Performance Enhancement of Large-Size NFC Multi-Touch System

Samuel King Opoku

Abstract—**Multi-touch technology interfaces are becoming omnipresent due to its user-friendly human-machine interaction. Medium to large size multi-touch interfaces are implemented using camera based systems, capacity, resistive or pressure sensing systems or LED system. This paper presents security and performance vulnerabilities of multi-touch systems implemented in a multi-user environment. The paper reviews the main related technologies and mechanisms such as authentication, auditing, non-repudiation and collision detection. The paper also proposes enhanced designs based on NFC tags and mobile readers. Algorithms are designed and implemented using Java.**

***Index Terms*—Algorithm, API, Bluetooth, Multi-touch, NFC, Collision Detection, Security mechanism,**

I. Introduction

SECURITY mechanisms have become prevalent issues in modern technology in curbing system malfunctioning. The predominant security features include authentication, auditing and non-repudiation [1]-[2]. Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. Authentication is usually achieved through user ID and password matching. Recent technologies involve the use of biometric features such as figure print. Non-repudiation refers to a state of affairs where the purported maker of an action will not be able to successfully challenge the validity of the action. Non-repudiation is achieved through the use of digital signatures – peculiar information related to a user such as email address. Auditing or traceability on the other hand is the ability to ascertain who used the system at a particular time and the activities the user undertook. This requires that users' activities are kept so that uses actions can be traced.

Multi-touch screens capture and react to simultaneous interactions of several touch devices such as fingers or hands allowing user-friendly human-machine interactions. Medium to large size of the tangible interface are implemented using camera based systems, capacity, resistive or pressure sensing systems or LED systems [3]. Camera based systems use cameras either behind or above screens to recognize movements on screens [4]-[6]. Capacitive sensing system [7], [8] detects the capacity of the screen by means of antenna sets. Other systems use communication technologies such as infra-red transmitters [9], [10] or NFC tags [11]. Camera based system that captures the full image of the user can perform security checks. However, it presents low resolution and high computational requirements for image processing. Capacitive sensing and infra-red systems are best suited for small to medium size screens but not scale for large size screens. NFC systems, however, support medium to large size screen with low computational requirements using NFC-enabled mobile devices to read the underlining NFC tags. It lacks collision detection mechanism and security implementations. The paper presents the implementation of such security mechanisms as authentication of mobile phones, auditing and non-repudiation of users. It uses authenticating code provided by the user for the NFC-enabled mobile device pair authentication. The Bluetooth address of the communicating pair device and the activities of users are used to audit and non-repudiate each user of the multi-user environment. The locations of the various NFC tags are used to detect and correct collisions. The NFC-enabled mobile device will run a J2ME application that reads the IDs of the NFC tags, captures user authenticating code and other activities such as names and tag IDs of selected images. The mobile devices use Bluetooth to send the captured information to a backend server which is a computer. A Java based desktop application receives the NFC tag IDs and other information associated with each mobile device and performs security vulnerability checks and collision detection mechanism

NFC is a wireless communication technology which is implemented on personal devices such as smart phones, tablets and other consumer electronic devices. However, to work with NFC for more than ten centimeter radius, it has to be supported by longer and faster communication technology. In view of the fact that NFC does not support online connectivity [12], it is better to support NFC with other such short range communication technologies as infra-red or Bluetooth. Infra-red technology is limited by its line of sight operation [13] and atmospheric conditions [14]. Bluetooth supports network of data exchange and can also cover a radius of up to thirty meters which is six times more that infra-red coverage [13], [15]. It is limited by excessive power consumption [13]. The APIs needed for NFC application is the Contactless Communication API (JSR 257). This supports read and write operations between mobile devices and smart posters or NFC tags. Coupled with the Security and Trust Service API (JSR 177), the read and write operations can be done through APDU communication. An extension of Contactless Communication API supports peer-to-peer communication

The author is with the Computer Science Department, Kumasi Polytechnic, Postal Code 854, Kumasi, Ghana, West Africa (phone: +233-242-124-291; e-mail: Samuel.k.opoku@gmail.com).

between NFC devices [15]. The third mode of operation of NFC technology is Tag emulation. In Tag emulation, NFC device behaves like a smart card permitting payment and ticketing. Fig. 1 illustrates the various APIs of NFC

