Hindawi
Journal of Engineering
Volume 2021, Article ID 4690611, 8 pages
https://doi.org/10.1155/2021/4690611

Research Article

An Algorithm for Improving Email Security on the Android Operating System in the Industry 4.0 Era

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 should be addressed to Gilbert Gilibrays Ocen; gilbertocen@gmail.com

Received 6 July 2021; Revised 24 October 2021; Accepted 9 November 2021; Published 20 November 2021

Academic Editor: Tian-Hua Liu

The world is attesting a tremendous change today, which is remarkably coined as industry 4.0. With several technologies shaping industry 4.0 epoch, notably, its cybersecurity entails the security of communication and network operations activities. The most common form of communication in organisations and business today is electronic mails (email). One of the major threats to email communication is the lack of confidentiality for emails accessed via Android mobile devices due to the weaknesses of the Android operating system (OS) platform. 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 stays in the receiver inbox before it automatically wipes off. Primary data were collected from email users using tightly structured questionnaires and respondents comprised of those with email technical background and typical email users, 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 autowipe algorithm addresses the confidentiality issues and threats on Android email clients.

1. Introduction

Android is today's most popular mobile operating system for smartphones, tablets, and other electronic devices including smart TVs. This popularity and endless use of the Android operating system creates many risks which are not fully recognized [1]. As technology continues to evolve, so also do the opportunities and challenges provide [2]. The increased use of technologies 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 adoption for usage by both businesses and individuals as a mean of communication presents a new form of concern [5]. The automation and digitalization of many business processes which are being adapted to technology forced people to depend on such technologies for communication and transactions. This has incredibly disrupted or retrofitted most industrial processes by the use or adoption industry 4.0 technologies [6]. Just as technology brings ever greater benefits, it also brings threats including cyberattacks. Therefore, protecting the communication mechanism such as email of organisations and businesses becomes a paramount priority [7].

Email is an electronic communication protocol used daily by most people, as well as governments and businesses across the world [8]. With the massive use of Internet and email communications, a new set of complementary standards and tools was created to harness the growing security and privacy concerns. However, these enhanced protocols

and tools have failed in practice to deliver effective protection [8]. To this end, worldwide email communications remain largely vulnerable to security and privacy threats [9]. Some researchers have suggested encrypting and signing emails to secure it [10–12]. This further complicates the means of information exchange since it places a greater load on the organization's network infrastructure [13]. However, for many organisations, the benefits of email encryption and signature will outweigh the costs [14].

A security module that provides protection to the mobile devices and the users 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, and collection of data about fraudulent and spam SMS messages) [12, 13, 15]. However, the popularity of mobile phones and the growing number of applications and different useful features include call features, calculations, maps, and applications for sending and receiving money, paying bills, and email communication [16]. Moreover, 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 leading to severe the damages [17].

There is a growing use of emails in the world, meaning there are higher possibilities of its usage to communicate confidential information. Confidential information like whistleblowing and sharing of bank information would prefer to be destroyed as soon as it is used. There is no reliable existing approach which provides the sender confidence that the information sent is safe or if needs to be deleted, that is, has been deleted unless a user initiates the action during message sending. Therefore, the outstanding contribution of this study is to develop an algorithm that gives the sender a role of deciding when their sent information is no longer available in the receiver's inbox. The algorithm is to prompt the sender at the time of sending on how long they want the email to stay in the receiver's inbox before autodeletion. The designed algorithm and application provide means of ensuring that user information is protected at the device end, thus strengthening confidentiality. The algorithm also empowers users with delete rights over their information after it has been decoded. The autodelete algorithm allows the sender of information to determine the time his/her information remains at the receiver's end. This study is structured as follows: Section 2 describes the methodology used to achieve the objectives of the study, while Section 3 presents the results and discussion followed by the conclusion section.

2. Methodology

The present study was conducted in four steps or phases as shown in Figure 1. 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 using questionnaires with selected participants, and SWOT analysis was done in the last phase.

2.1. Current State Analysis (Field Survey)

2.1.1. Study Area, Design, and Period. The survey was conducted in Kampala, Uganda. It was performed in the period of three months from January 15 to April 15, 2019. In this study, cross-sectional survey design was used.

