Digital Fingerprinting Based on Keystroke Dynamics

A. Ahmed, I. Traore and A. Almulhem

Department of Electrical and Computer Engineering
University of Victoria, Victoria, BC, Canada
e-mail: {aahmed, itraore, almulhem}@ece.uvic.ca

Abstract

Digital fingerprinting is an important but still challenging aspect of network forensics. This paper introduces an effective way to identify an attacker based on a strong behavioral biometric. We introduce a new passive digital fingerprinting technique based on keystroke dynamics biometrics. The technique is based on free text detection and analysis of keystroke dynamics. It allows building a behavioral model from passively collected data, and identifying users based on a very minimal amount of data.

Keywords

Network forensics, digital fingerprinting, biometrics technologies, keystroke dynamics, free text detection of keystrokes.

1. Introduction

In recent years, several studies have emphasized the unprecedented growth in security threats from multiple and varied sources. Traditional protection mechanisms such as access control or firewalls, although valuable, have shown their limits in dealing with these threats. In this context most organizations put a particular emphasis on investigating hacking incidents, allowing them to assess the impact of a penetration and take appropriate measures to ensure secure service continuity, and prevent future similar incidents. The process of analyzing network traffic with the purpose of identifying malicious activities and characterizing their authors is referred to in the literature as Network Forensics (Ranum, 1997). Not surprisingly, network forensics is often confused with the traditional field of computer forensics. In fact, the two are arguably more different than similar (Berghel, 2003).

In particular, computer forensics evolved over time following the standard methodologies used by law enforcement in investigating crimes in real life. It is basically concerned with the thorough investigation of computers found in crime scenes. The computer itself is not necessarily a victim of an attack, it is rather a tool used by a criminal. On the other hand, network forensics primarily emerged due to network hacking activities (Ranum, 1997). In this case, the target of an attack is the computer itself. Additionally, the primary objective of the investigation is not necessarily a legal action. Instead, the objective may reduce to typical computer security objectives such as availability and accountability. Investigating computer attacks is a challenging task that usually requires analyzing a mass of diverse data. Few organizations are well equipped to tackle this challenge mainly because there are few tools accessible to organizations that will help determine the impact of hacking incidents.

An important but extremely difficult aspect of network forensics is attacker profiling (Ahmed and Traore, 2005). This consists of maintaining and using information about the attacker to characterize and recognize him. The profile computed can be used as a digital fingerprint equivalent to physical fingerprint collected in crime scenes. We propose in this project a new approach to user profiling that is based on biometrics. The profiles computed in this case are more accurate since they are based on distinctive biological characteristics of users. The computed biometrics profile can be used to track an attacker throughout computing sessions, allowing identifying easily and accurately compromised systems within the environment.

Different types of biometrics are currently available in the market, and are widely used in various security applications. These technologies can be classified into two categories, "physiological biometrics" and "behavioral biometrics". Physiological biometrics identify the user based on physiological characteristics, such as fingerprints and eye retina/iris scanning, whereas behavioral biometrics depend on detecting the behavioral features of the user, such as signature, voice, and keystroke dynamics. The utilization of biometrics technology, however, has so far been limited to identity verification in authentication and access control systems. Hence, important security applications such as network forensics systems have been left out of this technology. The main reason is that traditional biometrics technologies cannot be collected passively which is necessary in network forensics. Behavioral biometrics based on mouse or keystroke dynamics are exceptions. We present in this paper a passive digital fingerprinting technique based on keystroke dynamics biometrics.

The rest of the paper is structured as follows. In Section 2, we give an overview of digital fingerprinting, and discuss the feasibility of biometric-based fingerprinting; we also review related work on keystroke dynamics-based fingerprinting techniques published in the literature. In Section 3, we present our approach for passive fingerprinting based on keystroke dynamics. In Section 4, we present and discuss experimental results. Finally, in Section 5, we make some concluding remarks.

2. Passive Fingerprinting

2.1. Background on Digital Fingerprinting

Digital fingerprinting consists of gathering knowledge about the attacker methods and tools, as well as his patterns of behavior. Traditionally, digital fingerprinting has targeted primarily the identification of the operating system, services, and applications of a remote host based on data collected through network sniffing or probing. Traditional fingerprinting techniques rely on the assumption that every operating system's IP stack and applications have unique characteristics. As such, querying or probing the target system and comparing its response against a database of known response can establish its specific characteristics.

