Mobile Device Biometric Touch Gesture Information Used to Give User Identity Evidence

Thomas Ruebsamen, Julia Bayer, Christoph Reich
Cloud Research Lab
Furtwangen University of Applied Science
Furtwangen, Germany
{Thomas.Ruebsamen, Julia.Bayer, Christoph.Reich}@hs-furtwangen.de

***Abstract*—Mobile devices, such as smart phones and tablets, are as popular as never before. Even corporations encourage their employees to use company services such as email and document management services on these kinds of devices. However, the security of mobile devices, especially when used in a professional environment, is still lacking behind compared to company controlled infrastructures. In our previous work, we described a novel system for securing mobile devices while leveraging the potential of seemingly unlimited resources provided by cloud computing. In this work, we extend this idea and focus on user identification using touch gesture analysis as part of the MoSeC architecture. We show that, using artificial neural networks, the analysis of user's touch behavior while using a mobile device can support authentication processes. This technology will also enable the collection of information for tracing legitimate as well as malicious access attempts and we will show how our proposed system may provide digital evidence.**

Keywords*—*Cloud Computing, Digital Forensics, Evidence, Mobile Security, Authentication

I. INTRODUCTION

In our increasingly networked world, IT security plays a more and more important role, especially in enterprise environments, where sensitive business information is stored and processed. Also, the widespread use of mobile devices such as smart phones and tablets in a professional environment is no longer limited to management staff. These devices, when used for work, are usually integrated into the enterprise's IT infrastructure and contain possibly sensitive information. However, these devices are not always under direct control by the enterprise. For example, employees are often encouraged to use their devices at home, which essentially means the risk of loss or theft is significantly higher, because of the uncontrolled environment. The user identification verification and assurance is of importance for enterprises providing their employees access to sensitive data. To support building a chain of evidence or chain of custody additional information about the user's identity is helpful.

Attacks on mobile devices can be classified into two groups: the ones where an attacker is in possession of the device and where attacks are carried out remotely. This paper deals exclusively with the former scenario, where the attacker has gained possession of the device. In such cases, additional access protection mechanisms are of utmost importance to prevent an attacker from accessing data stored on the device as well as enterprise services, which grant mobile devices access to the corporate network.

The most common protection for mobile devices against unauthorized access is locking it down using an additional password or personal identification number (PIN). This mechanism is easy to implement. However, to provide a reasonable addition to device security, the password must be complex enough. Secure complex passwords usually are hardly memorable. Additionally, to lock down the device, the user is usually required to enter the password on every device wake-up or screen activation and after a defined period of time. This is not a user-friendly solution.

In case of a breach of security caused by a mobile device, it is usually most important to gather as much information about the attack as possible. Therefore, evidence gathering systems, which monitor certain system and network related security parameters like unauthorized device accesses or even authorized device accesses, might prove useful for tracing and analyzing.

The remainder of this paper is structured as follows: After this section, we describe related work done in the field of advanced user authentication on mobile devices and continuous authentication. The integration of the gesture identification in the overall framework to give identity evidence can be found in Section III. In Section IV, we give an overview about touch gesture recording and analysis of touch attributes. In the following Section V, we describe our approach to using artificial neural networks (ANN) to analyze collected information in the cloud. In Section VI, we present our evaluation results, followed by a conclusion and the description of possible future research directions in Section VII.

II. RELATED WORK

Wong et al. [1] focus in their work on keystroke analysis for user authentication by analyzing users typing their passwords. The analysis process is carried out using artificial neural networks and the k-nearest algorithm. However, their conclusion is that keystroke analysis is too reliant on the physical condition of the user. In contrast to this work, their approach relies on keyboards being used, whereas our approach focuses on the increasingly popular current smart phone generations, where there usually is only a soft-keyboard, which is used rather sparsely in favor of touch gesture control.

