An algorithm for improvement of email security on android operating system in the era of industry 4.0.

Isaac Moses Kisembo [1], Gilbert Gilibrays Ocen[1]*, Ocident Bongomin[2], Andrew Egwar Alunyu[1], Ildephonse Nibikora[3], Davis Matovu[1] and Felix Bwire[1]

[1] Department of Computer Engineering & Informatics, Faculty of Engineering, Busitema University, P o Box 236, Tororo, Uganda

[2] Department of Manufacturing, Industrial and Textile Engineering, School of Engineering, Moi University, P.O. Box 3900-30100, Eldoret, Kenya

[3] Department of Polymer, Industrial and Textile Engineering, Faculty of Engineering, Busitema University, P.O. Box 236, Tororo, Uganda

* Correspondence: Gilbert G. Ocen; gocen@eng.busitema.ac.ug

Abstract: The world is attesting a tremendous change today which is remarkably coined as industry 4.0. Several terminologies have developed as a result of the emergency of industry 4.0, notably is cybersecurity which entails the security of communication and network operations activities either on or offline and the measures taken to achieve such security. The most common form of communication by organizations and Business today is the electronic mails (Email), although email is a valuable tool, it also creates security challenges when not properly managed. There is a growing adoption of email as official form of communication in many organizations with majority of users on mobile android devices due to the popularity of the android operating systems and the proliferation of mobile devices. Banks, health care, educational institutions and many other service providers are communicating to their clients through email where sensitive and confidential information are shared. One major threat to email communication is lack of confidentiality for emails accessed via android mobile devices due to weaknesses of android operating system (OS) platform that presents possibilities to penetrate by hackers and android email client since it accepts a onetime login and password authentication which is only required again if the email account is deleted from the android mobile device. In this study, an algorithm was designed and implemented on an android application that allows an email sender to compose an email and set the time the email will stay in the receiver inbox before it automatically wipes off. Primary data was collected from email users using tightly structured questionnaires and respondents comprised of those with email technical background and those that are typical email users inorder to get their opinion on the lack of confidentiality on the android mobile email client, while secondary data from scholarly journals and articles informed the study design. The designed algorithm was tested and evaluated through expert opinion. The result of the study indicates that the designed algorithm addresses the confidentiality issues and threats on android email clients.

Keywords: Cybersecurity, Industry 4.0, Android, Operating System, Algorithm, SWOT Analysis

1. Introduction

Android is today's most popular mobile operating system for both smartphones and tablets, this popularity creates many risks which are not fully recognized [1]. As technology continues to evolve, so also do the opportunities and challenges it provides [2]. The increased use of technology puts society at a crossroads as it moves from a society already entwined with the Internet to the emergency of industry 4.0 characterized by automation, Big Data, and the Internet of Things (IoT) [3, 4]. The proliferation of mobile devices and their adaption of usage by both businesses and individuals as a means of communication presents new form of concerns [5]. The automation of many business processes which are being adapted to technology has forced people to depend on such technologies for communication and transactions. Just as technology brings ever greater benefits, it also brings

ever greater threats: by the very nature of the opportunities, it presents, makes it become a pivotal point for cybercrime, industrial espionage, and cyberattacks. Therefore, protecting the communication mechanism such as email of organisations and businesses becomes paramount priority [6].

Email is the electronic communication protocol par excellence used on a daily basis by hundreds of millions of people, as well as by most governments and businesses across the world [7]. The email ecosystem is a highly interoperable one and relies on a core set of protocols initially designed more than three decades ago, in an early digital context much different from the one found today in terms of digital privacy and security risks [8]. The Electronic mail (email) is perhaps the most popularly used system for exchanging business information over the Internet (or any other computer network). At the most basic level, the email process can be divided into two principal components: (1) mail servers, which are hosts that deliver, forward, and store email; and (2) mail clients, which interface with users and allow users to read, compose, send, and store email [9].