Fingerprinting techniques are broadly categorized into two forms: active fingerprinting and passive fingerprinting. Active fingerprinting typically consists of (actively) sending a sequence of probe packets to the target host in order to learn more about the attacker. In this case the risk of being detected is greater for the user of such technique. Passive fingerprinting, in contrast, consists of characterizing the hacker without his knowledge, typically by (passively) observing or sniffing traffic generated by the target host. Passive fingerprinting is the preferred method for network forensic analysts because it gives them the opportunity to observe and study the attacker in an unbiased context, without the hacker knowing that he is being monitored.

Existing fingerprinting techniques have so far mostly focused on learning about the attacker methods and tools. Limited attention has been paid to profiling the attacker himself, by identifying unique and distinctive characteristics that could be used to accurately recognize the attacker. Establishing a unique and distinctive profile for the hacker is an important step towards bridging the gap between the current practice of network forensics and the traditional area of digital forensics. Because this would allow associating with the collected hacking evidence a unique signature or fingerprint that if properly handled could be admissible in court, and used to prosecute more effectively authors of hacking incidents.

In order to successfully profile attackers using passive fingerprinting, there is a need to look into invariant features that can distinguish one attacker from another. One of these features is the attacker's operating system (OS). In fact, OS passive fingerprinting is a well-established technique that can be used for OS identification by examining the headers of certain packets (especially SYN packets). At best, however, the identification of an attacker's OS is not specific enough to hold him/her accountable. Therefore, it is unlikely that the OS identification alone would lead to an accurate identification of the attacker. Taking OS fingerprinting a step further, physical fingerprinting is a promising technique that has been recently proposed in (Kohno *et al.* 2005). The technique enables the identification of a specific device, rather than its operating system. In essence, this technique takes advantage of the microscopic deviations in a device's hardware clock; i.e. clock skews. The authors show that these deviations are in fact unique across different devices. Hence, they can be used to uniquely identify different devices physically.

The next logical step is to identify the attacker (user) himself/herself. In this case, invariant features associated with the attacker himself/herself are needed. There are several possible features that can be used for such task. In (Almulhem and Traore, 2008), the authors proposed using two features; namely the attacker's linguistics and activity time. The authors showed that clues about the attacker's language and consistent time patterns of activities would enhance the attacker's profiling and identification process.

2.2. Biometric-based Fingerprinting

Traditionally, biometric systems used for authentication or access control operate in two modes: the enrollment mode and the recognition mode. In the first mode, raw biometric data is acquired and processed to extract the biometric features representing the characteristics, which can be used to distinguish between different users. This conversion process produces a processed biometric identification sample, which is stored in a database for future recognition needs. In network forensics analysis, enrollment is not always feasible, simply because a large number of hackers are outsiders, although inside attacks still represent a significant portion of hacking incidents. The focus in network forensics is on recognition, by being able to build from collected data a unique profile for the hacker. Note, however, that storing the initial profile formed for a hacker can speed up future recognition tasks. In this regard, the first recognition session can be considered as a form of enrollment.

There are two kinds of biometric recognition processes, namely verification and identification, described a follows: 1. *Verification* involves one-to-one matching by comparing the processed data sample against the enrolled sample of the same user, resulting in a match or non-match. 2. *Identification* consists of matching the processed sample against a large number of enrolled samples by conducting a 1 to *N* matching to identify the user, resulting in an identified user or a non-match.

Forensics analysis typically uses identification. For instance, in traditional forensics field the physical fingerprint collected in a crime scene is matched against an existing fingerprints database built and updated over time. Although digital profiling follows the same concept, it faces many practical implementation challenges. First there is no existing hacker profile database. Even though such database can be built progressively, it would require a lot of cooperation between Internet service providers to be effective against outsider attacks. Nonetheless a local database can be implemented by an organization in order to combat insider threat. Another key challenge concerns the collection of the forensics data. In effect most traditional biometrics systems require, in general, special hardware device for data collection and an active involvement of the user who is asked to provide some data sample that can be used to verify his identity. These represent limiting factors for network forensics analysis, which should cover a wider networking scope and should be able to operate passively and transparently.

