deleted along with the uninstallation process. But there is a threat on this category if user grants the root privileges on his device, these files are easily accessible.

An example to get the directory path to the App specific directory

```
String packName = getPackageName();
File dirc = getFilesDir();
String location=dirc.getAbsolutePath()
```

c. "/res" Resource Directory

For arbitrary raw asset files. These files are processed by aapt and must be referenced from the application using a resource identifier in the R class.

Followed by the above options mentioned, if cooperate files are only downloadable to the app specific directory these cooperate files would be maintaining more secure manner.

2) Encryption Avoid the Problems if phone has rooted

As mentioned earlier there is a threat, if user grant root access to the device he/she can easily access those cooperate files in the BringOWN app specific director,

To mitigate that risk,

- Can invent a company policy that those BYOD's should not be rooted.
- If by any chance if user grant the root access on the device, there should be a way to secure those files.

In following section, it will be discussed what happen if a user grants the root permission on their particular android device?

Even if a user grants the root privilege on the device and compromised the cooperate data, what if he can't read that cooperate data on the device, the data still keep as secured. The best way to achieve this is to perform encryption on the cooperate data.

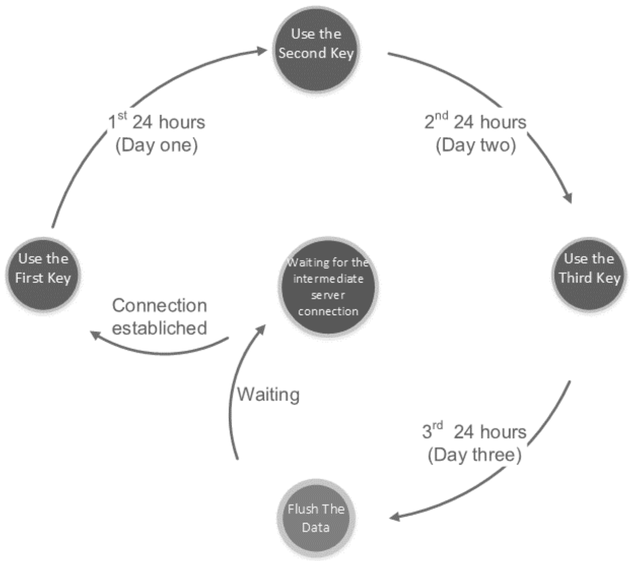
a. Encryption using Light weighted algorithms

As these encryption functions are performing on the mobile devices those encryption functions should be device friendly while securing the data. There are few concerns when it's selecting proper encryption function. First it should strong enough to secure the data, second it should be a less resource consumption algorithm and it is not practical to use an encryption algorithm which is fresh to cryptography as a best practice.

DES	• DES is a standard. The actual algorithm used is also called DES or sometimes DEA (Digital Encryption Algorithm). • DES is now considered insecure (mainly due to a small key size of 56-bits). • Triple DES (3DES) applies the DES algorithm thrice and thus has better practical security. It has 3 keys of 56-bits each (applied to each pass of DES/ DEA). • DES-X is another variant. • DES is a block cipher.
AES	• Supported by TrueCrypt, SSH. • AES and Rjindael are block ciphers. • AES can operate in many modes. AES-GCM (It is fast and secure and works similar to stream ciphers. Can achieve high speeds on low hardware too. Only supported on TLS 1.2 and above.) AES-CBC
Blowfish	• Patent free. In public domain. • Supported by SSH. • Much faster than DES and IDEA but not as fast as RC4.

	• Uses variable size keys of 32 to 448 bits. Considered secure. Designed for fast CPUs, now slower / older CPUs. • Blowfish is a block cipher.
Twofish	• Was one of the finalists in the AES competition • Most CPUs now include hardware AES support making it very fast than Twofish. • Patent free. In public domain. • Uses keys of size 128, 192, or 256 bits. Designed to be more flexible than Blowfish (in terms of hardware requirements). • Supported by TrueCrypt, SSH. • Twofish is a block cipher.

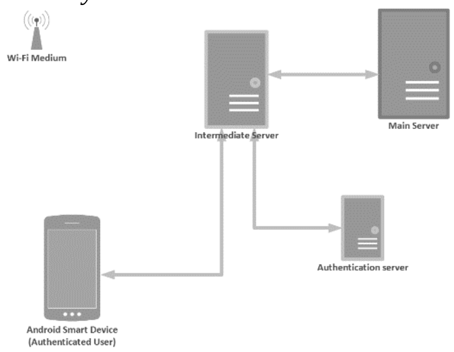
b. *Encryption Key cycle*



When it comes to encryption one of the key factor is how to manage the keys because weak keys are vulnerable to brute force attacks. Frequently renewing of encryption keys reduce this vulnerability. As an example if a light weighted encryption algorithm could break within two days using brute forcing, but if that encryption key changes in every 24 hours it makes it nearly unbreakable using brute forcing.