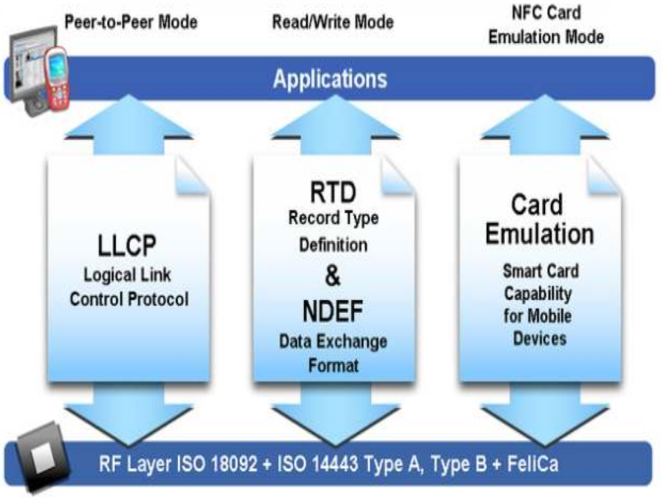


Fig. 1. Functional View of NFC Operations (Source [17])

The Bluetooth API (JSR 82) is integrated with the above-mentioned APIs when Bluetooth is required. J2ME is the platform used in order to develop applications for mobile devices. The application is developed using MIDlet with the help of CLDC configuration supported by MIDP profile. Pre-commercial SDKs widely used for NFC applications are Nokia 6131 NFC SDK 1.1 and Series 40 Nokia 6212 NFC SDK. These SDKs provide the APIs required for NFC applications. The SDKs are integrated into IDEs such as NetBeans for application development. The compatibility of J2ME and J2SE enables the development of a backend server to assist the limited processing power of mobile devices especially mobile phones. J2SE is ideally used for GUI applications [16]. The Bluecove API is integrated into J2SE for Bluetooth communication between the backend server and the mobile devices.

A typical NFC based multi-touch system [11] localizes and tracks the movements of the hands using NFC phones to read NFC tags which are placed behind the screen. The NFC phones run J2ME application which reads the IDs of the tags and send them to a computer controlling the projection screen using Bluetooth. The Java based desktop application receives the NFC tag IDs and perform gesture and shaper identification of each of the users and eventually control the projection of the system. The basic infrastructure of the NFC multi-touch system is shown in Fig. 2 which consists of at least two NFC mobile phones, a PC with a Bluetooth adapter and a projector. The rear view of the projecting screen is captured in Fig. 3.

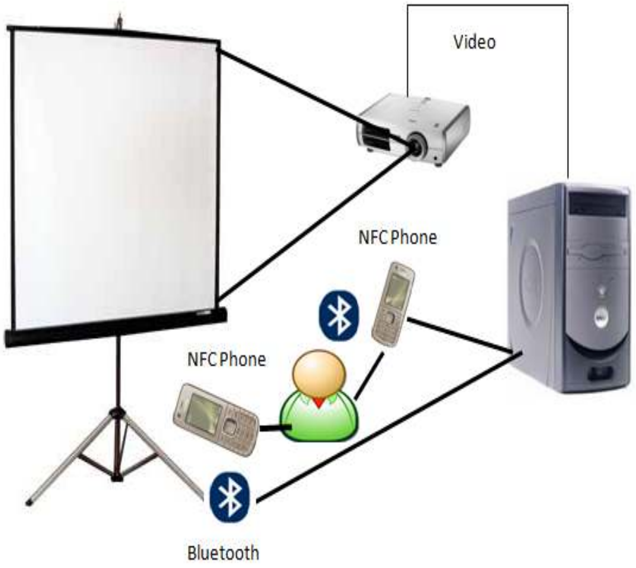


Fig. 2. NFC Multi-touch System

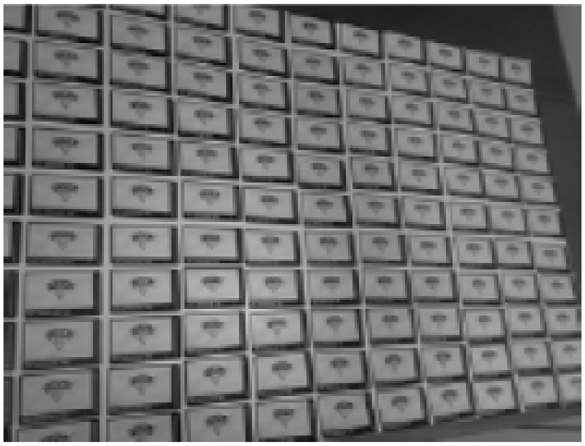


Fig. 3. Rear View of the Projecting Screen with NFC tags (Source [11])

II. SYSTEM DESIGN

This section focuses on designing algorithms needed to implement authentication, user initiation and collision detection. The system knows all the tags and their respective locations based on the IDs.