In contrast with traditional biometrics systems, keystroke dynamics, can suitably be used for network forensics and intrusion detection, because they can be collected using readily available human-computer interaction devices (i.e., mouse and keyboard) (Ahmed and Traore, 2005). We propose in this work a passive fingerprinting technique that is based on free-text detection of keystroke dynamics; we discuss related work in the next section.

2.3. Fingerprinting based on Free-text Detection of Keystrokes

Keystroke dynamics recognition systems measure the dwell time and flight time for keyboard actions (Dowland *et al.* 2002). The raw data collected for keystroke includes the time a key is depressed and the time the key is released. Based on this data, the duration of keystroke (i.e., length of time a key is depressed), and the latency between consecutive keystrokes are calculated and used to construct a set of digraphs, tri-graphs or *n*-graphs producing a pattern identifying the user. Traditionally, keystroke dynamics recognition has been done by asking the user to type a pre-defined word or set of words in order to get reasonable amount of data for the identification (Bergadano *et al.* 2002, Bleha *et al.* 1990, Brown and Rogers, 1993, Gaines *et al.* 1980, Legget and Williams, 1988). During the enrollment process, the user is also required to enter the same fixed text. For passive fingerprinting, we need to be able to detect the user without requiring him to enter a predefined message or text. So we need to be able to detect dynamically a text freely typed by the user. Research on free text detection of keystroke dynamics has so far been limited. Few related works have been published in the literature including (Dowland *et al.* 2002, Guneti and Picardi, 2005, Monrose and Rubin, 1997).

In (Monrose and Rubin, 1997), the authors collected typing samples from 42 users over a period of 7 weeks, during their routine computing usage, without any particular constraint. The collected timing data consist of keystroke durations and latencies. To recognize individual users, they use a clustering algorithm to partition the data into cluster domains. The typing speed or number of words typed per minute in a given profile is used as the clustering criteria. Various distance measures were studied along with the clustering scheme, including Normalized Euclidian distance, and weighted and non-weighted maximum probability measures. Although the authors obtain 90% correct classification for fixed text detection, they obtain at best (using the weighted probability measure) only 23% correct classification for free text detection. In (Dowland *et al.* 2002), the authors collected typing samples by monitoring users during their regular computing activities, without any particular constraints imposed on them. A user profile is determined by calculating the mean and standard deviation of digraph latency and by considering only the diagraphs occurring a minimum number of times across the collected typing samples. By collecting and analyzing data for five users they achieve correct acceptance rates in the range of 60%.

The results achieved in these earlier works, although encouraging, are highly insufficient in network forensics analysis, especially if one expect using corresponding profiles as legal evidence. Recently in (Guneti and Picardi, 2005), the authors significantly improve on the performance of earlier works by using the degree of disorder of an array to discriminate between two different typing samples. The degree of disorder of an array is computed by summing the distances between the positions of its elements, with respect to the positions of the same elements in the ordered array. Based on the degree of disorder, they define sophisticated distance measures that they use to compare different typing samples. They evaluated their technique through a set of experiments involving 205 individuals, achieving a False Acceptance Rate (FAR) of 0.00489% and a False Rejection Rate (FRR) of less than

5%. We propose in this work a new approach for free text detection of keystroke dynamics based on a keyboard layout approximation technique. Although our approach is different from the one proposed by Guneti and Picardi, it achieves comparable performance.

3. Free Text Detection based on Keyboard Layout Mapping

3.1. Detection Approach and Architecture

The main challenge faced in free text detection is the need to develop a technique that helps in minimizing the amount of data used to establish the user profile, by only extracting the needed information from the information detected so far in an active session. We propose in this work a free text detection approach based on keyboard layout approximation. Our approach utilizes neural networks to simulate and analyze the user behavior based on the detected digraphs. Figure 1 shows the detector architecture and the flow of data in enrollment and detection modes. In enrollment mode extracted digraphs are encoded with a mapping algorithm. This process is needed in order to convert key codes into another representation, which is relevant and meaningful as an input to the neural network.