Pannell et al. [2] follow a more comprehensive approach. Besides using keystroke analysis, they include other attributes like running applications and a classification of users into ones with basic computer knowledge or profound knowledge. With this information they build user models. In their system, intruders do not fit those user models and can therefore be detected. By including multiple attributes, the detection rate

 ISBN: 978-1-61208-292-9

can be increased significantly. However, data recording and analysis is performed locally, whereas, in our work, a remote proxy instance is used.

A similar approach is followed by Anand et al. [3], where they try to identify users by using data collected on mobile devices. This data includes the call history and typing patterns. Their main conclusion is that a balance has to be found between the Masquerade Detection Rate, the relation between detected attacks and all attacks, and the time to detect an attack. To accomplish this, some parameters need to be adjusted (e.g., the time frame for data collection).

Imsand et al. [4] try to identify users by analyzing how they are using graphical user interfaces. A simple example is how a user copies text to the clipboard, either by using primarily keyboard shortcuts or context menus. Similar to Pannell's approach, this information is then stored in user profiles, which are compared to the current user's behavior. However, their evaluation didn't seem to be very successful. One reason, which the authors are giving is the small amount of test data sets and the overfitting of their neural network.

Another gesture-based approach for user identification is followed by Guse et al. [5]. It is based on capturing motions of users with 3D acceleration sensors and gyroscopes. However, they acknowledge several problems with this approach. It is difficult to consistently perform the same gestures for identification and those gestures have to be performed in secure locations, because it might be very easy to imitate them. Nevertheless, Guse et al. come to the conclusion that gesture based identification could be a viable alternative to PINs and are significantly less cumbersome than entering PINs. The main difference to our approach is, that we are focusing on smart phone touch gestures without any additional hardware.

SenGuard [6] is a system, which leverages virtualization techniques on mobile devices. It collects information about the voice of the user, GPS location, multi touch gestures and the user's movement. If SenGuard detects possibly non-authorized usage of the device, traditional authentication mechanisms (e.g., PINs) are cut in. SenGuard runs on the device and compared to our approach does not use any external resources. However, the authors recognized power consumption of their system as one of the major issues and try to solve this, by selectively removing data collection sensors the lower the state of charge gets. SenGuard reaches very good user authentication results by combining all of the collected attributes. However, touch gestures on their own are regarded as not sufficient for reliable user authentication by Shi et al. In our approach, touch gestures are used in conjunction with additional continuous authentication mechanisms. As part of the decision making system, touch gestures may very well add to device security.

Schneier et al. [7] describe an approach to logging information while protecting against malicious manipulation to those logs. Several other papers work on similar systems and extend the idea of collecting logging information in a tamper-evident way for making such logs available as evidence. Most of these approaches build on the idea of hash chaining.

III. User Identification Evidence Architecture

In a previous project called Mobile Security by Cloud Computing (MoSeC) [8] at the HFU Cloud Research Lab [9], we addressed these problems by designing a scalable cloud architecture for enhancing mobile device security. Every mobile device is assigned to a proxy instance inside the cloud. The proxy is used to offload performance-intensive tasks to the cloud, where computing resources are seemingly unlimited or at least can be easily scaled out. The mobile device uses a lightweight software agent to communicate with its cloud proxy via a virtual private network (VPN). Besides other information, the agent transmits recorded and aggregated touch gesture profiles of the current user to the cloud, where sophisticated analysis methods are used to detect suspicious activity.

The primary goal of this MoSeC module is providing information for the decision making process for whether a user of a mobile device is authentic or not. In a broader sense, this also means a theft and loss detection for mobiles devices. Additionally, this module provides information for assigning a device to an adequate security level. In MoSeC, security levels are used to control which corporate services and data are accessible by mobile devices. The security level is computed using different information sources like device management, intrusion detection, malware detection, traffic analysis and the gesture analysis depicted in this paper. Depending on the security scores collected in each of these modules devices and their users are classified ranging from *critical* to *highly secure*, resulting in the previously mentioned access control decisions. This effectively leads to a more secure integration of mobile devices (also private devices as in bring-your-own-device) in enterprise environments while protecting sensitive corporate information.

