Keylogger Threat to the Android Mobile Banking Applications

Naziour Rahaman, Salauddin Rubel, and Ahmed Al Marouf

Abstract Android is presently the world's most prevalent operating system, reaching more mobile customers than any other operating system to date by providing numerous services via smartphone and various android devices to make our life easy. Most of the android applications are developed by third-party android developers, android provides them an enormous platform to build their application. Modern cyber attackers are highly interested in this platform to access user's sensitive information; with their own build malicious application or take amenities of other android developer's application to spy on user's activity. We have found that keyloggers can thieve personal information from users, such as credit card information or login pin/password from their typed keystroke in social networking and mobile banking apps. In case of mobile banking generally the mobile devices such as smartphones, tablets are being used for financial communications with the banks or financial institutions, by allowing clients and users to conduct a variety of transactions. In android app store (Google Play) keylogger apps are initially blocked but using some vulnerabilities in app permission it can be installed with benign and trusted apps. Both expert and maladroit android smartphone users use the mobile banking application, inexpert users are unable to find the vulnerabilities and attacker's use this as an advantage to place an attack. The security android has provided for all the application is not sufficient for the sensitive application such as mobile banking application. In our paper, we discuss how attackers steal mobile banking app users sensitive information for their financial gain and also proposed a method to avoid keylogger attacks on android mobile banking apps.

N. Rahaman · S. Rubel · A. A. Marouf (✉)
Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh
e-mail: marouf.cse@diu.edu.bd3

N. Rahaman
e-mail: naziour.cse@diu.edu.bd

S. Rubel
e-mail: salauddin15-7033@diu.edu.bd

S. Smys et al. (eds.), *Computer Networks and Inventive Communication Technologies*,
Lecture Notes on Data Engineering and Communications Technologies 75,
https://doi.org/10.1007/978-981-16-3728-5_12

Keywords Keylogger · Prevalent OS · Mobile banking · Mobile security · Financial Institute · Maladroit

1 Introduction

Cyberattacks on financial services firms have risen by 72% globally between 2014 and 2018. According to The Cost of Cyber Crime Study [1], the average cost per company increased around $1.3 million between 2017 to 2018. With over 3.7 billion smartphone users worldwide, the growth of the mobile app industry is unsurprising. More than 2.8 million android applications are in Google Play Store. About 25.75% of android user have used finance categories app in September 2019 [2].

Users, developers, and cybercriminals are all drawn to the versatility and functionality that Android has to deliver. Because of its ease of use and ability to cover high mobility, mobile banking has become a norm in the banking industry. Individuals who use mobile banking should be aware of the potential for cybercrime to affect their banking statements. Keylogger attack can be one of the criminal techniques that may occur on the mobile banking application.

Keylogger can embed themselves into PCs, Macs, Androids and iPhones in the same way like other malwares does and these types of malware called ‘rootkit’ viruses. Only in 2019, 9.9 billion malware attacks are reported along with mobile malware [3] which convey a security threat to the mobile banking application. Google Play Store has taken a wise step by removing all the keylogging application. As a result, the only option that mount a keylogger program is to do so remotely. Since we all know, Android asks for system permissions before running any app; however, often people disregard certain approvals. Since the user is ignorant to the application’s secret permissions, keyloggers take advantage of the situation. Keeping these approaches in mind, using social media analytics and insights from social media such as recommendation engines [4, 5], personality trait polls [6, 7] or the stylometric features [8] can be utilized for such attacks.

Different methods may be used to prevent such keylogger attack. Now mobile baking applications are used by all kind of people, most of the people do not have enough knowledge about security. Keylogger attack can be mostly prevented by using antivirus or antimalware software. But most of these software are paid, users have to spend money for the service. Users from poor or developing country are not interested of using paid antivirus software but these countries have the most app user.

2 Literature Review

Keylogger attack takes place from the client-side because it steals a user’s confidential and personal information from the user’s input channel. The openness of the android platform has brought out a huge privacy risk, especially on transaction-based activity.

Quite a number of works have been done on keyloggers but most of them are focused on computers' keylogger. Recently, a few keylogging attacks on mobile devices have been studied. The method of stealing data from a mobile banking app has been identified by Prayogo Kuncoro [9]. However, their research did not include any mobile banking apps and did not provide any solutions for preventing keylogging attacks. Many attacks from third-party keyboards were introduced by Fadi Mohsen in [10]. Their research mostly focused on analyzing current keyboard authorization which can be still abused. However, this information is not sufficient to prevent keylogger attack. Junsung Cho's study analyzes the security of the third-party keyboards on android system, they have proposed to use a trustworthy keyboard for the bank Web site but keyboard design is not specified [11].