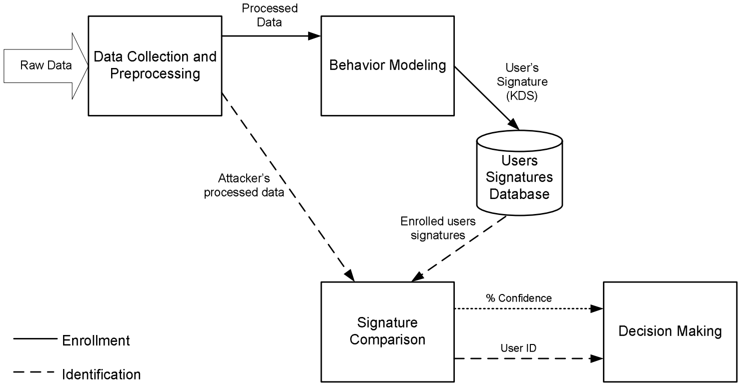

Figure 1: Detector architecture showing the flow of data in enrollment and detection modes.

Raw data collected through various users sessions are processed and converted to a set of digraphs. In enrollment phase the data will be sent to a behavior-modeling unit. The function of this unit is to build a model for the user behavior, which can be represented by a set of numbers. The output of this unit is the keystrokes dynamics signature (KDS), which is stored in a database for future comparisons. When a suspicious session is monitored, the collected data will be processed and sent directly to the signature comparison unit for identification. This unit utilizes a signature comparison technique, which is capable of finding similarities in behaviors and identifying the user to whom this data belongs. The output of this unit is sent to a decision-making module in the format of *(UID, CR (%))* where *UID* is the identified user *ID*, and *CR (confidence ratio)* is a ratio in percentage format representing how confident the system is that the monitored data belongs to a specific user.

One of the important factors to be considered in the enrollment phase is the amount of data needed to enroll the user and create a signature modeling his behavior. The aim is to minimize the enrollment time as much as possible without affecting the accuracy of the system in detection mode. Since the key codes do not reflect the relation between the keys like their absolute or relative positions, the pre-processing stage, illustrated in Figure 2, should include a mapping mechanism in order to convert those sets of keystrokes into numbers, which are suitable to train the neural network with. Such numbers should reflect a specific characteristic for each key and its relation to other keys. The proposed approach is based on a keyboard layout mapping technique. The main idea underlying this technique is to replace each key code by its location on a previously identified keyboard layout. In our implementation we use the QWERTY keyboard layout; each key will be presented by a pair of numbers representing its *x* and *y* location.

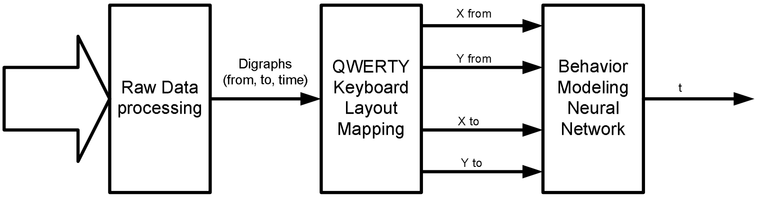
Raw Data processing
Digraphs (from, to, time)
QWERTY Keyboard Layout Mapping
X from
Y from
X to
Y to
Behavior Modeling Neural Network
t

Figure 2: Preprocessing of keystrokes data and modeling the user behavior in enrollment mode.

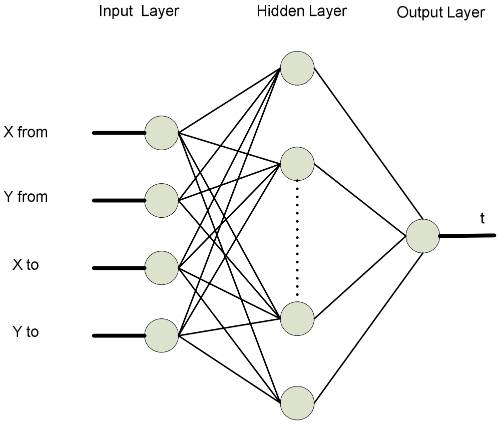

Figure 3: Architecture of the behavior modeling neural network.

Neural networks are used to model the relation between the keys and the time used to move between them. Figure 3 shows the neural network architecture used. The network is a feed forward multi layer perceptrons (MLP). The number of input nodes is 4 representing the *x* and *y* location of the two digraph keys. The output layer consists of one node, which represents the time needed to perform the digraph. The hidden layer consists of 12 nodes. The back propagation technique is used to train the network. A neural network will be trained for each of the enrolled users. The weights of the trained network represent the user keystrokes dynamics signature (KDS). Those signatures will be saved in a repository for the detection process.