With the massive adoption of internet and email communications, a new rich set of complementary standards and tools were created in order to tackle the growing security and privacy concerns, however, these enhanced protocols and tools have failed in practice to deliver an effective protection [7]. As a result, world-wide email communications remain largely vulnerable to security and privacy threats [8]. Some researchers have suggested encrypting and signing emails in order to secure it [10–12]. This further complicates this means of information exchange since it places a greater load on the organization's network infrastructure, may complicate malware scanning and email content filtering, and often requires significant administrative overhead [13]. However, for many organisations the benefits of email encryption and signatures will outweigh the costs [9].

A security module that can protect the mobile device (and the user) against malicious communication, unauthorized access to resources and user private data, and against other security threats includes a combination of features such as; control of third-party applications, validation of the SMS sender's number, protection against fake contact name of the SMS sender, collection of data about fraudulent and spam SMS messages, robust sending of SOS SMSs and SOS e-mails with geographic coordinates of the mobile device, verification of validity of the base station, deletion of user data from a mobile device remotely, locking of a phone until the password is entered and filtering calls and SMS messages [12-15].

However, the popularity of mobile phones and growing number of applications and different useful features like call features, calculations, maps, applications for sending and receiving money, paying bills , email communication and many others [16] coupled with the easy accessibility of android applications from "Play Store" with the advantage of easy developer registration and distribution has made many "ill-intended" developers to take advantage of such characteristics to implant malware in Android applications which causes severe the damages [17].

With the increasing number of Android devices and users, it is imperative to prepare security measures for its communication applications such as emails [17]. Moreover, even the most hardened system can be breached through social engineering – the 'hacking' of people, or user innocently clicking on an email link. A recent study by researchers at the Friedrich-Alexander University of Erlangen-Nuremberg, Germany, revealed that just over 50% of people click on links in emails from strangers, even when they were aware of the risks.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0126.v1

2. Methodology

The present study was conducted in three steps or phases as illustrated in Figure 1.

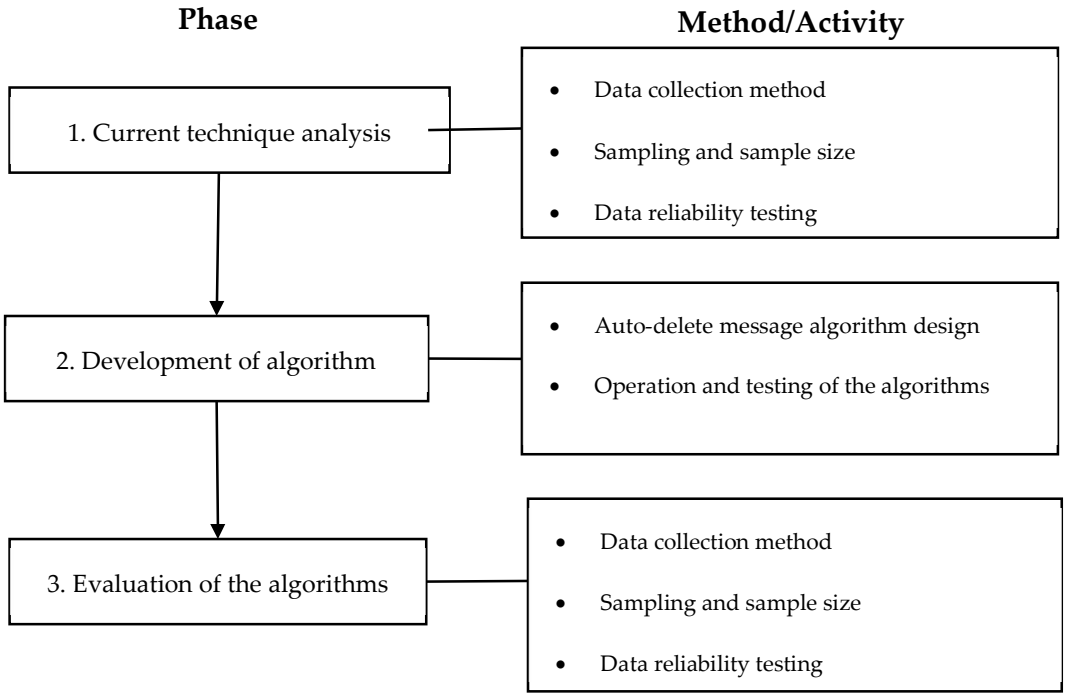

Figure 1: Methodology Approach.