To avoid and prevent keylogging attack several models has been proposed by [12–14]. These prevention methods are applicable for malicious behavior of keylogger but with advanced android, permission keylogger can still place an attack. A study of Dr. Manisha M. More has shown the current scenario of cyberattacks in online banking [15]. Dr. N. Bhalaji has shown a very useful method for secured mobile experience by replicating data method [16]. To detect suspicious activity, Dr. Joy Iong have used hybrid deep learning which is capable of detecting keylogging [17]. Their technique can detect harmful activity but prevention is no specified. We, on the other hand, showed different ways of tracking the mobile banking apps by keylogger through an intense test with a potential solution.

3 Keylogger for Mobile Banking Application

3.1 Frequently Used Permissions in Mobile Banking Application

Permissions are used by an application for getting authorized access to different components of android. Developers declare all the required permission in the AndroidManifest.xml file and the permission are granted by the user while installation. In recent android devices (Android 6.0 and above) have a runtime permission system [18]. Permissions are granted when the applications are running and also have an enough context on why permission is required. We have installed 50 mobile banking applications from 15 different countries. Total 16 permissions are requested by these applications, 50% of them require only 5 or 6 permissions. There are two types of mobile banking applications: (i) only transaction-based and (ii) transaction and lifestyle-based. Only transaction-based applications provide money transfer, ATM and balance information. On the other hand, transaction and lifestyle-based applications provide many services including mobile recharge, bill payment, shopping offers, etc. This second kind of mobile banking application requires many permissions which makes them more vulnerable. For user personalization, the developer needs to access more user's data, thus make the application a good source of user

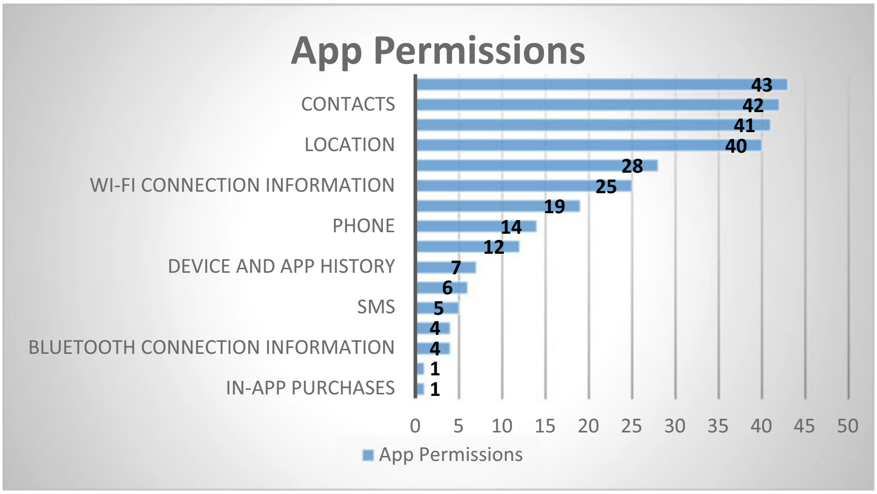


Fig. 1 Frequently used app permissions

information and others application try to take this as advantage. Problems happened when other applications having less permission, uses these sensitive permissions by corrupting ICC (Inter Component Connection) [19, 20]. Broadcast theft, activity hijacking, service hijacking, intent spoofing, privilege escalation and application collusion attack are some of the possible ways to do this [21] (Fig. 1).

3.2 How Keylogger Gets Installed

In android, keylogger app can be installed in various way with the app's apk. Consciously a user will not install a keylogger apps can get installed by either with a pop-up ad or pleasant app. After being installed, the apps began to log the user's keystrokes. The attacker saves keystroke data in a file in local storage, which is then submitted to a remote server. Advance keylogger can record and send the keystrokes to a database that is updated frequently. We demonstrate the attack scenario of a keylogger application (Fig. 2).

From the following figure, we can find three ways to install a keylogger application in android mobile.