2.1.2. Population and Sample Size. In this study, the users' selection criteria were based on the following: (1) users that are literate in the concept of security and confidentiality selection required experience, (2) frequent email users, and (3) those who have adopted emails from their Android email clients. While three institutions were randomly selected, both technical and nontechnical email users from each institution were considered. The fact that there were few users in number for each institution, the whole population was taken for the study. A nonprobability sampling was used to ensure that the samples are all frequent email users. This method allowed samples that are knowledgeable of email usage, are themselves email users, and have been using emails to communicate sensitive information, having significant experience of over 10 years. The purposive sampling technique was used to get knowledgeable respondents, and these were drawn from three companies including Luzira Prisons, Neptune Software Group, and UGAFODE Microfinance Limited. This was because these organisations had users with significant experience in email usage and sharing of confidential information through emails.

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

2.1.3. Data Collection. Two sets of closed questionnaires were used to collect the opinion of the users; this is because, we wanted the users to limit their opinion only to the subject we are researching. This assessment was conducted using closed questionnaires submitted to selected users based on their knowledge, email adoption, and how much they rely on email for sensitive communication. Supplementary Material (S1) provides detailed information on the two sets of questionnaire survey (scenario 1 and 2).

2.1.4. Data Quality Assurance. The two set of questionnaires (scenario 1 and 2) were validated by two associate professors in the field of ICT. To establish the reliability, the internal correlation method was utilized for each questionnaire. The Cronbach's alpha for the mentioned dimensions were 0.722 and 0.706, respectively.

2.1.5. Ethical Consideration. The ethical clearance for the survey was obtained from the Institutional Research Ethical Committee of Busitema University and informed consent of respondents before enrolling them voluntarily in the study. Ethics issues such as privacy and confidentiality of the

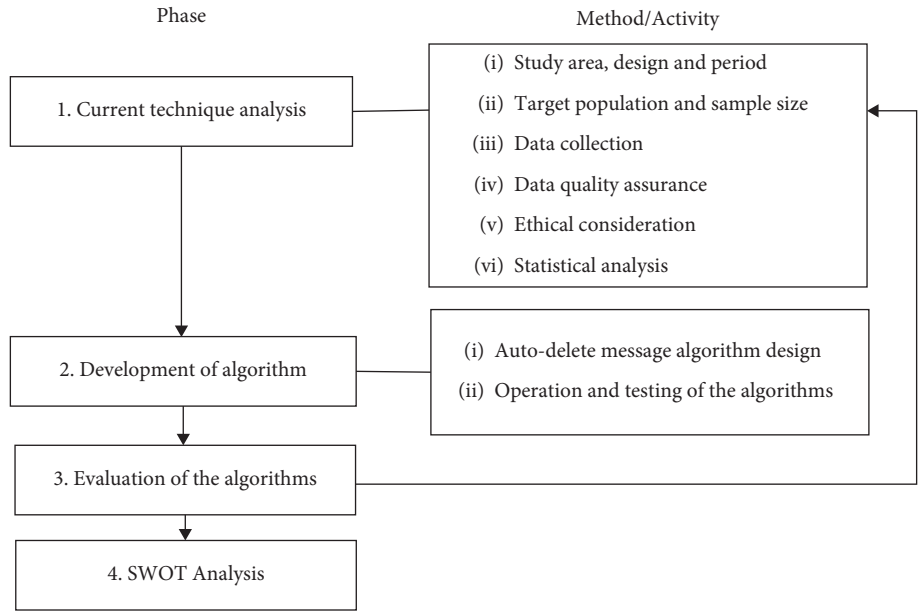

Figure 1: Methodology approach.

Table 1: Sample size of the population.

Institution	Technical email users	Nontechnical 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

respondents were ensured. Besides, the letter was acquired from the university that acted as an introductory document to different organization and individuals engaged on this research. It was also ensured that the algorithm developed does not execute any unintended/undisclosed activity in the users' devices.

2.1.6. Statistical Analysis. The participants' responses to both set of questionnaires were measured by questions on a five-point Likert scale rating, ranging from strongly agree (5), agree (4), neutral (3), disagree (2), and strongly disagree (1). The mean score of every question was calculated out of five. Descriptive statistics were calculated for all items. The results were analyzed with the use of SPSS software version 20.0 (SPSS, Chicago, Illinois). Internal consistency reliability of each questionnaire was measured by Cronbach's alpha, where coefficients of ≥0.7 demonstrate acceptable internal consistency.