A. Authentication of Mobile Phone Pair

Authentication ensures that the communicating phone pair comes from the same user and also two phones are exactly used. This process is crucial for the purpose of authentication and non-repudiation. Authentication of mobile phones requires that the same three-digit number is entered for the two phones. The algorithm illustrated with flowchart in Fig. 4 below shows the design of authentication process.

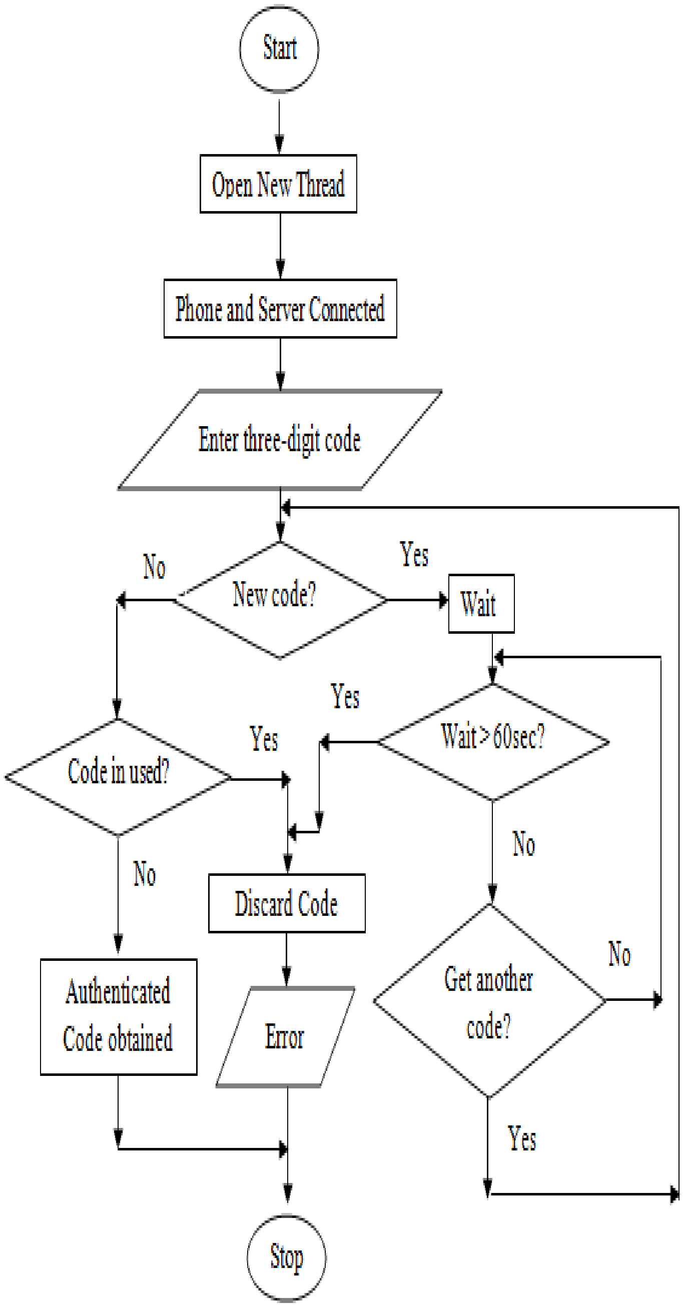


Fig. 4. Authentication Process of mobile phone pair

B. *Initialization of Working Area*

The system cannot be used unless it is initialized. Initialization allows the user to select a portion of the large screen as a working area. The selection is done by using only one phone of the already authenticated phone pair. The user can draw a rectangular area on the large screen or only a vertical line which represents the y-axis and horizontal line which represents the x-axis with the phone as a working area. The drawing lines representing the working area must meet in order to have a continuous plane. The flowchart shown in Fig. 5 depicts the initialization process