In this contest BringOWN defines a cycle of encryptions (BringOWN encryption key cycle) assuming that an employee only allowed to stay away from BringOWN networks for 72 hours.

c. *Encryption Key escrow*

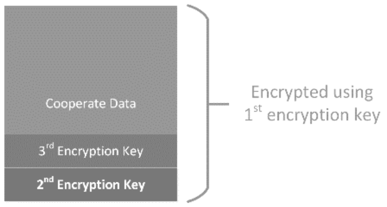


Allowing encryption management responsibility to end user's device (in this contest to the android device) will enable the vulnerabilities which leads cooperate data to a risk. Due to that all the key management functions perform on the BringOWN intermediate server.

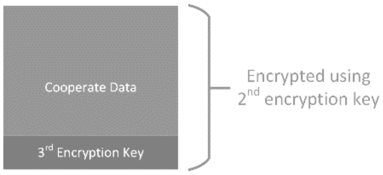
For the encryption, BringOWN intermediate server pushes the three subsets of key and stores inside the app-specific folder (Which has limited access only to the BringOWN application).

Here is the data structure inside the android device related to each encryption phase,

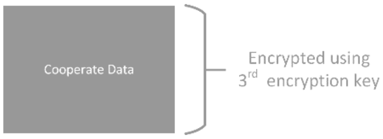
Day one,



Day two,



Day three



d. *Encryption Triggering*

Encryptions triggers can be defined by factors,

- *Location Based Triggering,*

Location Based triggering which BringOWN use its proximity functions and detect the user's location based on Wi-Fi strength and Trigger the encryption/decryption functions.

- *Time Based Triggering,*

It can be defined particular time, when it should encryption/decryption perform.

- *Manual Triggering by a third party,*

Involving 3rd party such as a data administrator or some specific person and by using their permission can trigger the encryption/decryption. As an example when user need to read a file, user can connect to the authorities mentioned above and grant the temporary permission on the particular file and then read.

Considering about these factors BringOWN uses both time-based and location based encryption triggering mechanisms, but in fact BringOWN always recommended location based triggering rather than time based triggering.

D. *Perform & Maintain Android Application's Security*

BringOWN security mechanisms are not only limited to the operational level but also it covers the entire deployment of the android application. BringOWN secure the duplication of its copies with the unauthorized parties.

1. *Use of Root Detection Functions*

As described earlier BringOWN uses some ciphering technics if any case user gains root access to his android device. If user has granted the root permission BringOWN application frees itself and Delete all the cooperate files.

2. *Use of application embedded program to open the downloaded files*

Letting user to read downloaded cooperate file using third party application other than BringOWN itself will welcome plenty vulnerabilities on those files.

BringOWN Uses its own embedded application to open the downloaded cooperate files or some file like '.docx', '.ppt' will temporary broadcast using content provider functions then they can read using limited third party application.

3. *Use of stamp*

These mechanisms use to restrict user by sharing BringOWN source, aka limit the distribution of the sources.

Stamp is a pseudo random key which generates after installation of the application on the android device. When user connecting to the BringOWN back bone it's IMEI and stamp code should be related to each other.

If user shares his BringOWN apk with another employee the employee who has the duplicate 'apk' will not able to connect with BringOWN back bone.

E. *Perform & Maintain the Back End Security*

Securing the front-end aka BringOWN android application is not enough, but there should be a proper mechanism to secure the back-end which cooperate administrators are dealing on.

BringOWN uses shell scripts to register user and other back end activities. Due to those shell scripts contain database connectivity and related critical details, those shell scripts are fully encrypted.

All the data base password fields are encrypted using AES and further Relevant AES encryption keys managed by centralized encryption key file which configured to refresh every month.

F. Perform & Maintain Accounting for Users

Every BringOWN user assign to one particular privilege level depend on the organization's infrastructure. Each privilege level defined as a fileserver user.