2.2. Development of the Algorithm

2.2.1. Design of the Autodelete Algorithm. The class diagram describing the autodelete algorithm was designed using Unified Modelling Language (UML) tools as shown in Figure 2.

Being an additional feature to a given application, the data-wipe algorithm gives the message sender authority to the message, and the size of the application needs to be small.

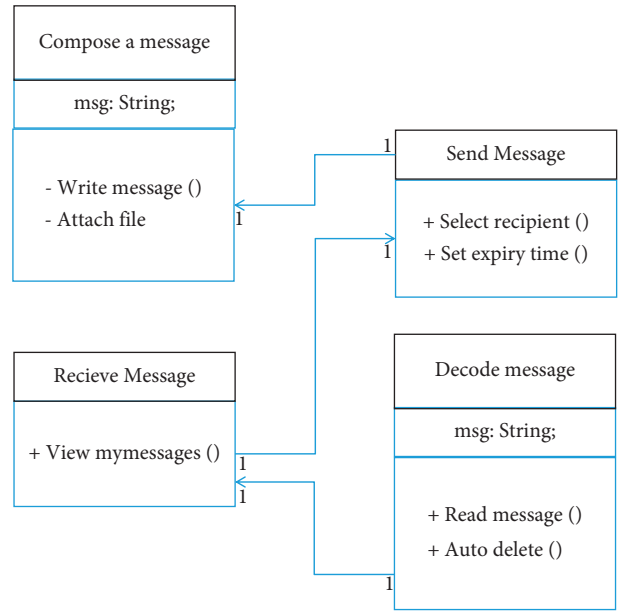

Figure 2: Class diagram of the autodelete algorithm.

Therefore, several loops and languages used in building the algorithm were examined. As the result, 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 compared to ordinary Java programming language. This would mean that Kotlin presents shorter function than Java programming

language. For code optimisation and algorithm efficiency, Kotlin was the preferred base programming language used to develop the algorithm. The algorithm for autodeleting the message was designed using UML and structured programming. The pseudocode of the developed autodelete algorithm is shown in Figure 3.

2.2.2. Description of the Algorithm. The pseudocode algorithm for autodelete messages in Figure 3 is clearly described as follows:

Step 1: the algorithm captures the message or any file to send

Step 2: the algorithm captures the destination device IP

Step 3: instantiate the date class and capture the sending time and the receiving time for the shared message

Step 4: the algorithm captures the expiry time for the shared message. If the 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 device.

The operation of the designed autodelete message algorithm is shown in Figure 4.

2.3. Evaluation of the Algorithm. To evaluate the developed algorithm, a mobile software application was developed where the algorithm was embedded to test its ability to accurately work. Two users were selected and given access to the developed application and one sent a message to the other with a set time for this message to be visible and after expire. When the other user of the algorithm successfully received the message, a test was done to ensure that the received message is autodeleted from the receiver's inbox at the specified time as set by the sender. The results of this system testing were positive and according to the expectation of the researcher and the testing team. Furthermore, field survey was conducted following the same principle as in phase 1 of the study to evaluate the algorithms based on the users' opinion. In this phase, the second set of questionnaires (scenario 2) was used. Due to the 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 it is where the autodelete 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 to the use of the algorithm. Supplementary Material (S2) provides the security demo of the designed application.

2.4. SWOT Analysis. In this study, the SWOT analysis of the autodelete data algorithms was based on the literature search and review. Relevant literature from the latest articles and publication about data confidentiality trends, analysis on various encryption, and data security-related articles were collected. Altogether, a review of 45 latest documents selected from 2008 to date, 15 (33.33%) were from selected journals, 15 (33.33%) from conference papers, 10 (22.22%) from books, and five (11.11%) from selected Wikipedia. The journals and conference papers filtered from Google Scholar were looked at. Given that most of these articles and conference papers did not tackle confidentiality at the device end, we concluded on the 45 articles and based our research on them.