3.2. Forensics Analysis

Forensics analysis fundamentally involves the detection of the user identity using a set of collected data. This process is usually conducted by comparing the signature of the unknown user to the set of signatures stored in the database. This is not a straightforward process, as attackers are not expected to produce large amount of data. Furthermore in most cases the data collected in detection mode is very small compared to the amount of data used to create each of the users' signatures. It is also not possible to force the attacker to provide a specific amount of data or limit him to specific types of actions. This small amount of data cannot be used to build a model and compare it directly to the stored signature. Doing so will require collection of large amount of data and increase the time to detect. In order to overcome this problem another preparatory stage, described in Figure 4, is needed before a detection process can take place. In this stage the same network used for behavior modeling is used. The network is loaded with the weights stored for each user and then fed with a set of inputs representing all possible *from-to* key combinations. Those inputs and their corresponding outputs are used to train the detection neural network as shown in Figure 4.

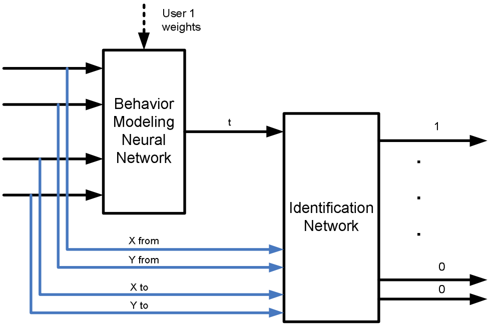

Figure 4: Training the identification network with data representing enrolled users' signatures produced from their behavior modeling neural network.

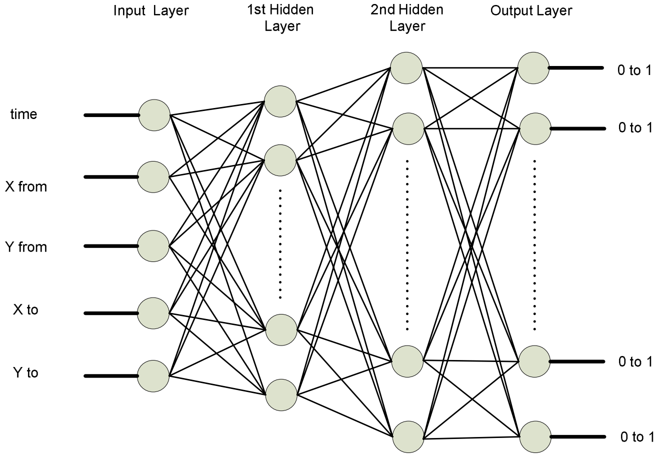

Figure 5: Architecture of the identity detection neural network.

The detection neural network (Figure 5) consists of 4 layers. The input layer consists of 5 nodes representing the complete digraph data and the elapsed time, which corresponds to the output of the behavior-modeling network. The output layer consists of *N* nodes each representing one of the enrolled users. The (two) hidden layers consist of 10 and *N* nodes, respectively. The number of nodes in the network will scale according to the number of enrolled users. Two nodes will be added to the network for each new user. The training process consists of feeding the detection network with the pre-defined set of digraphs and their corresponding output from the behavior-modeling network. The training output will be set to 1 for the node representing the current user and to 0 for the other nodes representing all other enrolled users. For each of the enrolled users a similar set will be created, in each cycle the behavior-modeling network will be loaded with the weights representing the signature of the current user.

Figure 6 gives an example of the model built by the behavioral neural network. The 3-D surface reflects the time needed to perform a digraph action from a *(x,y)* keyboard position to the location of letter "a" on the keyboard for one of the enrolled users. The figure also shows digraphs calculated from an attacking session. The identification network should be able to detect the visible deviation from surface and try to match those points to the closest model. During the detection mode the collected data set is processed and passed directly to the trained detection network. Averages of the outputs for each of the output nodes are calculated. The higher the value at the output node, the more confident the system that this data belongs to the user linked to this node.