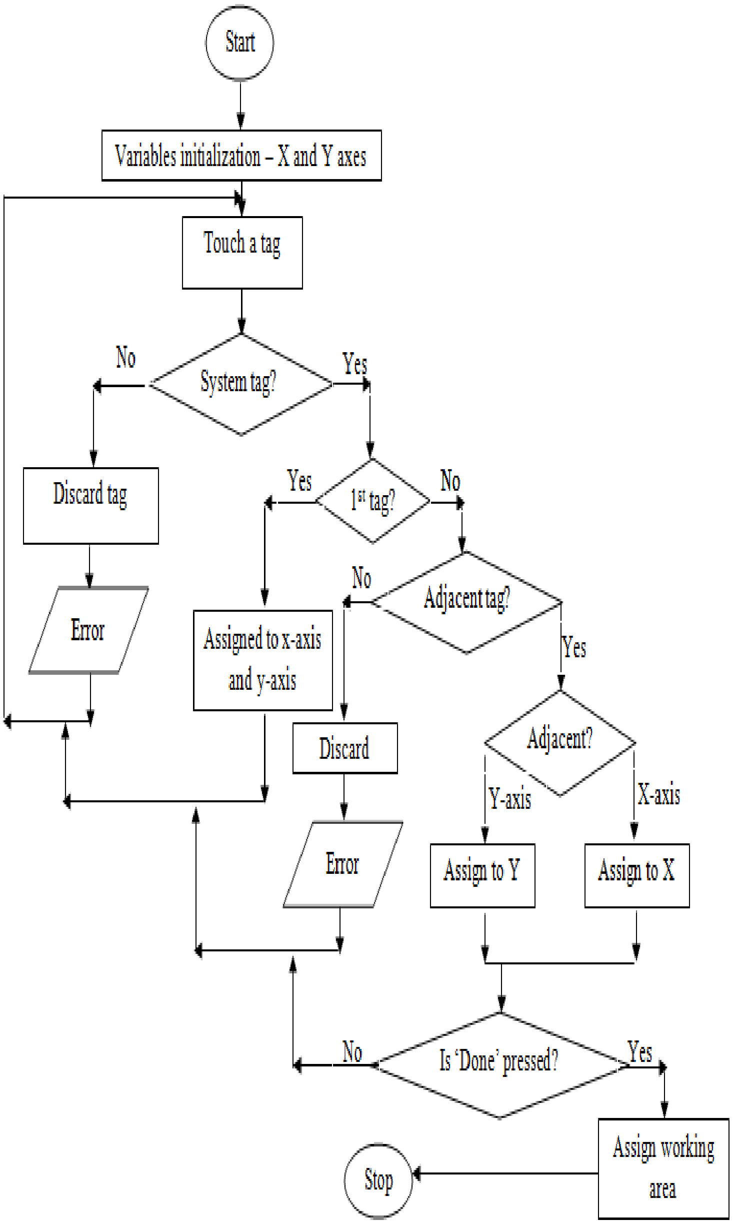


Fig. 5. Initialization Process of Working Area

C. *Collision Detection Mechanism*

The following mechanism is used to control image resizing in order to avoid collision with the sides of the screen or other working areas and also to prevent disappearance of the image after reaching the minimum size. Given a working are A, let W and H be the maximum width and height respectively. Moreover w and h respectively be the minimum width and height of the image such that $w \epsilon A$, $W \epsilon A$, $h \epsilon A$ and $H \epsilon A$. Using the number of tags in each axis such that T_x >5, the number of tags in x-axis and T_y > 5, the number of tags in the y-axis, it implies that W = T_x and H=T_y. The maximum allowable resize area of the screen is divided into five. This follows that w = W / 5 and h = H / 5. Let x and y be the width and height respectively of the image at any given time. Then, There is no collision or image disappearance if and only if $w \leq x \leq W$ and $h \leq y \leq H$.

To resize an image, consider W_c and H_c as the current width and height respectively of the image and W_k and H_k be the next width and height respectively of the image after resizing such that $W_k > W_c$ and $H_k > H_c$, it follows that $W_k = (c+1)\,W_c$, $H_k = (c+1)\,H_c$, $\forall\, c = 1,2,3,4$ and $\forall\, k = 2,3,4,5$. Similarly, if $W_k < W_c$ and $H_k < H_c$, it follows that $W_k = (c-1)\,W_c$, $H_k = (c-1)\,H_c$, $\forall\, c = 2,3,4,5$ and $\forall\, k = 1,2,3,4$. Where c is the current level of allowable resizing such that $1 \le c \le 5$. The initial level when the image is first displayed is four (c = 3).