- The attacker can create a benign application such as a calculator application with hidden keylogger malware, and it will be very difficult for a user to identify the keylogger. Even attackers' benign applications can incline users to install other keylogger application from other resources. Stalkerware is a process of installing an app without user interaction, this model can be used by attackers to install an unwanted keylogger app [22].

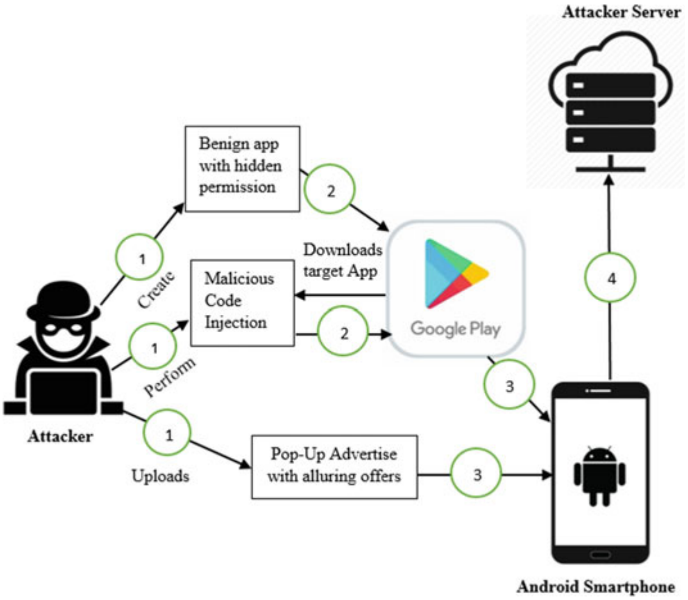


Fig. 2 Keylogger attack scenarios

- Attackers target most popular application from play store, with reverse engineering they can retrieve the code and inject their malicious code into the application [23]. Most of the time it happened for the paid application. Attackers provide the paid application for free with malicious code inside it.
- A keylogger can be installed from pop-up advertisement, attackers can provide an advertising with an alluring offer like, install this application and get 10$. Once a user installs that kind of application, constantly user gives access to any permission, thus keylogger can get installed.

4 Proposed Mechanism to Prevent Threat

We know that sensitive banking information is saved as an encrypted cipher. The client clicked the unencrypted value while entering their pin in the mobile banking app for logging. As a result, a keylogger application can easily monitor the value (Fig. 3).

Keylogger records typed value according to the sequence of typing. Each value in a keyboard layout design has two parts: labeled value and codecs value [24]. The label value is what we can see on keyboard but the value of codecs isn't really available in user interface. The ASCII value of a character is the codecs value of a key. Every key has a unique ASCII value, but every keyboard has a certain key-label and key-codecs value. The keylogger extracts the key's label value and stores it in a file. In the

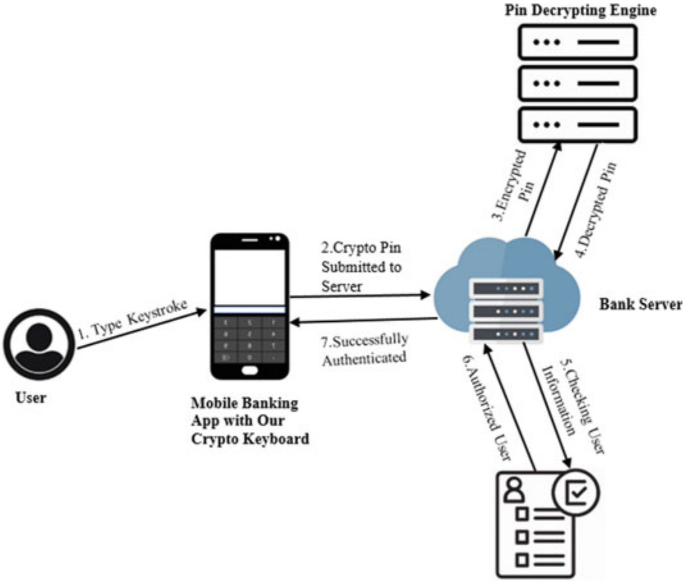

Pin Decrypting Engine
3.Encrypted Pin
4.Decrypted Pin
2.Crypto Pin Submitted to Server
Bank Server
1. Type Keystroke
7.Successfully Authenticated
User
Mobile Banking App with Our Crypto Keyboard
6.Authorized User
5.Checking User Information