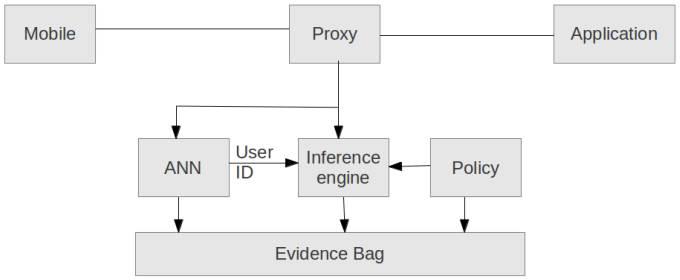

Fig. 1. Architecture Overview

Figure 1 depicts an architectural overview of the continuous authentication process in MoSeC. The Proxy takes control of communication of mobile devices with enterprise applications and services. It also provides the runtime environment for the Inference engine, which uses an ANN specifically trained to detect gestures not performed by the device owner. ANNs were chosen based on their ability to provide a reasonably accurate input about user authenticity to the authentication process. Additionally, company policies are considered by the inference engine (e.g., assigning an appropriate security level depending on the policy, and the analysis results). For further company-wide evidence collection, an evidence bag [10] is provided, which allows retrospective analysis of the collected data (e.g., tracing authorized as well as suspicious accesses for forensics in case of an intrusion). The evidence bag may be used internally, but may also prove useful during the prosecution of security breaches. One key feature of the digital evidence

 ISBN: 978-1-61208-292-9

bag is storing information in a tamper-proof or at least tamper-evident way.

IV. Collecting Input Data on Android

To demonstrate the recording of touch gesture information and the evaluation using an artificial neural network, an Android application has been developed. In this section, we will describe how touch gestures are recorded in our prototype application. Furthermore, a list of different touch attributes are presented and discussed according to their suitability for identifying users of mobile devices.

A. Data Collection

Touch gestures are characterized by the movement of one or more fingers (multi-touch) on a touch-sensitive display and result in actions performed by the device. The Android SDK provides developers with the ability to capture information about touch gestures using the *MotionEvent* class. The most important gestures are scrolling (usually performed by swiping horizontally or vertically over the touchscreen, this is also called flicking), zooming-in (the movement of two fingers away from each other) and zooming-out (the movement of two fingers towards each other). Figure 2 illustrates these basic touch gestures. Starting with scrolling on the left, followed by zooming-in and zooming-out.

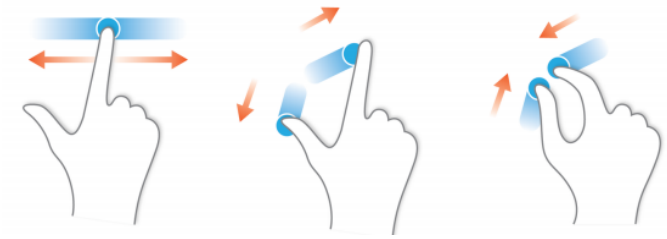

Fig. 2. Multi Touch Gestures [11]

To identify users according to their different touch gesture behavior, a profile of the user has to be created first. In our case, this means that an ANN has to be trained for every user. For this purpose, an Android App has been developed, which records the touch characteristics of users. For gestures like scrolling and zooming, parameters like x/y-coordinates, pressure applied by the finger on the screen, and the length of the gesture are recorded. To get a precise profile of the users touch gestures, this gesture collection course has to be completed multiple times.