III. SYSTEM IMPLEMENTATION

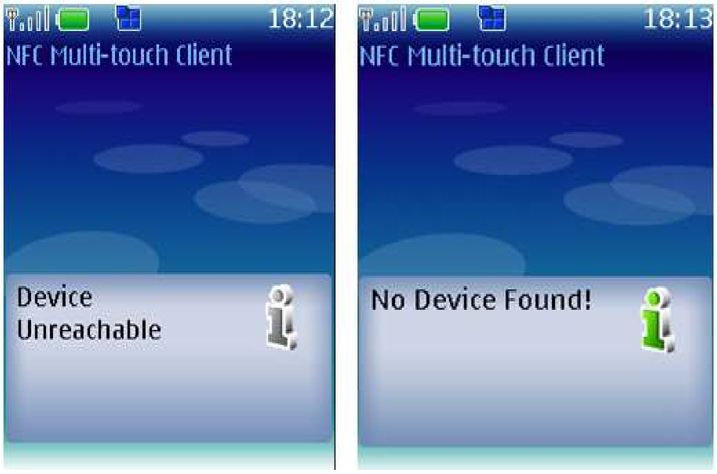
The mobile phone application was implemented using J2ME (CLDC 1.1 / MIDP 2.0). The Contactless Communication API (JSR 257) was used for accessing NFC tags by the phones. The backend server application was implemented using J2SE and the Bluecove API was used for the communication with the Nokia NFC phones. The following sub-sections describe the implementation of the various features to enhance system performance and maintain security

A. *Client and Server Connectivity*

The SDK used to implement the system was Nokia 6131 and made compatible with other Nokia NFC phones. A major finding revealed that the system's performance is affected by the battery level of the mobile phones in discovering devices and services especially when the phones are not Nokia 6131 whose SDK was used. The following inferences were made:

- When the battery level was half, the phones discovered the server but could not communicate with the service running on the server
- When the battery level was less than half, the phones could not even find the server let alone the services located on the server
- When the battery level was more than half, the phones were able to discover the server and communicate effectively with the service

Fig. 6 shows the messages usually displayed when there is error in connection.

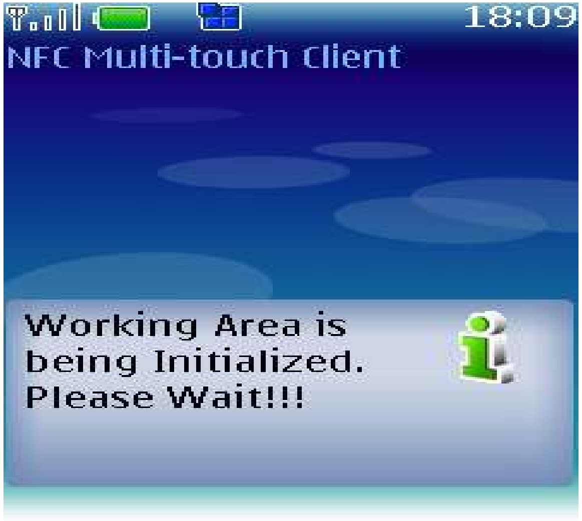


Device Unreachable

No Device Found!

Fig. 6. Client and Server Connectivity Error Messages

During system initialization, the idle phone is blocked from usage. Fig. 7 shows the error message displayed when the idle phone of the paired phone is used during initialization

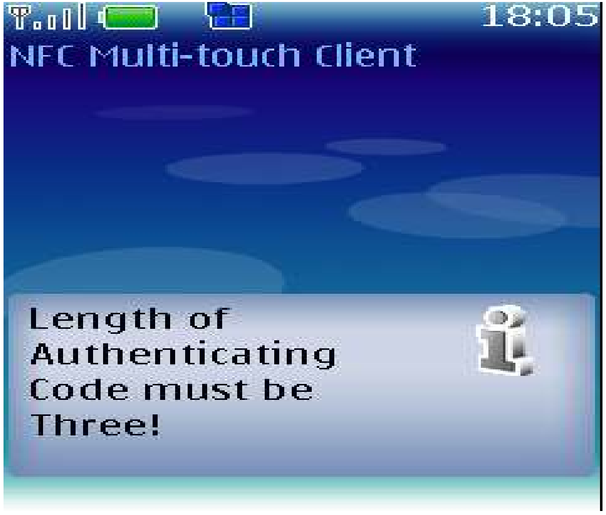


Working Area is being Initialized. Please Wait!!!

Fig. 7. Idle Phone Used During Initialization

B. *Authentication of Mobile Phone Pair*

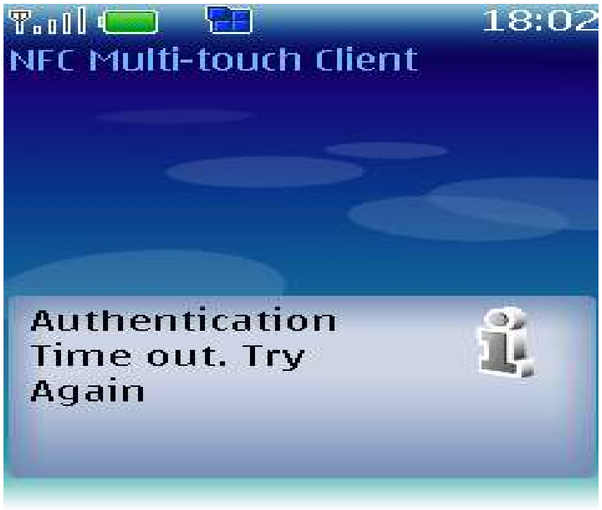
Users are prompted to enter three-digit code for the communicating phone pair. Fig. 8 illustrates the message displayed when the user enters less than or more than digits



Length of Authenticating Code must be Three!