Fig. 3 Our proposed model of encryption

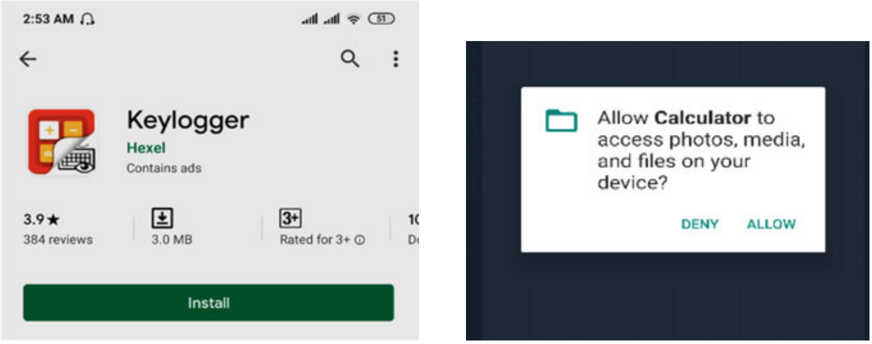
model we've suggested, the key label will be similar to the default keyboard layout. On the other hand, key codec value will be diverse. User's typed keystroke will be different from the labeled key value. The actual labeled value will not be recorded by the keylogger because of varied codec value of the key. As an Example: keylogger will store 'm' instead of recording '3' for the labeled key of '3' as we assign the codec value of '3' to the ASCII value of 'm'. The keylogger will unable to record the original key value and store encrypted value. The decryption engine will decode the encrypted value of the key and the original value will be sent to server. The encryption process will be started from the user end while typing the Pin/Password. Encrypted pin will be sent to banking server where a pin decrypting engine will decrypt the pin. Then the decrypted information will be used to authenticate user. After successful authentication user will be able to login.

5 Experimental Analysis

5.1 Proof of Concept

For our experiment, we downloaded a keylogger application from the Google Play store. Though keylogger is illegal in most of the terms but some keylogger application still available in Google Play Store. Before publishing any application in play store Google has a review process [25]. So, attackers designed their application in such way that it can be able to pass the review process. We downloaded the application named ‘calculator’ developed by Hexel. During installation, this application only requires ‘Photos and Media’ permission. Recently Google has updated the permission system, permissions are allowed by user during using the application. But not any effective warning messages were displayed for using the permission of WRITE_EXTERNAL_STORAGE, most users are unlikely to notice the significant threat of this approval and will proceed with the installation. But this application needs to enable ‘accessibility’ for the keylogging process. After enabling accessibility keylogger to start keylogging. It records all the keystrokes typed by the user and saved into a file (Fig. 4).

We run our experiment on an app named ‘Rocket’ (An application of Dutch Bangla Bank Limited) [26]. We login into the mobile banking application and after some activities we logged out of the application. In this period of time, our keylogger enabled us to store the login pin of the application in a text file which was later manually checked. Most of the mobile application has same pin as the pin for the bank account. Once the application pin accessed by an unauthorized person, the account will be in danger (Fig. 5).

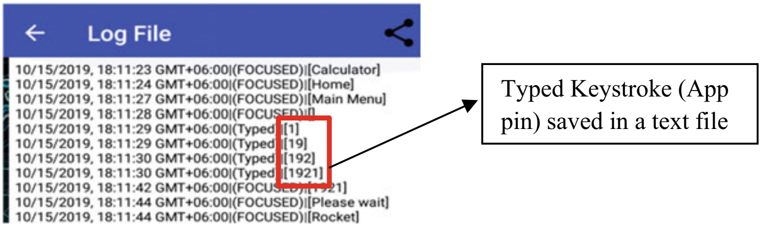


Allow **Calculator** to access photos, media, and files on your device?