3. Results and Discussion

3.1. Demographic Characteristics. The results showed that out of 31 respondents who were given the questionnaires, 100% returned valid results and 14 (45%) were technical users. The data collected were then categorized, quantified, and then coded. Data analysis in this study was done using Statistical Package for Social Science (SPSS). The majority of the respondents were male with a percentage of 64% while 36% were female. Regarding the departments where the respondents came from, 34% came from ICT-related departments, while 66% came from non-ICT-related departments, when examining the expert period taken; while using emails, we discovered that 43%, 28%, and 29% had used emails for, respectively, between 1 and 3 years, 4 and 7 years, and above 8 years.

3.2. Reliability Testing. Reliability is the degree to which the study provides consistent results when analyzing a similar population [18]. Therefore, it helps to explain the degree to which an instrument measures the same way, and every attempt is employed under similar conditions 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 shown in Table 2. Supplementary Material (S3) provides detailed information on the reliability statistics.

Table 2 provides that all parameters are above Cronbach's alpha's 0.7 value, which is considered acceptable for academic research.

3.3. Evaluation of Autodelete Data Algorithms. The evaluation of autodelete data algorithms was performed using the second set of the questionnaire (scenario 2). Regarding each of the research questions (R1–R12), all evaluation questions were positively set, and the results showed that all questions (strongly agree or agree) gave an average of 65%. Figure 5 shows that the majority of the respondents agree with the algorithm provided and have recommended the algorithm as a better solution to ensure confidentiality of the email clients. It was concluded that there is a problem of confidentiality on mail accessed using Android email clients, and companies have done little to control the safety of their email users.

3.4. SWOT Analysis. The SWOT analysis here considered mail security services that apply to email client services and concerning the assessment. The analysis was done on

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: Pseudocode for autodelete email message.

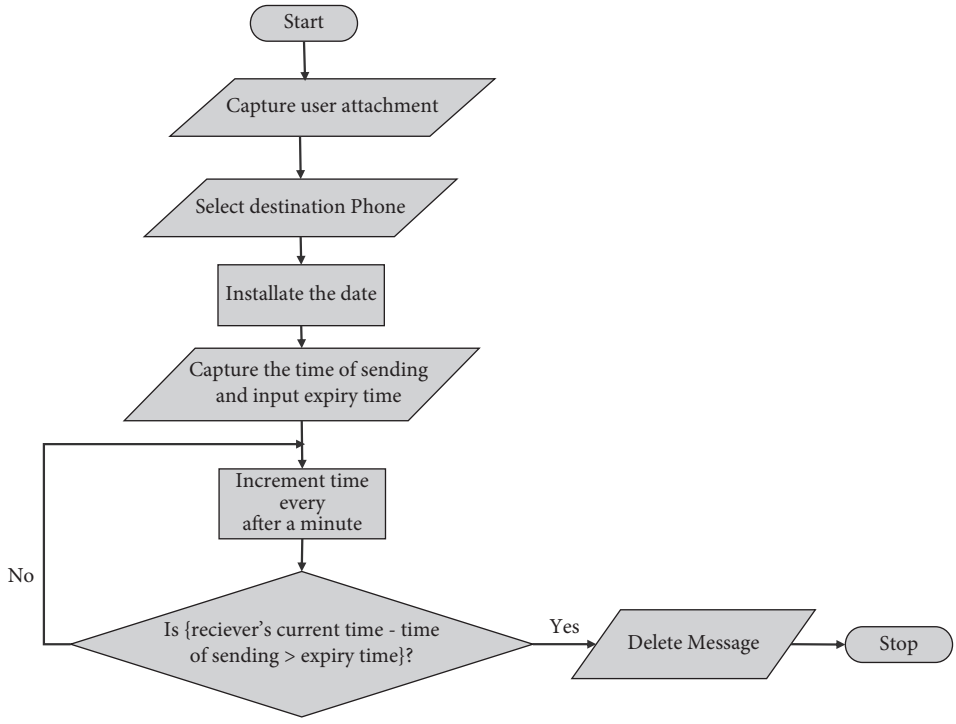

Figure 4: A flowchart showing the description of the algorithm.

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

antivirus, PGP, and OS platform security as given in Table 3. 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 the 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, specific email security techniques were prioritized based on the literature that were reviewed. The SWOT analysis of the designed data wipe algorithm was developed as shown in Figure 6.