Fig. 9. Wrong Authentication Code Entered

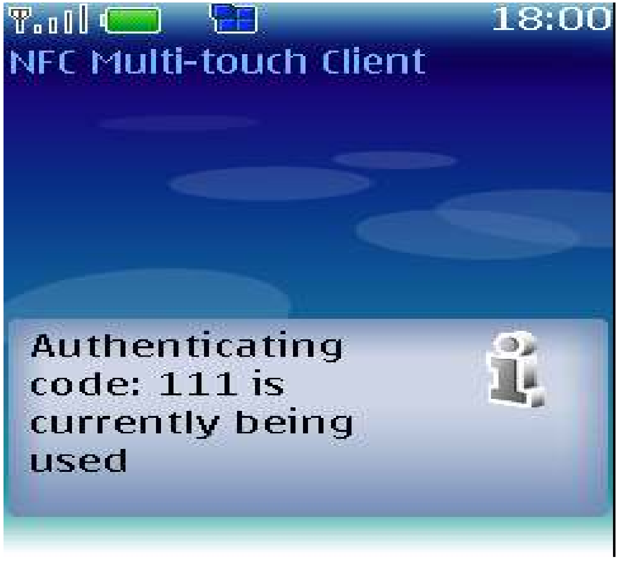
The system waits for sixty seconds for the other pair to be authenticated. The message displayed after sixty seconds is shown in Fig. 10.



Authentication Time out. Try Again

Fig. 10. Authentication Time-out

The system rejects any incoming connection with authentication code already in used as shown by Fig. 11.

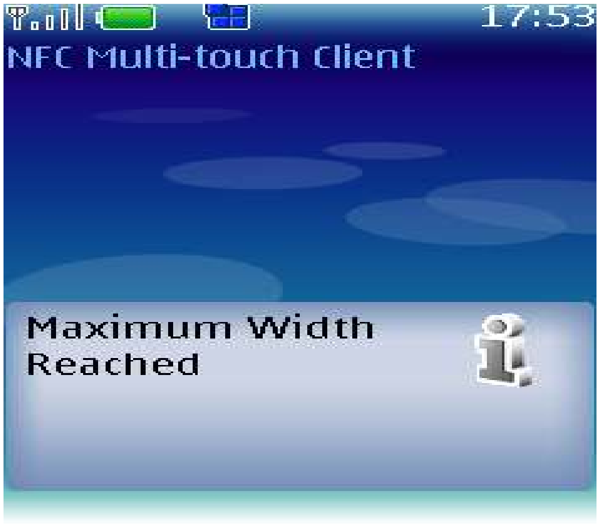


Authenticating code: 111 is currently being used

Fig. 11. Rejection Message for Existing Authenticating Code

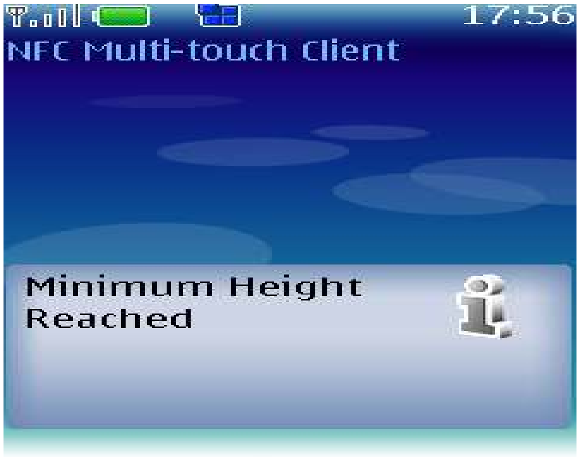
C.   *Collision Detection Implementation*

The collision detection mechanism was implemented as part of the backend server activities. However, during collision or image disappearance, an error message is issued by the backend server which is displayed on the mobile phone which was lastly used in touching the tags. The following figures illustrate some of the messages displayed when maximum or minimum size is obtained.