Literature on specific email security techniques were reviewed in phase one, in phase two the algorithm was developed and in phase three the algorithm was evaluated through by use of questionnaires to selected participants and SWOT analysis was done.

2.1. Analysis of Techniques used in this Study

The goal of going to the field was to assess whether the current techniques are failing in protecting mails in the inbox. A closed questionnaire was used to collect the opinion of the users. This is because the researchers wanted the users to limit their opinion only to the subject being researched. The category of users that are literate to the concept of security and confidentiality selection required experience, frequent email users and those users who have adopted emails on their android email clients were selected for this study and three institutions were randomly selected for this study. Both the technical and non-technical email users from each institution were considered. Since they were few in number, the whole population was taken for the study. In this study a non-probability sampling was used in order to ensure that the samples are all frequent email users. This method allowed a sample that is knowledgeable of email usage, and are themselves email users, and have been using emails to communicate sensitive information, had significant experience of over 10years. Purposive sampling technique was used to get knowledgeable respondents and these were drawn from three companies, Luzira Prisons, Neptune Software Group and UGAFODE Microfinance Limited because these organisations had users with significant experience in Email usage and sharing of confidential information through Emails.

Using the data in Table 1, we came up with the sample size that was used to make precise generalizations with confidence for the entire population. The sample size selected addressed issues of precision that is (how close the computed estimate is to the true characteristics of the population) and confidence that is (how certain is the estimate to hold true for the population).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0126.v1

Table 1. Sample size from the Population.

Institution	Technical Email Users	Non-Technical email users	Total number of Staff
Neptune Software Group (Consultancy Department)	5	5	10
Luzira Prisons (Rehabilitation Department)	0	7	7
UGAFODE Microfinance Ltd (ICT and Risk Department)	5	10	15

This assessment was conducted using closed questionnaire submitted to selected users based on their knowledge, email adoption and how much they rely on emails for sensitive communication. The determination of internal consistency for the data collected was carried out using Cronbach's alpha test and Cronbach's alpha of 0.722 was obtained.

2.2. Develop of Data Wipe Algorithm

2.2.1. Design of message auto-delete algorithm

The class diagram describing the auto delete algorithm was designed using UML tools as illustrated in Figure 2.

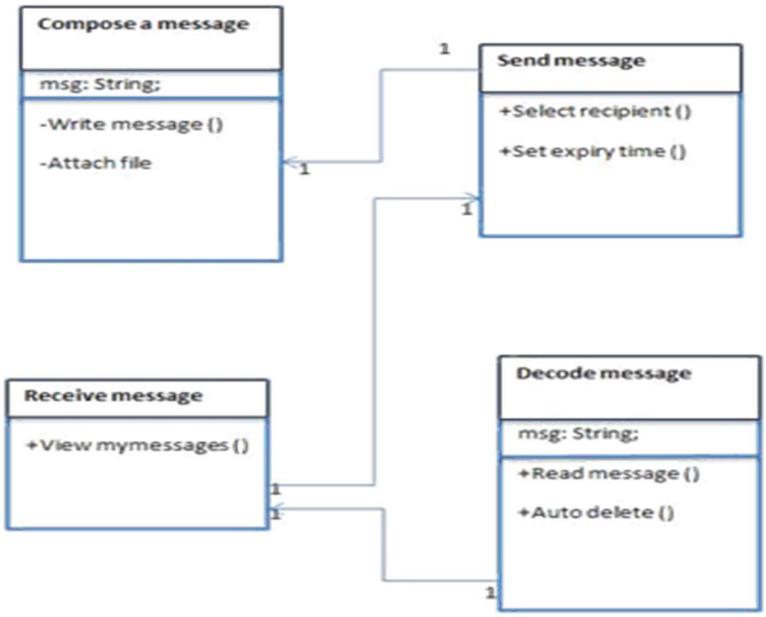

Figure 2: Class diagram of auto-delete algorithm

Being an additional feature to a given application, the data wipe algorithm that gives the message sender authority to the message, the size of the application needed to be small. This prompted us to examine the kind of loops and languages used in building the algorithm. Kotlin used in connection with android studio were chosen looking at its simplicity and few lines of code, which could be used to implement given flows of the algorithm as compared to ordinary java programming language. This would mean that instead of having say 10 lines of code function written in java, kotlin would represent it in a shorter function. For code optimisation and algorithm efficiency, Kotlin was the preferred base programming language used to develop the algorithm. The algorithm to auto delete the message was designed using Unified Modelling Language (UML) and structured programming. The pseudo code of the developed auto-delete algorithm is shown in Figure 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0126.v1