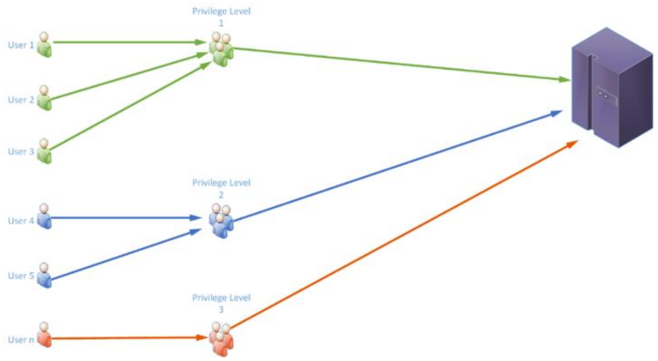


Figure II-1 BringOWN privilege Levels

BringOWN end user is always a logical file server user which means this user has no direct access to the main file server. When user logging to the system by using provided BringOWN user credential and request the file server access BringOWN will provide relevant credential to the server followed by assigned privilege level.

G. Users Authentication and Authorization

Authentication, authorization and also user accounting in BringOWN performs in its authentication servers. BringOWN uses its own authentication server to manage Authentication, Authorization and accounting.

As it has mentioned in earlier sections each user has assigned to one single privilege level further the actual privilege level meaning is a File Server user, as an example if there is a File Server user aka Samba user, call admin there can be one or more BringOWN user who refers to that admin the file server user. Due to that all the BringOWN users who follows the 'admin' privilege level will obtain same authority on the file server.

Each and every user activity on the BringOWN systems will captures and stores it the separate logging database.

H. Parameters of security

BringOWN uses following parameters in order to validate/authenticate user.

1) Use of IMEI

BringOWN Uses IMEI number which is hard to forge when user registering to the system. Device's EMEI number will pushed in to the BringOWN authentication server, each and every time when user tying to access the BringOWN backbone.

Even if user try to access the BringOWN server using provided credential but the different device it won't be permitted by the BringOWN authentication servers.

2) Use of Stamp

As described in the earlier sections stamp is pseudo random value which combined with IMEI. If user reinstall or installed application unauthorized manner, stamp will defer from original stamp which stored earlier, due to that stamp also should be match with the authentications database values.

3) Use of access point id

BringOWN end users are restricted to access the BringOWN main servers only through a predefined access point. As an example BringOWN end user who is belongs department 'A' cannot access the BringOWN main servers through the any other access point rather than Department A itself.

III. SECURED BYOD SOLUTION

A. User registration process

At first user need to download the BringOWN android application through an authorized party also it is a must that relevant user provided his employment details and android device's IMEI number to that authorized party. Then user will be provided a relevant serial number to register to the system.

The authorized party for the registration process first need add those employment details and IMEI number to the BringOWN user registration database along with serial number. To perform these activities and all the back end activities BringOWN provides secured shell-scripts.

Then user can use provided serial number along with desired username to register to the BringOWN backbone. Also note that during this process of registration user's stamp will pushed to the BringOWN Backbone.

As the final step of user registration process above mentioned authorized party should activate the particular user by using "createuser.sh" shell script. At this point that particular user required to be assigned to the relevant privilege level and allow an access point to access the BringOWN backbone.

After successful user registration user will able to access the BringOWN back bone and access the cooperate files. Note that for the initial logging user will be provided a password which should change as the first thing when user logging to the system.

B. User logging process

Registered user can log in to the system by using their usernames and passwords. If user has to reset the device, reinstall the BringOWN application or user change their device user will always have to follow the above registration process.

IV. RESULTS

B. VPN servers

As a product, BringOWN's cooperate customers always have their flexibility to choose the desired VPN server though BringOWN recommended OpenVPN as the first place. BringOWN server deployment includes one VPN server (OpenVPN, PPTP VPN or n2n VPN) deployment, for the Both n2n and OpenVPN server BringOWN cooperate customers have to purchase the relevant VPN client application from the original vendor.

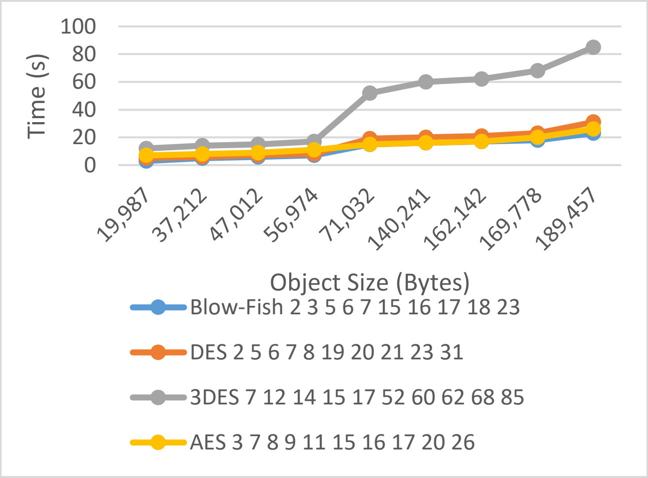
C. Resource consumption of Proposed Encryption Functionalities

Following Charts and Tables show the speed comparison of BringOWN Encryption Technologies. For the following testing it has used an implemented android java application and installed it in to a LG-G3 mobile phone with the following specification.