Maximum Width Reached

Fig. 12. Maximum Width is Reached

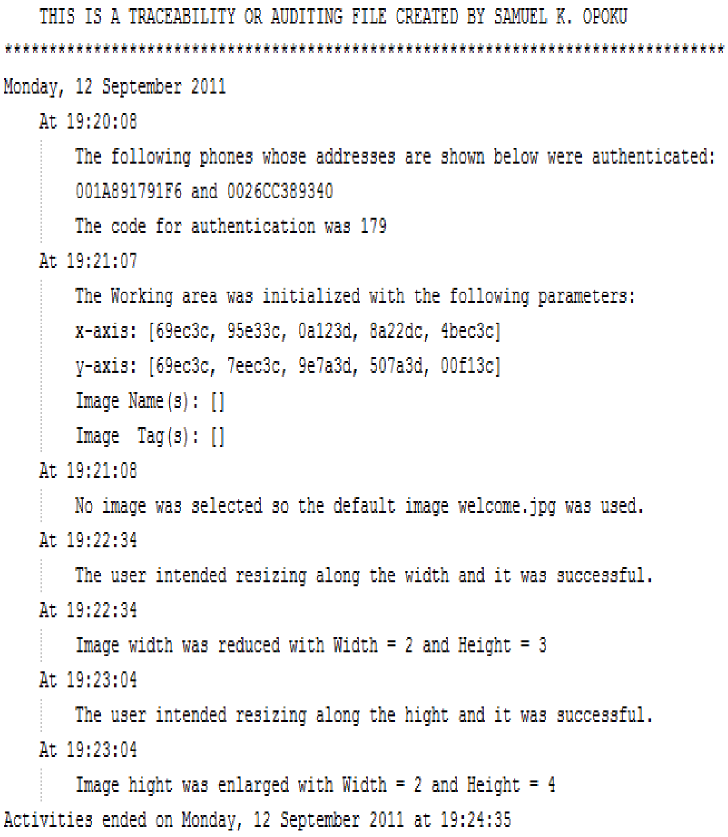


Minimum Height Reached

Fig. 13. Minimum Height is Reached

A.   *Auditing and Non-Repudiation Implementation*

There is no interface associated with this functionality since its implementation is hidden from users. Information about the activities transpired is stored. This helps administrators to ascertain who used the system at any given time and what the user did. The information is stored in a text file. Any time an authenticated phone sends a message to the server, usually tag ID, the message also contains the address of the phone and the authenticating code which has a copy stored on the phone. With this information, the system is able to identify the action carried out by each of the multi-users. The Bluetooth address of the phone ensures that users cannot deny their actions. Fig. 14 depicts typical information found in an auditing or traceability file.

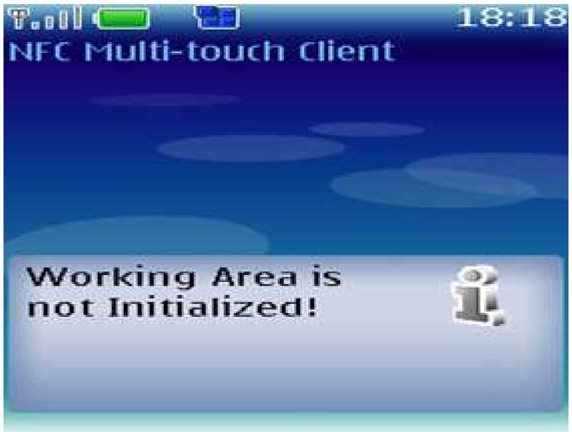


```
    THIS IS A TRACEABILITY OR AUDITING FILE CREATED BY SAMUEL K. OPOKU
*******************************************************************************
Monday, 12 September 2011
    At 19:20:08
        The following phones whose addresses are shown below were authenticated:
        001A891791F6 and 0026CC389340
        The code for authentication was 179
    At 19:21:07
        The Working area was initialized with the following parameters:
        x-axis: [69ec3c, 95e33c, 0a123d, 8a22dc, 4bec3c]
        y-axis: [69ec3c, 7eec3c, 9e7a3d, 507a3d, 00f13c]
        Image Name(s): []
        Image  Tag(s): []
    At 19:21:08
        No image was selected so the default image welcome.jpg was used.
    At 19:22:34
        The user intended resizing along the width and it was successful.
    At 19:22:34
        Image width was reduced with Width = 2 and Height = 3
    At 19:23:04
        The user intended resizing along the hight and it was successful.
    At 19:23:04
        Image hight was enlarged with Width = 2 and Height = 4
Activities ended on Monday, 12 September 2011 at 19:24:35
```

Fig. 14. Typical Information in an Auditing File

Any vulnerability action against the working area that is indicated by x-axis and y-axis tags within the stipulated time can therefore be traced. Users' actions being successful as shown in Fig. 14 above indicated that the system was functioning well when it was being used and that the user cannot deny that s/he could not use the system because it was already malfunctioning. The non-repudiation service implemented is considered weak in this work in the sense that stolen phones can be used when the user whose activity adversely affected the system's functionality is not caught instantly.