(a) App Information

(b) Storage Permission

Fig. 4 No effective warning message to user mobile



```
10/15/2019, 18:11:23 GMT+06:00|(FOCUSED)|[Calculator]
10/15/2019, 18:11:24 GMT+06:00|(FOCUSED)|[Home]
10/15/2019, 18:11:27 GMT+06:00|(FOCUSED)|[Main Menu]
10/15/2019, 18:11:28 GMT+06:00|(FOCUSED)|[]
10/15/2019, 18:11:29 GMT+06:00|(Typed)|[1]
10/15/2019, 18:11:29 GMT+06:00|(Typed)|[19]
10/15/2019, 18:11:30 GMT+06:00|(Typed)|[192]
10/15/2019, 18:11:30 GMT+06:00|(Typed)|[1921]
10/15/2019, 18:11:42 GMT+06:00|(FOCUSED)|[1921]
10/15/2019, 18:11:44 GMT+06:00|(FOCUSED)|[Please wait]
10/15/2019, 18:11:44 GMT+06:00|(FOCUSED)|[Rocket]
```

Typed Keystroke (App pin) saved in a text file

Fig. 5 Keystroke typed by Android built-in keyboard (Gboard)

From the above figure, we can find the recorded keystrokes. Keystrokes are also tokenized in the segment so that the application login pin can be easily found.

5.2 Applying Our Mechanism

From our proposed mechanism (Sect. 4) we design a prevention method to avoid keylogging attack in mobile banking applications. We have implemented our mechanism with the same mobile banking application and keylogger application. We face some problems in implementing single digit encryption. In our present android application programming, we cannot implement a function in key codes. That's why encryption from the key codecs cannot apply. As android is controlled by Google, so changes in android are dependent on them. In this scenario, we just change the codecs value with encrypted value and record the keystroke (Fig. 6).

To prevent keylogging attack, we have designed an encryption method for the keyboard. For implementing our mechanism of encrypting keystroke digit from the user end this model can be used. This algorithm will encrypt the typed keystroke in a single digit, our proposed methodology is Single Digit Key Encryption (SDKE) which is an application of AES (Advanced Encryption Standard) encryption algorithm [27]. User will type pin/password for login into his account, this pin will be encrypted as AES cipher. Each digit will be encrypted and save into a stack. From the encrypted cipher, a digit will be choosen randomly and this random digit will be the codec value for the digit. If any keylogger able to steal the keystroke, actually it will record this random digit. When user press the submit button for login, all the

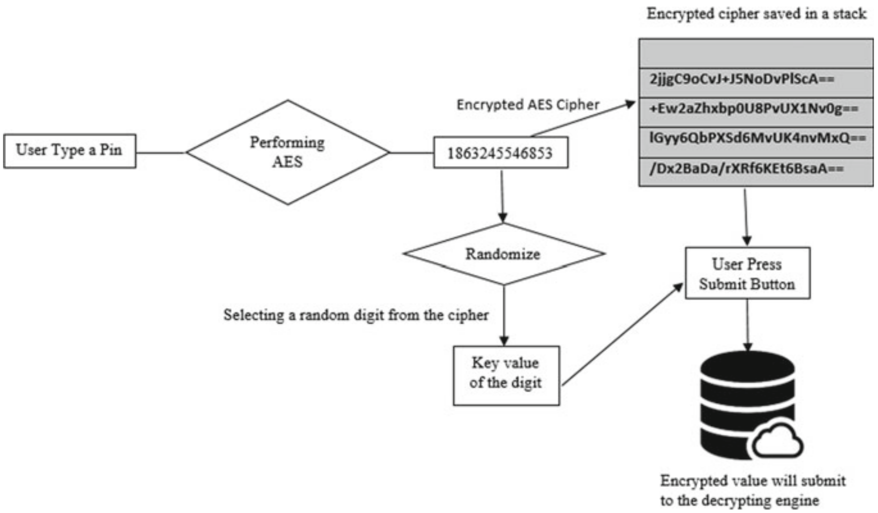
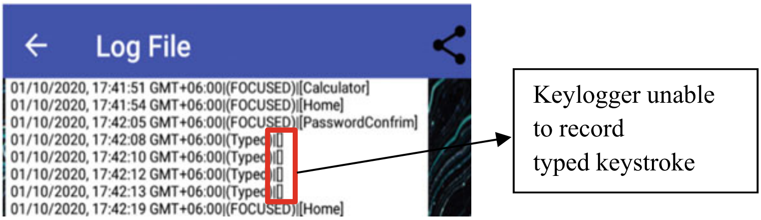