Algorithm: auto-delete message

1. Capture User attachment. (Say String attach = request.getParameter ('attachment'))
2. Capture Destination phone. (Say String recipient = request.getParameter ('recipient'))
3. Instantiate the date. (Say Date, date = new Date ()) Capture receiver's current time. (Say int timenow = date.getMinute ()) Capture Sender's sending time. (Say int timeofsendingmessage = date.getMinute ()) Capture Sender's expiry time. (Say string expirytime = request.getParameter ('expirytime'))
4. Begin loop

   IF '(time now – timeofsendingmessage) = expirytime' Delete message from database

Figure 3: Pseudo code for auto-delete email message

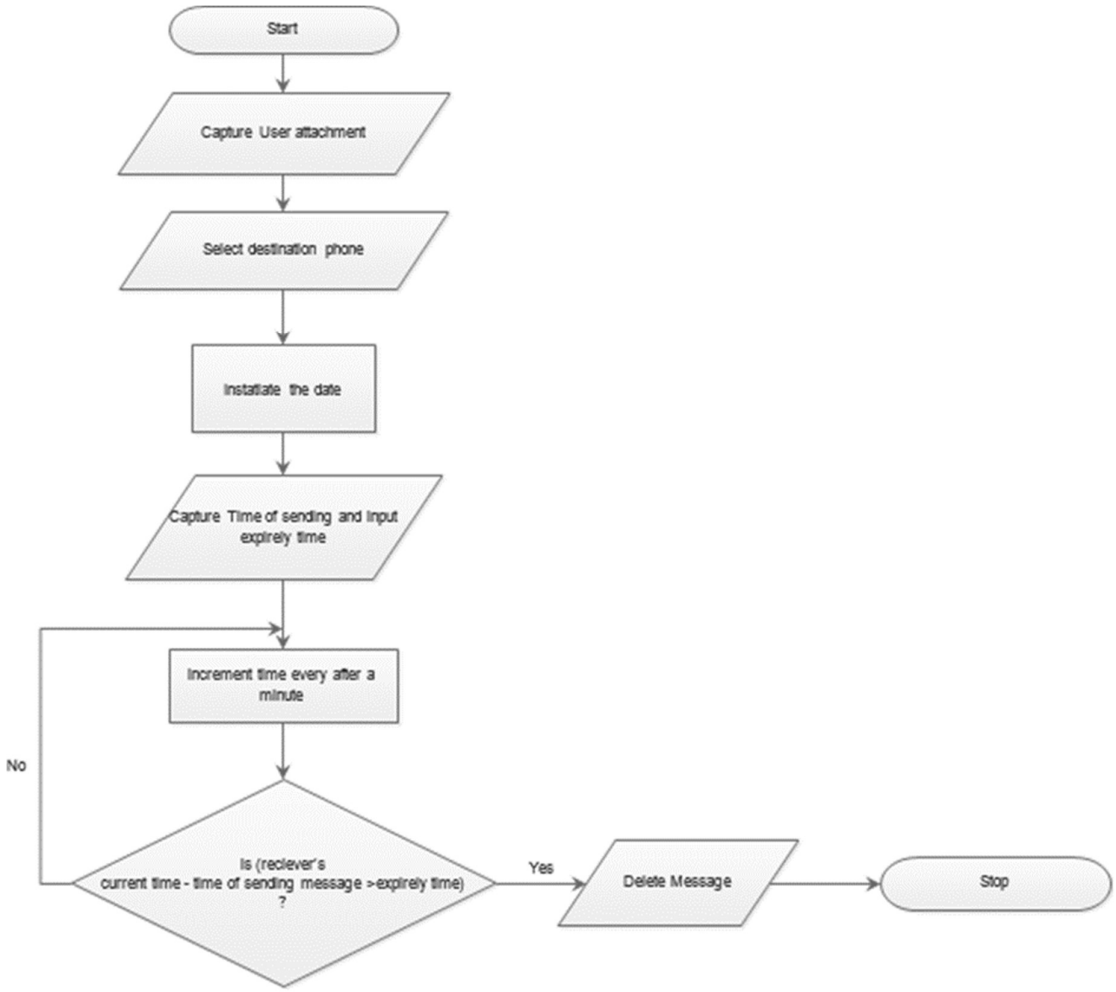

Figure 4: A flowchart showing the description of the algorithm

2.2.2. Description of the algorithms

The pseudo code algorithm for auto-delete messages in Figure 2 is clearly described as follows:
STEP 1: The algorithm captures the message or any file to send.
STEP 2: The Algorithm captures destination device IP.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0126.v1

STEP 3: Instantiate the date class and capture the sending time and the receiving time for the share message.
STEP 4: The algorithm captures the expiry time for the shared message. If receiver's current time minus the sending time is less than the expiry time. The receiver's time increments by one minute, else the message is deleted from the devices.
The operation of the designed auto-delete messages algorithm is presented in Figure 4.

2.1. Evaluation of the algorithm

Due to limited access to the email server applications API's, a customized email server and client android mobile application was developed using java and android studio to enable message composition and transmission to a given recipient and its where the auto delete algorithm was embedded. The application was made available to the users chosen and was used to evaluate if it still achieves the stated function. A questionnaire was dispatched to these users to collect their response on the use of the algorithm and the responses was used to draw conclusions of the research.