B. Touch Attributes

Touch gestures have multiple attributes, which characterize different users. The course App records a total of 87 attributes (e.g. position, no. of fingers, pressure, speed, etc.). Depending on the performed touch gesture, a subset of these attributes is included. For the simple scrolling gesture 23 of the total 87 attributes are used. Zooming-in uses 32 and zooming-out an additional 32. The most significant difference between normal touch gestures like scrolling and multi-touch gestures like zooming is the amount of different attributes, which can be identified. This is reasonable, because normal touch gestures only use one finger and therefore inherently have less attributes which need to be considered.

Additionally, some of the attributes are more significant than others. For example, the overall duration, distance, average pressure applied by the fingers and distance between the fingers during gestures have been identified as very significant and at least in combination possibly unique to a user.

V. Analyzing Touch Gestures Using Neuronal Networks

For making the decision whether or not the current user of a mobile device is the actual owner of the device, an interconnected feed forward neural network is used. In this section we describe the structure, training (supervised) and validation process.

A. Structure of the Neural Network

The neural network consists of three layers (input, hidden and output). The input layer consists of 87 input neurons, each of them representing one of the touch gesture attributes (see Sec. IV). The activation function for the input neurons is the identical function. Additionally, there are three hidden layers with each of them having 60 hidden neurons. The hidden neurons use the logistic function:

$$f(x) = \frac{1}{1 + e^{-3*x}}$$

The output layer consists of two neurons. The value “one” signals that the neural network suspects that the user is authentic, the other neuron signals the opposite case. The number of input and output neurons have been decided according to the input attributes and the two possible results. The amount of hidden layers and number of neurons contained within them have been chosen carefully during the design process of the neural network. The resulting network is rather huge, but the size is justified by the promising results.

B. Training the Neural Network

Artificial neural networks in the MoSeC architecture operate according to a specific life cycle depicted in Figure 3.

In the **Data Collection Phase (DCP)** (1) touch gesture data generated by the user is recorded. Subsequently, the ANN is trained using this data. One essential requirement in this phase is that the device is used exclusively by the legitimate owner. Otherwise, the training data gets tainted, which may lead to the ANN not being able to detect gesture patterns and to a significantly higher rate of false positives.

After the DCP, follows the **Kickoff Learning Phase (KLP)** (2), during which the ANN is trained the first time using the previously collected data. The target network error and desired count of learning iterations is determined empirically. This process is described in Section VI.

After the ANN has been trained, the **Monitoring Phase (MP)** (3) is started. This phase constitutes the main operational phase, where the ANN is actually used to identify unauthorized device access. Touch gesture data, which is collected during the normal usage of the device, is validated against the previously trained ANN during this phase. If irregularities

 ISBN: 978-1-61208-292-9

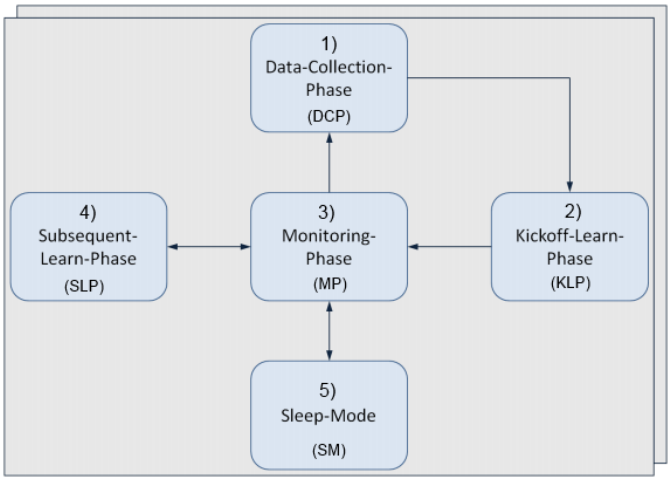

Fig. 3. ANN Lifecycle

are detected, strong authentication mechanisms (such as PIN request, biometric authentication etc.) are triggered. To adapt the ANN to possible changes in behavior of the user, new learning data is collected during the Monitoring Phase.