Fig. 6 Applying Pin-Crypto method to avoid keylogger attack



```
01/10/2020, 17:41:51 GMT+06:00|(FOCUSED)|[Calculator]
01/10/2020, 17:41:54 GMT+06:00|(FOCUSED)|[Home]
01/10/2020, 17:42:05 GMT+06:00|(FOCUSED)|[PasswordConfrim]
01/10/2020, 17:42:08 GMT+06:00|(Typed)|[]
01/10/2020, 17:42:10 GMT+06:00|(Typed)|[]
01/10/2020, 17:42:12 GMT+06:00|(Typed)|[]
01/10/2020, 17:42:13 GMT+06:00|(Typed)|[]
01/10/2020, 17:42:19 GMT+06:00|(FOCUSED)|[Home]
```

Keylogger unable
to record
typed keystroke

Fig. 7 Keystroke typed by our designed keyboard with our recommended settings

encrypted cipher from the stack will be submitted to the decrypting engine along with the key value. Then the decrypting engine will decrypt the cipher and process for the next authentication task. Using this model keylogger unable to record the actual value. Suppose user type the pin ‘5678’ for login after performing 128-bit AES with the secret key value of ‘1,863,245,546,853,697’ this will return some cipher for ‘5’ like ‘2jjgC9oCvJ+J5NoDvPlScA==’ [28] and from this cipher a random digit ‘j’ will be choose. So if the user type ‘5’ keylogger will receive ‘j’ for the typed keystrokes (Fig. 7).

From the above figure, we can find that the keylogger application unable to store the keystrokes. The keylogger record the codecs value which we have changed into an encrypted value. So, the value is untraceable by the keylogger. If any keylogger is able to record the typed keystroke, it cannot find the original key value. Because the value is already encrypted. In that case, it can record the encrypted value. Example: user type a pin ‘1234’, the keylogger record encrypted value like ‘h&)*’. Its method ensures more integrity of a mobile banking application.

6 Recommendation

Application login Pin length should be more than 4 Digit: Password length is a very important issue in the security of any online account. Having password length equal to or less than 4 digits lays a platform for brute-force attack. In mobile banking application, there are many applications which are using only 4 digits for application login. Though a small password is easy to remember but it is less secure. Using 5 digits is 6 times (30,240) more and using 6 digits is 30 times (151,200) more secure than 4 digits pin in a mobile banking application.

Password field input type change from number pin to text password: Existing mobile banking applications’ password field is designed for the number input type. We are talking about digit encryption, when we want to encrypt any digit in numberPassword field, the encrypted digit will be also a number digit. Encrypting any number digit to another number digit does not effective as there is a chance of having same input and encrypted digit. Using text type password field will provide access to a large number of Unicode characters. Then the encryption combination will be huge. In a password field, the user cannot see the typed digit so the changes may not affect the user.

Disable Copy-Paste option in the password field: To avoid collecting the application pin, this function can be effective. Most of the mobile banking application is using this function.

Mobile Banking application should have their own keyboard layout: Using the android's default or third-party keyboard for mobile banking application can be dangerous. Mobile banking application should use their own keyboard layout for their application.

7 Conclusion and Future Work

The paper presented the potential keylogger threat to Android Mobile Banking App. Keylogger attack may not happen if a user is conscious while using their smartphone or have a little security knowledge. The Trojan that delivers keylogger can drop more malware such as adware, spyware, ransomware or even a legacy virus on the system. So, it is not only about keylogger attack but also other malware that can take place in an android phone. In our proposed method we use AES algorithm which may slow down the overall process of user logging. We need to find more efficient and optimized algorithm for this process and specifically for digit encryption. This method can be only implemented when we can able to use functions in android xml file which provides the codecs value for the key.

For our future work, we will expand our keyboard model and build a fully cryptographic keyboard to ensure proper security. During our experimental analysis, we have found that keylogger can also read messages from Facebook, Gmail and WhatsApp. We will find more efficient algorithm and design for the keyboard that can be used to stop or prevent keylogging attack.

References

1. Help Net Security: Financial services firms most adept at making balanced security investments—Help Net Security, 2020 [Online]. https://www.helpnetsecurity.com/2018/02/14/financial-services-security-investments
2. Statistica: Leading Android App Categories Worldwide 2019, 2020 [Online]. https://www.statista.com/statistics/200855/favourite-smartphone-app-categories-by-share-of-smartphone-users/
3. Securitymagazine.com, 2020 [Online]. https://www.securitymagazine.com/articles/91660-more-than-99-billion-malware-attacks-recorded-in-2019
4. Marouf, A.A., Ajwad, R., Tanbin Rahid Kyser, M.: Community recommendation approach for social networking sites based on mining rules. In: 2nd IEEE International Conference on Electrical and Information and Communication Technology (iCEEiCT), Jahangirnagar University, Bangladesh, 21–23 June, 2015