3. Results and Discussion

3.1. Field Study

The study conducted sought among others to determine the email user experience on confidentiality of email access over email clients on android platforms, the study also engaged the technical users on their opinion about the security adequacy to guarantee confidentiality of emails accessed through android mail clients. The results showed that out of 31 (thirty-one) respondents who were given the questionnaires, 100% returned valid results, 14 (fourteen) (45%) were technical users. The data collected was then categorized, quantified and then coded. Data analysis in this study was done using SPSS. The data analysis here is aimed at reaffirming the problem statement.

3.2. Reliability Testing

Reliability is the degree with which the study provides consistent results when studying a similar population [18]. Therefore, it helps to explain the degree to which an instrument measures the same way, every attempt it is employed under similar condition with the same subjects. The reliability test was run on two Constructs which came as different questionnaires and were responded to by the same group of respondents as depicted in Table 2.

Table 2. Reliability statistics

Construct	Cronbach's Alpha	Cronbach's alpha based on standardisation items	Number of items
Scenario 1	0.722	0.742	20
Scenario 2	0.702	0.701	12

Table 2 shows that all the parameters are above Cronbach's Alpha's 0.7 value which is considered acceptable for academic research. Regarding each of the research questions (R1 to R12), all evaluation questions were positively set and results shows all questions (strongly Agree or Agree) gave an average of 65% as illustrated in Figure 5 implying that the majority of the respondents agree with the algorithm provided and have recommended the algorithm as a better solution to ensure confidentiality on the e-mail clients. We conclude that there is a problem of confidentiality on mails accesses using android email clients and companies have done little to control the safety of their email users.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0126.v1