When users select image to manipulate without initializing a working area first, the system rejects their activity and issues a message as shown in Fig. 15.



Working Area is not Initialized!

Fig. 15. Using the System before Initialization

IV. Conclusion

The implementation of security mechanism into NFC multi-touch system indicates that NFC as a technology can be used to develop complex systems and applications depending on the implementation of the backend server which usually communicates with mobile devices using Bluetooth. The implemented system requires low computing resource systems and minimal initial calibration to initialize the working area.

References

[1] C. Kaufman, R. Perlman, M. Speciner, E. Cliffs, "*Network security : Private Communication in a Public World*" Prentice Hall, 1995

[2] W. Stallings, "Cryptography and Network Security. Principles and Practice", Fouth edition Prentice Hall 2006. http://williamstallings.com/

[3] Touch Base, "*Windows 7 Touch Implementation*", 6th November 2009, http://touchbase.com/documentation/Windows%207%20Touch%20Implementation.htm

[4] G. D. Morrison, "*A camera-based input device for large interactive displays*," Computer Graphics and Applications, IEEE, vol. 25, pp. 52-57, 2005.

[5] J. K. Parker, R. L. Mandryk, and K. M. Inkpen, "*Integrating Point and Touch for Interaction with Digital Tabletop Displays*," Computer Graphics and Applications, IEEE, vol. 26, pp. 28-35, 2006.

[6] A. Agarwal , S. Izadi, M. Chandraker, and A. Blake, "*High Precision Multi-touch Sensing on Surfaces using Overhead Cameras*." Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop, pp. 197-200, 2007

[7] J. W. Roach, P. K. Paripati, and M. Wade, "*Model-based object recognition using a large-field passive tactile sensor*", Systems, Man and Cybernetics, IEEE Transactions on, vol. 19, pp. 846-853, 1989.

[8] P. T. Krein and R. D. Meadows, "The electroquasistatics of the capacitive touch panel," Industry Applications, IEEE Transactions on, vol. 26, pp. 529-534, 1990.

[9] D. Pasquariello, M. C. J. M. Vissenberg, and G. J. Destura, "*RemoteTouch: A Laser Input User- Display Interaction Technology*", Display Technology, Journal of, vol. 4, pp. 39-46, 2008.

[10] S. Izadi, A. Butler, S. Hodges, D. West, M. Hall, B. Buxton, and M. Molloy, "*Experiences with building a thin form-factor touch and tangible tabletop*", Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on, pp. 181-184., 2008

[11] M. M. Organero, S. K. Opoku, "*Using NFC Technology for Fast-Tracking Large-Size Multi-Touch Screens*", Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), Vol. 2, No. 4 April Edition, 2011.

[12] M. Sallinen, E. Strommer, A. Ylisaukko-oja, "*Application Scenario for NFC: Mobile Tool for Industrial Worker*", Second International Conference on Sensor Technologies and Applications, SENSORCOMM '08, IEEE, 2008, p. 586 – 591.

[13] S. Williams, "*IrDa: Past, Present and Future*", Personal Communications, IEEE Vol: 7, Issue: 1, 2000, p. 11-19

[14] F. M. Chen, X. Y. Jin, Y.J. Liu, H. X. Yang, Z. M. Li, "*Engineering Algorithm of the Atmospheric Attenuation in the Infrared Wireless Communication*", IEEE International Conference on Wireless Communications, Networking and Mobile Computing, WiCom '07, 2007, p. 984 – 987

[15] O. C. Enrique, "*An Introduction to Near-Field Communication and the Contactless Communication API*", NFC Article, Sun Developer Network (SDN), 2008, http://java.sun.com/developer/technicalArticles/javame/nfc/

[16] A. Gupta, M. Srivastava, "*Integrated Java Technology for End-to-End m-Commerce*", Sun Developer Network, May 2001

[17] NFC Forum, "NFC Forum Technology Architecture", 2006 Available: http://www.nfc-forum.org/news/june06_architecture_and_specs/nfc_architecture_schematic