5. Mehedi Hasan, M., Shaon, N.H., Marouf, A.A., Kamrul Hasan, M., Mahmud, H., Mohiuddin Khan, M.: Friend recommendation framework for social networking sites using user's online behavior. In: 18th IEEE International Conference on Computer and Information Technology (ICCIT), MIST, Bangladesh, 21–23 December, 2015
6. Marouf, A.A., Kamrul Hasan, M., Mahmud, H.: Comparative analysis of feature selection algorithms for computational personality prediction from social media. IEEE Trans. Comput. Soc. Syst. **7**(3), 587–599 (2020)
7. Marouf, A.A., Kamrul Hasan, M., Mahmud, H.: Identifying neuroticism from user generated content of social media based on psycholinguistic cues. In: 2019 2nd IEEE Conference on Electrical, Computer and Communication Engineering (ECCE 2019), CUET, 7–9 Feb, 2019
8. Hossain, R., Marouf, A.A.: BanglaMusicStylo: a stylometric dataset of bangla music lyrics. In: 1st IEEE International Conference on Bangla Speech and Language Processing (ICBSLP), SUST, 21–22 Sept 2018
9. Kuncoro, A., Kusuma, B.: Keylogger ıs a hacking technique that allows threatening ınformation on mobile banking user. In: 2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE), 2018
10. Mohsen, F., Shehab, M.: Android keylogging threat. In: Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2013
11. Cho, J., Cho, G., Kim, H.: Keyboard or keylogger?: a security analysis of third-party keyboards on android. In: 2015 13th Annual Conference on Privacy, Security and Trust (PST), 2015
12. Enck, W., et al.: TaintDroid. ACM Trans. Comput. Syst. **32**(2), 1–29 (2014)
13. Nauman, M., Khan, S., Zhang, X.: Apex. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security—ASIACCS'10, 2010
14. Pearce, P., Felt, A., Nunez, G., Wagner, D.: AdDroid. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security—ASIACCS'12, 2012
15. More, D.M.M., Nalawade, M.P.J.D.K.: Online banking and cyber attacks: the current scenario. Int. J. Adv. Res. Comput. Sci. Softw. Eng. Res. Paper, 2015
16. Bhalaji, N.: Effıcıent and secure data utilization in mobıle edge computing by data replication. J. ISMAC **2**(1), 1–12 (2020)
17. Chen, D., Smys, S.: Social multimedia security and suspicious activity detection in SDN using hybrid deep learning technique, vol. 2, no. 2, pp. 108–115 (2020)
18. Google Play|Android Developers: Android Developers, 2020 [Online]. https://developer.android.com/distribute/best-practices/develop/runtime-permissions
19. Li, L., Bartel, A., Klein, J., Traon, Y.: Automatically exploiting potential component leaks in android applications. In: 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, 2014
20. Schartner, P., Bürger, S.: Attacking Android's Intent Processing and First Steps Towards Protecting it. Technical Report TR-syssec-12-01, Universität Klagenfurt, 2012
21. Wang, J., Wu, H.: Android Inter-App Communication Threats, Solutions, and Challenges. arXiv:1803.05039, 2018
22. Baraniuk, C.: The rise of stalkerware. New Scientist **244**(3257), 20–21 (2019)
23. RSAC: Reverse-Engineering an Android App in Five Minutes. PCMAG, 2020 [Online]. Available https://www.pcmag.com/news/rsac-reverse-engineering-an-android-app-in-five-minutes
24. Keyboard|Android Developers: Android Developers, 2020 [Online]. https://developer.android.com/reference/android/inputmethodservice/Keyboard
25. Google Play|Android Developers: Android Developers, 2020 [Online]. https://developer.android.com/distribute/best-practices/launch/launch-checklist
26. Play.google.com, 2021 [Online]. https://play.google.com/store/apps/details?id=com.dbbl.mbs.apps.main&hl=en&gl=US

27. Search Security: What is Advanced Encryption Standard (AES)? Definition from WhatIs.com, 2020 [Online]. https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard
28. Online Tool for AES Encryption and Decryption. devglan, 2020 [Online]. https://www.devglan.com/online-tools/aes-encryption-decryption