This new learned data is used to further train the ANN in the **Subsequent Learn Phase (SLP)** (4). To trigger the SLP, different approaches are possible:

- After the user has been authenticated using strong authentication mechanisms (e.g., PINs), the SLP is triggered. This can be done after each or n-th authentication in a given duration.
- Regardless of the authentication state of the user, the SLP is triggered in set intervals.
- The SLP is triggered manually by an administrator either to reset the state of the ANN.

Selecting data, which is being used during the SLP, can be selected using different strategies:

- Learning all previously recorded data including the data collected during the KLP may prove problematic, because more recent changes in behavior are not weighed strong enough (overfitting of old data).
- Only recently recorded data is used to generate training lessons for the ANN. This approach reacts to behavior changes in a very agile way, but may "forget" old behavior too soon.
- The middle-ground between the previous two approaches is using newly acquired data as well as old data collected over a given duration, merged into a lesson for the ANN.

After the SLP is complete, the operation mode is switched back to the MP. To address the problem of a user changing in behavior over time, a backup AAN can be trained in parallel. When needed, the running ANN is replaced by the backup ANN. Another approach would be to introduce a forgetting-factor. However, in our work we focused on the first approach.

The last operation mode is **Sleep Mode (SM)** (5). This deactivates the gesture authentication process temporarily. This mode is supposed to be used by an administrator or by the user after he has been authenticated using a strong mechanism. But, after switching to this mode, the security level of the mobile device is reduced significantly, which results in blocked access to corporate services and information.

VI. Evaluation

As previously mentioned, an application which records information about touch gestures has been developed to show whether or not using artificial neural networks on the cloud for identifying mobile device users is a feasible approach. In this section we will describe our evaluation methodology and results.

The pattern recognition for the detection of unauthorized users is done using neural networks. During the implementation process "Membrain" [12], a neural network editor and simulator which also provides JAVA bindings, has been used for editing and simulating neural networks. The Samsung Nexus S served as a development platform.

A. Results

To test the effectiveness of our approach, 6 persons were selected for testing the application. The number of people has been chosen arbitrarily. Table I shows how often each test person completed the course for recording touch profile information sorted by whether the data is collected for network training or actual testing.

TABLE I
Number of Course Passed by Test Person

	Courses Passed (learning)	Courses Passed (checking)
Person 1	15	15
Person 2	15	15
Person 3	22	15
Person 4	76	15
Person 5	20	20
Person 6	50	15

An example for how data sets for the training phase of the neural network look like is shown in Table II. The attribute columns contain the normalized values of the attributes extracted from the touch gesture (e.g., coordinates, length, applied pressure etc.). The learning behavior of the network is heavily dependent on the amount and quality of learning data as well as the ratio between positive and negative learning data and the network error. Because our approach to training the network involves observed learning, a learning data set also contains the values of the output neurons. Positive learning data sets have the value "1" for the "yes" output neuron and negative ones vice versa. It is obviously very important to find the right ratio between positive and negative patterns to get the best results. To get negative patterns for each person, patterns from the other test persons were included.

To find a suitable ratio between the amount of positive and negative patterns, 4 training passes have been carried out. The amount of positive learning patterns remains constant for each

 ISBN: 978-1-61208-292-9

TABLE II
EXEMPLARY LESSON STRUCTURE

Attribute 1	Attribute 2	Attribute 3	...	Output YES	Output NO
0.4979	0.7892	0.1983	...	1	0
0.5043	0.7721	0.2112	...	1	0
0.3094	0.9798	0.6983	...	0	1
0.3176	0.9943	0.7002	...	0	1

TABLE III
POSITIVE AND NEGATIVE LEARNING PATTERNS COUNT BY PASS

	Positive Pattern Count	Negative Pattern Count			
		Pass 1	Pass 2	Pass 3	Pass4
Person 1	15	10	10	15	20
Person 2	15	10	10	15	20
Person 3	22	10	15	20	25
Person 4	76	55	60	65	70
Person 5	20	10	15	20	25
Person 6	50	35	40	45	50

pass, while the amount of negative patterns has been varied arbitrarily. Table III depicts the evaluated ratios.