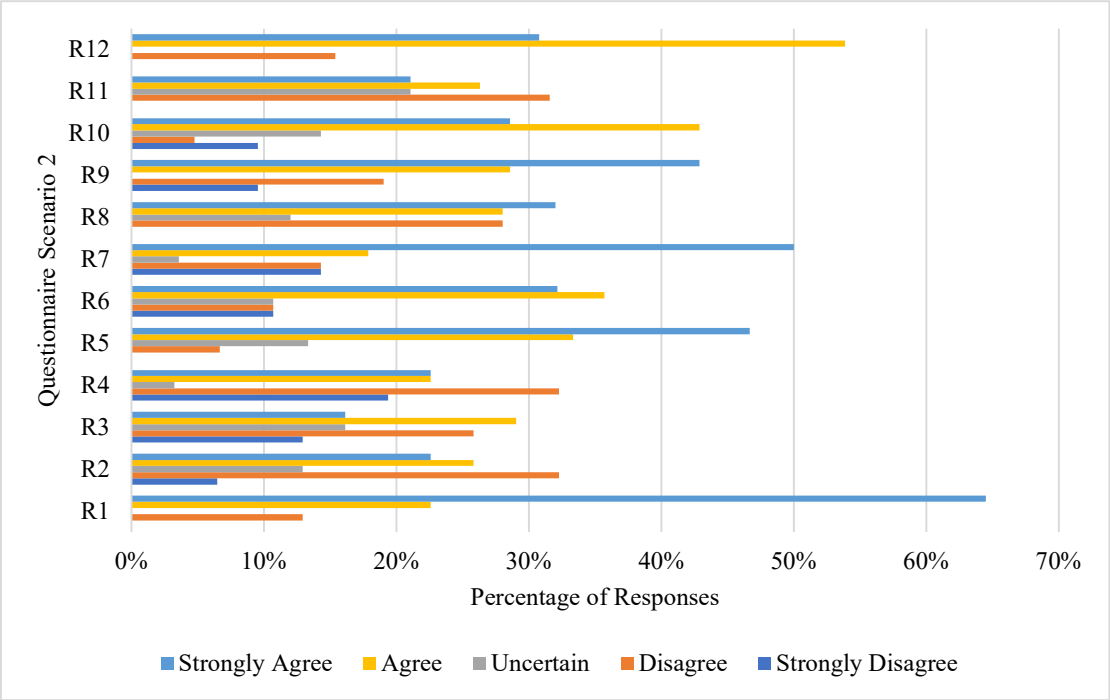

Figure 5: Percentage of Responses on the algorithm evaluation

The scenario 1, we conclude that there is a problem of confidentiality on mails accesses using android email clients and companies have done little to control the safety of their email users.

3.3. SWOT Analysis

The SWOT analysis here considered mail security services that apply to email client services, and concerning the assessment Table 3, the analysis was basically done on Antivirus, PGP, and OS Platform security. In information security, a SWOT analysis can be useful for developing a better understanding of the security environment. It can also support the business' overarching strategy by giving insight into security assets, risks, issues, and challenges that the information technology department - and thus, the business as a whole - will be faced with. In this analysis, we shall focus on specific email security techniques based on the literature were reviewed. The SWOT analysis of the designed data wipe algorithm was developed as depicted in Figure 6.

Table 3. Identification of client security techniques

Component	Antivirus	SSL	PGP	TLS	Anti-Spam software	Password	S/MIME	Platform (OS) Security	Dedicated Firewall
Client Security	✓	c	✓	○	○	✓	○	✓	○
Transit	○	ˬ	✓	✓	○	○	✓	○	○
Server Security	✓	c	✓	○	✓	○	○	✓	✓

SSL- Secure Socket Layer, PGP-Pretty Good Privacy, TLS-Transport Layer Security, S/MIME-Secure/Multipurpose Internet Mail Extensions, OS-Operating System