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

To this end, the autowipe algorithm is reliable, prevents access to data even when an account has successfully hacked attempts, can be incorporated into standard email communication protocols, easy to use, and generally presents better control attribute 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.

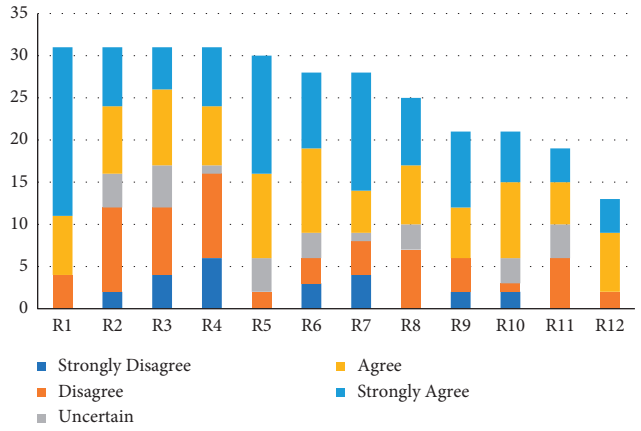

Figure 5: Autodelete data algorithm evaluation chart.

Table 3: Identification of client security techniques.

Component	Antivirus	SSL	PGP	TLS	Antispam software	Password	S/MIME	Platform (OS) security	Dedicated firewall
Client security	√	○	√	○	○	√	○	√	○
Transit	○	√	√	√	○	○	√	○	○
Server security	√	○	√	○	√	○	○	√	√

SSL, secure socket layer; PGP, pretty good privacy; TLS, transport layer security; S/MIME, secure/multipurpose Internet mail extensions; OS, operating system.

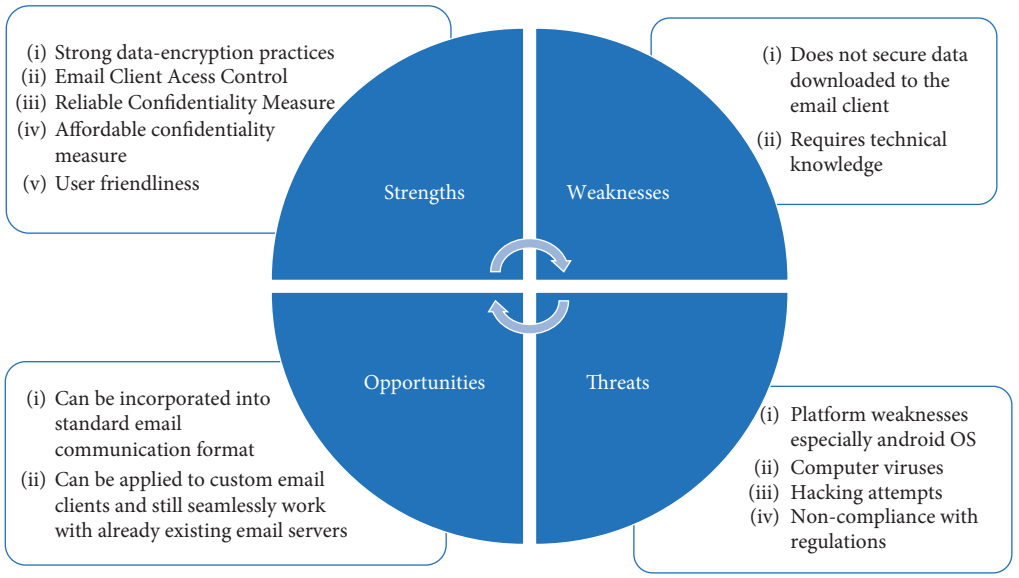

Figure 6: SWOT analysis of the designed email wipe algorithm.

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

S. No.	Features	PGP	Password	OS security	Antivirus	Autodelete 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	√	○	√	○	√
7	Secure data downloaded to the email client	√	○	√	○	√
8	Does not require technical knowledge	○	√	○	○	√

Table 4: Continued.