Another parameter, which had to be evaluated is the desired network error. The lower the network error, the longer it takes to train the ANN. Therefore, several different values (3%, 6%, 9%, 12% and 15%) have been evaluated using different ratios of learning patterns. In our experiments we came to the conclusion that a ratio of 1:1 regarding positive and negative learning data results in a minimal occurrence of false positives and false negatives during the MP. Figure 4 shows the trend for the ratios described in Table III. Figure 4 also depicts the summed up values (and therefore seems unusually high) for false-positives and shows how results change depending on different combinations of positive-negative learning patterns. Therefore we restricted our further evaluation to the use of the 1:1 ratio for positive and negative learning patterns. Based on this information, Table IV shows a detailed evaluation of the detection rate for unauthorized users. It shows for each user and lesson the output values of the neurons: The higher value wins. For example, for person 1 and the learning data of person 1 (first row, column) the YES-neuron fires 70 times whereas the NO-neuron fires 30 times. Therefore, the user is successfully authenticated. However, we also detected some false positives. For instance, person 2 is the authorized user and checking patterns of person 5 fed to the neural network results in a false positive. The same applies to person 5.

TABLE IV
RESULTS OF EVALUATION WITH NETWORK ERROR 0.12

	P1	P2	P3	P4	P5	P6
P1(learn)	72/30	7/95	32/41	12/85	19/75	27/93
P1(verify)	64/39	28/38	24/60	10/89	31/71	31/86
P2(learn)	18/90	85/4	35/66	9/78	16/84	1/96
P2(verify)	15/85	44/12	39/67	1/80	16/86	52/67
P3(learn)	28/78	18/93	74/21	61/73	10/80	42/73
P3(verify)	26/86	26/60	69/19	5/75	13/69	33/70
P4(learn)	16/87	3/99	22/77	71/45	18/67	19/75
P4(verify)	14/86	18/56	19/88	92/24	18/79	36/61
P5(learn)	21/91	26/88	7/88	34/78	88/20	9/92
P5(verify)	23/88	58/21	14/93	32/89	83/63	6/89
P6(learn)	20/81	16/95	47/79	4/79	18/66	75/20
P6(verify)	10/87	23/79	32/98	1/61	87/22	75/27

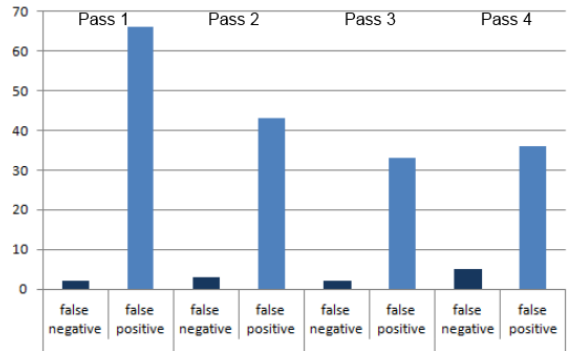

Fig. 4. False Positives and False Negatives for Different Ratios

B. Data Collection Optimisation and Compression

Currently, learning data sets as well as actual input for the neural network during normal usage of the mobile device is collected and recorded into simple CSV files. This may prove to be a problem regarding the size of collected data. Transmission of data using WLAN or 3G networks is a very expensive task in terms of power consumption. Therefore, compression techniques may need to be applied on the device to reduce the size of transmitted data. However, this has not been considered to date.