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0126.v1

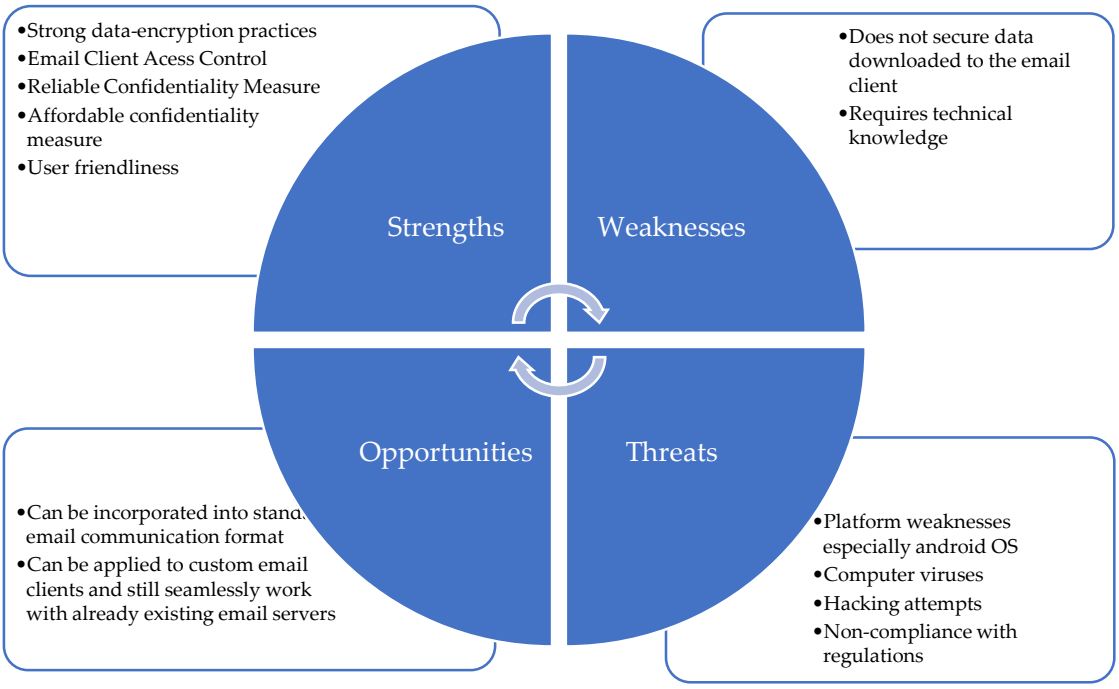

Figure 6: SWOT Analysis of the designed email wipe algorithms

These three features outlined above make the system such a unique communication system since the user has capabilities to track any changes that happened to his or her data. To establish facts, we eliminated the techniques that do not apply to email client data access security. In Table 4, then compared the confidentiality was compared and that the email-client security techniques provide to the emails a user sends or receive.

To this end, the auto-wipe algorithm is reliable, prevents access to data even when an account has successfully hacking attempts, can be incorporated into standard email communication protocol, easy to use and generally presents a better control attributes compared to password, OS platform security, antivirus and PGP techniques. The algorithm is designed to close the gap where other email security techniques fail to prevent unauthorized access to personal emails.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0126.v1

Table 4. Analysis of the developed algorithm against existing techniques based on android platform

S/N	Features	PGP	Password	OS Security	Antivirus	Auto-Delete Algorithm
1	Strong data-encryption practices	✓	○	✓	○	○
2	Email Client Access control	○	✓	✓	○	✓
3	Reliable confidentiality measure	○	○	○	○	✓
4	Affordable confidentiality measure	○	✓	✓	○	✓
5	User friendliness	○	✓	○	○	✓
6	Can ensure safety and confidentiality after other security techniques are	✓	○	✓	○	✓
7	Secure data downloaded to the email client	✓	○	✓	○	✓
8	Does not requires technical knowledge	○	✓	○	○	✓
9	Can be incorporated into standard email communication format	✓	✓	○	○	✓
10	Can be applied to custom email clients and still seamlessly work with already existing email servers	✓	✓	✓	✓	✓
11	Prevents threats on platform weaknesses especially android OS	✓	✓	✓	✓	✓
12	Computer viruses	○	○	○	✓	○
13	Prevents access to data when an account is successfully hacking attempts	✓	○	○	○	✓
14	Compliance with regulations	✓	✓	✓	✓	✓

Conclusion

With the robustness in the algorithm system and with its functionalities all aimed at ensuring the confidentiality of the message on the client's inbox, the result of this research gives assurance to the email sender, that outside the stipulated time interval, the confidentiality of the email at the email client is reliable. The algorithm offers an all-round confidentiality buildup on the security of the transmitted message. With most organisations and institution swiftly embracing the benefits of industry 4.0, this algorithm comes at such a point where email communication is at a great ordeal in becoming the next-generation business communication model because of its instant messaging features as opposed to its old approach. Future research should focus on scaling this algorithm to other email clients and mobile operating system other than android.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest

References

[1] I. Gorbans and U. Straujums, "The Myths about and Solutions for an Android OS Controlled and Secure Environment," in *Environment Technology Resources Proceedings of the International Scientific and Practical Conference*, 2015, vol. 3, no. September, pp. 54–64.