S. No.	Features	PGP	Password	OS security	Antivirus	Autodelete algorithm
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	√	√	√	√	√

4. Conclusions

The robustness of the algorithm system with its functionality ensures the confidentiality of the message on the client's inbox. The autodelete or autowipe algorithm offers an all-round confidentiality build up on the security of the transmitted message. With most organisations and institutions 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. This research introduced new knowledge to the research fraternity by bringing a new security approach to emails and introducing need for more research on security on email client. Future research should focus on scaling this algorithm to other email clients and mobile operating systems other than Android.

Data Availability

The research data underlying the findings of the study can be accessed from the Figshare data repository at the link https://doi.org/10.6084/m9.figshare.14916030.v1.

Disclosure

The preprint version of this work is deposited with Preprints.org at the link https://www.preprints.org/manuscript/202107.0126/v1s [19].

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Supplementary Materials

S1. Supplementary material containing the questionnaire survey for scenario 1 and scenario 2 (docx). S2. Supplementary material showing the security demo of the data wipe application (docx). S3. Supplementary material detailing the reliability testing for scenario 1 and scenario 2 surveys (docx). . (*Supplementary Materials*)

References

[1] I. Gorbans and U. Straujums, "The myths about and solutions for an android OS controlled and secure environment," *Environment Technology Resources Proceedings of the International Scientific and Practical Conference*, vol. 3, pp. 54–64, 2015.

[2] G. G. 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," *International Journal of Theoretical and Applied Sciences*, vol. 72, no. 04, 2019.

[3] ACS, *Cybersecurity: Threats, Challenges and Opportunities*, ACS, Sydney, Austalia, 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," *Journal of Engineering*, vol. 2020, Article ID 4280156, 17 pages, 2020.

[5] G. G. Ocen, G. B. Mugeni, K. Simon, M. Stephen, and M. Davis, "Evaluating factors responsible for inconsistencies in mobile devices digital forensic evidence extraction process model," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 5, no. 6, 2019.

[6] O. Bongomin, A. Yemane, B. Kembabazi et al., "Industry 4.0 disruption and its neologisms in major industrial sectors: a state of the art," *Journal of Engineering*, vol. 2020, Article ID 8090521, 45 pages, 2020.

[7] Y. Lu, "Industry 4.0: a survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

[8] J. Vandermeer, "Seven Highly Successful Habits of Enterprise Email Managers: ensuring that your employees' email usage is not putting your company at risk," *Information Systems Security*, vol. 15, no. 6, pp. 64–75, 2006.

[9] I. Sanchez, A. Malatras, and I. Coisel, *A Security Analysis of Email Communications*, JRC, Ridgefield, CT, USA, 2015.

[10] S. Ruoti and K. Seamons, "Johnny's journey toward usable secure email," *IEEE Security & Privacy*, vol. 17, no. 6, pp. 72–76, 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 *Proceedings of the CHI 2005: Technology, Safety, Community: Conference Proceedings-Conference On Human Factors in Computing Systems*, pp. 701–710, Portland, OR, USA, April 2005.

[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)* LNCS, Berlin, Germany, 2020.

[13] J. Clark, P. C. van Oorschot, S. Ruoti, K. Seamons, and D. Zappala, "SoK: Securing Email-a stakeholder-based analysis," ArXiv, 2018.

[14] M. Tracy, W. Jansen, K. Scarfone, and J. Butterfield, *Guidelines on Electronic Mail Security*, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2007.

[15] J. Wei, X. Chen, J. Wang, X. Hu, and J. Ma, "Enabling (End-to-End) encrypted cloud emails with practical forward secrecy,"

IEEE Transactions on Dependable and Secure Computing, 2021, preprint.

[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," *PeerJ*, pp. 1–18, 2017, Preprint.

[17] J. H. Park, D. Kim, J. S. Park, and S. Lee, "An enhanced security framework for reliable Android operating system," *Security and Communication Networks*, vol. 9, no. 6, pp. 528–534, 2016.

[18] M. Bashir, M. T. Afzal, and M. Azeem, "Reliability and validity of qualitative and operational research paradigm," *Pakistan Journal of Statistics and Operation Research*, vol. 4, no. 1, p. 35, 2015.

[19] I. M. Kisembo, G. G. Ocen, O. Bongomin et al., "An algorithm for improvement of email security on android operating system in the era of industry 4.0," *MDPI*, pp. 1–11, 2021, Preprint.