Chipset	Qualcomm MSM8974AC Snapdragon 801
CPU	Quad-core 2.5 GHz Krait 400
Memory	32 GB, 3 GB RAM

Object Size	Blow-Fish (s)	DES (s)	3DES(s)	AES (s)
19,987	2	2	7	3
37,212	3	5	12	7
47,012	5	6	14	8
56,974	6	7	15	9
71,032	7	8	17	11
140,241	15	19	52	15
162,142	16	20	60	16
169,778	17	21	62	17
189,457	18	23	68	20
229,151	23	31	85	26
Average Encryption Speed	10026.66071	7908.35211	2864.7602	8507.469697

Table IV-1 BringOWN encryption Speed comparison



D. Overall BringOWN performance

Overall BringOWN performance depends on the Organization network and specs of the Mobile device. Network speed will define the speed of the file downloading while the CPU and RAM of the mobile device defines the Encryption Decryption speed.

V. CONCLUSION AND FUTURE WORKS

BringOWN currently work with the third party VPN client if user use OpenVPN instead of PPTP. Due to that OpenVPN client fiction should be integrated with BringOWN android application.

As BringOWN limited to use its own embedded function to open the file which are downloaded from the cooperate servers, Apache OpenOffice is recommended to integrate with BringOWN android application.

BringOWN only provided the file read option to the end user which means there must be a separate to policy to be found in order to file edit and upload process.

VI. REFRENCE

[1]. D. Evans, "What is BYOD and why is it important?", TechRadar, 2015. [Online]. Available: http://www.techradar.com/news/computing/what-is-byod-and-why-is-it-important--1175088.

[2]. WhatIs.com, "What is BYOD (bring your own device)? - Definition from WhatIs.com", 2016. [Online]. Available: http://whatis.techtarget.com/definition/BYOD-bring-your-own-device.

[3] CISCO, “Byod and virtualization insights from the cisco

ibsg horizons study,” May 2012. [Online]. Available:

http://www.cisco.

[4]D.Jaramilo[N.Katz,B.Bodin,W.Tworek,R.smart and T.Cook], “Cooperative solutions for Bring Your Own Device (BYOD)” .

[5] S.Chung[T.Escrig and Y.Bai], “2TAC: Distributed Access Control Architecture for "Bring Your Own Device" Security,” in ASE/IEEE Int. Conf., Washington., USA, 2012, pp.123-126..

[6] S.Chung[T.Escrig and Y.Bai], “2TAC: Distributed Access Control Architecture for "Bring Your Own Device" Security,” in ASE/IEEE Int. Conf., Washington., USA, 2012, pp.123-126.

[7] G.Russelo[M.conti,B.Crispo and E.Fernandes], “MOSES:supporting operation modes on Smartphones,” in 17th ACM symposium on Access Control Models and Technologies Conf., New York., USA, 2012, pp 3-12.

[8] McAfee An Intel company, “Smarter BYOD”.

[9]S. Verma, "A new tool for lightweight encryption on android," in Advance Computing Conference (IACC), 2014 IEEE International, Gurgaon, 2014

[10]M. Eslahi, M. Naseri, H. Hashim, N. Tahir and E. Saad, "BYOD: Current state and security challenges", 2014 IEEE Symposium on Computer Applications and Industrial Electronics(ISCAIE),2014. https://www.researchgate.net/publication/261871646_BYODCurrent_State_and_Security_Challenges

[11] Y.Bai[L.Dai,S.chung and D.Devaraj], “Access control for cloud-based eHealth social networking,” in Wiley Journal of Security and Communication Networks 1st ed., 2012, pp.574–587.

[12] G.Portokalidis[P.Homburg,K.Anagnostakis and H.Bos], “Paranoid Android: versatile protection for smartphones,” in 26 th Annu.Computer Security Applications Conf., New York., USA, 2012, pp.347-356.

[13] Ruebsamen and C. Reich, “Enhancing Mobile Device Security by Security Level Integration in a Cloud Proxy,” in 3rd Int. Conf., Furtwangen., Germany, 2012, pp. 159-168.

[14] Cite This For Me. (2016). Save Time and Improve your Marks with CiteThisForMe, The No. 1 Citation Tool. [online] Available at: http://www.citethisforme.com [Accessed 2 Sep. 2016].