VII. CONCLUSION

In this paper, we analyzed whether or not and to which certainty users of mobile devices can be authenticated based on their touch gesture attributes. As a part of the MoSeC architecture this module is not required to achieve a 100% success rate, but rather serves as a supporting mechanism for detecting possibly unauthorized accesses in combination with other techniques. One major aspect of this system is to offload as much resource-intensive tasks as possible into the cloud to preserve battery power. Therefore, we use the mobile device only to record and save touch gesture information. After transmitting the data to a proxy virtual machine in the cloud, a neural network evaluates this data. Depending on the result, the agent on the mobile device is informed on which action must be taken (e.g., force re-authentication using strong mechanisms like passwords). Additionally, the authentication information is used to trace usage of the device retrospectively for evidence collection, when a breach of security is detected.

As we have shown, touch gesture analysis based on artificial neural networks is a suitable solution for detecting unauthorized access to mobile devices. However, the inherent uncertainty of the analysis results provided by the network need to be considered. We have shown that by choosing a 1:1 ratio between positive and negative learning patterns relatively stable results can be reached. Another important factor is the training time. Because users might change their touch gesture behaviour over the time, the network has to be re-trained in defined intervals.

In our future work, we focus on widening our pool of test users, to achieve even more stable results. Also, further optimisation needs to be considered. This includes the integration of tamper-proof recording of touch gesture information, compression of data as wells as finding the right balance

 ISBN: 978-1-61208-292-9

between transmission frequency and power consumption due to active WLAN or 3G. Also, network traffic generated by our solution is a big concern and needs to be measured and quite possibly optimized in our application to reduce resource consumption to a minimum, which in turn would also reduce power consumption.

REFERENCES

[1] F. W. M. H. Wong, A. Supian, A. Ismail, L. W. Kin, and O. C. Soon, "Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm," in Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, vol. 2, 2001, pp. 911–915.

[2] G. Pannell and H. Ashman, "User modelling for exclusion and anomaly detection: a behavioural intrusion detection system," in Proceedings of the 18th international conference on User Modeling, Adaptation, and Personalization, ser. UMAP'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 207–218.

[3] A. Gupta, D. Gupta, and N. Gupta, "Infosec-mobcop framework for theft detection and data security on mobile computing devices," in Contemporary Computing, ser. Communications in Computer and Information Science, S. Ranka, S. Aluru, R. Buyya, Y.-C. Chung, S. Dua, A. Grama, S. Gupta, R. Kumar, and V. Phoha, Eds. Springer Berlin Heidelberg, 2009, vol. 40, pp. 637–648.

[4] E. Imsand, D. Garrett, and J. Hamilton, "User identification using gui manipulation patterns and artificial neural networks," in Symposium on Computational Intelligence in Cyber Security (CICS 2009), 2009, pp. 130–135.

[5] D. Guse, N. Kirschnick, S. Kratz, and S. Möller, "Gesture-based user authentication for mobile devices," in Proc. MobileHCI 2011, Workshop on Body, Movement, Gesture & Tactility in Interaction with Mobile Devices, 2011.

[6] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, "Senguard: Passive user identification on smartphones using multiple sensors," in WiMob'11, 2011, pp. 141–148.

[7] B. Schneier and J. Kelsey, "Secure audit logs to support computer forensics," ACM Trans. Inf. Syst. Secur., vol. 2, no. 2, pp. 159–176, May 1999.

[8] T. Ruebsamen and C. Reich, "Enhancing mobile device security by security level integration in a cloud proxy," in CLOUD COMPUTING 2012, The third International Conference on Cloud Computing, GRIDs, and Virtualization, 2012, pp. 159–168.

[9] "HFU Cloud Research Lab," http://www.wolke.hs-furtwangen.de/, [retrieved: july, 2013].

[10] P. Turner, "Unification of digital evidence from disparate sources (Digital Evidence Bags)," Digital Investigation, vol. 2, no. 3, pp. 223–228, Sep. 2005.

[11] "Wikipedia on Multi Touch Gestures," http://en.wikipedia.org/wiki/Multi-touch, [retrieved: july, 2013].

[12] "Membrain," http://www.membrain-nn.de/, [retrieved: july, 2013].

 ISBN: 978-1-61208-292-9