[2] Gilbert Gilibrays Ocen, S. M. Karume, M. S. Mutua, G. B. Mugeni, and D. Matovu, "AN ALGORITHM AND PROCESS FLOW MODEL FOR THE EXTRACTION OF DIGITAL FORENSIC EVIDENCE IN ANDROID DEVICES," *Int. Sci. J. Theor. Appl. Sci.*, vol. 72, no. 04, Apr. 2019.

[3] ACS, "Cybersecurity: Threats, Challenges and Opportunities," Sydney, 2016.

[4] O. Bongomin, G. Gilibrays Ocen, E. Oyondi Nganyi, A. Musinguzi, and T. Omara, "Exponential Disruptive Technologies and the Required Skills of Industry 4.0," *J. Eng.*, vol. 2020, pp. 1–17, 2020.

[5] Gilbert Gilibrays Ocen, Gilbert Barasa Mugeni, Karume Simon, Mutua Stephen, and Matovu Davis, "Evaluating factors responsible for inconsistencies in mobile devices digital forensic evidence extraction process model," *Int. J. Adv. Res. Ideas, Innov. Technol.*, vol. 5, no. 6, 2019.

[6] Y. Lu, "Industry 4.0 : A survey on technologies , applications and open research issues," *J. Ind. Inf. Integr.*, vol. 6, pp. 1–10, 2017.

[7] J. Vandermeer, "Seven highly successful habits of enterprise email managers: Ensuring that your employees' email usage is not putting your company at risk," *Inf. Syst. Secur.*, vol. 15, no. 6, pp. 64–75, 2006.

[8] I. Sanchez, A. Malatras, and I. Coisel, "A security analysis of email communications," 2015.

[9] M. Tracy, W. Jansen, K. Scarfone, and J. Butterfield, "Guidelines on Electronic Mail Security: Recommendations of the National Institute of Standards and Technology," Gaithersburg, 2007.

[10] S. Ruoti and K. Seamons, "Johnny's Journey Toward Usable Secure Email," *IEEE Secur. Priv.*, vol. 17, no. 6, pp. 72–76, Nov. 2019.

[11] S. L. Garfinkel, D. Margrave, J. I. Schiller, E. Nordlander, and R. C. Miller, "How to make secure email easier to use," in *CHI 2005: Technology, Safety, Community: Conference Proceedings - Conference on Human Factors in Computing Systems*, 2005, pp. 701–710.

[12] A. Reuter, K. Boudaoud, M. Winckler, A. Abdelmaksoud, and W. Lemrazzeq, "Secure email - a usability study," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12063 LNCS, pp. 36–46.

 doi:10.20944/preprints202107.0126.v1

[13] J. Clark, P. C. van Oorschot, S. Ruoti, K. Seamons, and D. Zappala, "SoK: Securing Email---A Stakeholder-Based Analysis," Apr. 2018.

[14] "US8732827B1 - Smartphone security system - Google Patents." .

[15] J. Wei, X. Chen, J. Wang, X. Hu, and J. Ma, "Enabling (End-to-End) Encrypted Cloud Emails With Practical Forward Secrecy," *IEEE Trans. Dependable Secur. Comput.*, 2021.

[16] K. M. Awan, M. Waqar, M. Faseeh, F. Ullah, and M. Q. Saleem, "Resource Management and Security issues in Mobile Phone Operating Systems : A Comparative Analysis," *PeerJPreprint*, pp. 1–18, 2017.

[17] J. H. Park, D. Kim, J. S. Park, and S. Lee, "An enhanced security framework for reliable Android operating system," *Secur. Commun. Networks*, vol. 9, pp. 528–534, 2016.

[18] M. Bashir, M. T. Afzal, and M. Azeem, "Reliability and Validity of Qualitative and Operational Research Paradigm," *Pakistan J. Stat. Oper. Res.*, vol. 4, no. 1, p. 35, 2015.