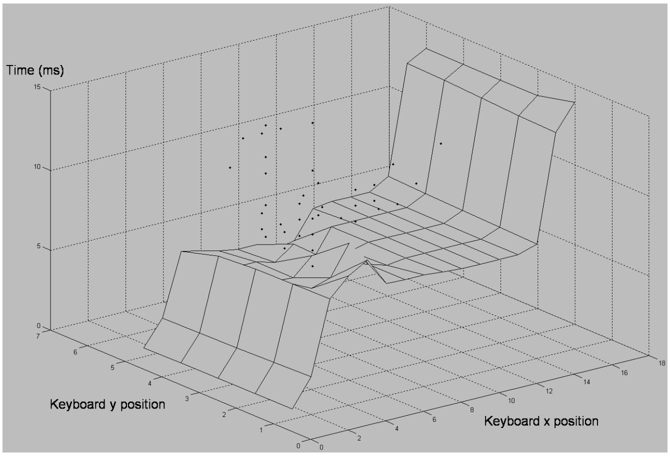

Figure 6: Attacker's data compared to the model built by the behavior modeling network. The surface represents user 1's model. The scattered points belong to user 2. Noticeable deviation from the surface indicates a behaviour difference.

4. Experiments

We conducted a wide range experiment involving twenty-two participants. 16 males and 6 females, with varying computer skills and ages ranging from 13 to 48 years, were involved in this experiment. Each user was asked to install the client software on his workstation and to use the workstation for his daily use throughout the duration of the experiment. The experiment lasted for 9 weeks and we were able to collect an average of 119979 digraphs per user. The overall performance achieved by our detector includes a false acceptance rate (FAR) of 0.0152% and a false rejection rate (FRR) of 4.82%.

5. Conclusions

Network forensics focuses on the capture and analysis of the actions of an intruder for investigative purpose. Using our proposed biometrics system it would be possible to derive from intrusive actions a unique profile that can be used to identify the perpetrator. This may be used to track perpetrators across cooperating sites, fight against insider threat or could be used as evidence against malicious users in court. The main challenge in this research is that the evidence has to be collected and preserved correctly to pass the admissibility test - a screening process by the court. The signature created by the technique presented in this paper can be considered as a strong identity proof as it is based on a human behavioral biometric which is unique for each user. Another issue is the accuracy of the computed profiles. Although the performance results achieved above are encouraging, we still need to improve them significantly for the proposed technology to achieve widespread acceptance.

References

Ahmed, A.A.E, and Traore, I. (2005), "Detecting Computer Intrusions Using Behavioral Biometrics", *Proc. of 3rd Ann. Conf. on Priv., Sec. and Trust*, St. And. NB, Canada, pp 91-98.

Almulhem, A., and Traore, I. (2008), "Profiling distributed connection chains", *International Journal of Communication Networks and Distributed Systems*, 1(1), pp 4-18.

Bergadano, F., Guneti, D., and Picardi, C. (2002), "User Authentication through Keystroke Dynamics", *ACM Trans. on Info and System Security*, Vol. 5, No. 4, Nov 2002, pp 367-397.

Berghel, H. (2003), "The discipline of internet forensics", *Com. of the ACM*, 46(8), pp 15-20.

Bleha, S., Slivinsky, C., and Hussein, B., (1990), "Computer-access Security Systems using Keystroke Dynamics", *IEEE Trans. Patt. Anal. Mach. Int.*, PAMI-12, 12, pp 1217-1222.

Brown, M., and Rogers, S.J. (1993), "User Identification Via Keystroke Characteristics of Typed Names Using Neural Networks", *Int. J. Man-Machine Studies*, Vol. 39, pp 999-1014.

Dowland, P., Furnell, S., and Papadaki, M. (2002), "Keystroke Analysis as a Method of Advanced User Authentication and Response", *Proceedings of the IFIP TC11 17th Int. Conference on Information Security: Visions and Perspectives*, May 07-09, 2002, pp 215-226.

Gaines, R., Lisowski, W., Press, S., and Shapiro, N. (1980), "Authentication by Keystroke Timing: Some Preliminary Results", *Rand. Report R-256-NSF*, Rand Corporation.

Guneti D., and C. Picardi (2005), "Keystroke Analysis of Free Text", *ACM Transactions on Information and System Security*, Vol. 8, No. 3, Aug. 2005, pp 312-347.

Legget, J, and Williams, G. (1988), "Dynamic Identity Verification via Keystroke Characteristics", *Int. J. Man-Mach. Stud.,* 35, pp 859-870.

Monrose, F., and Rubin, A. (1997), "Authentication via keystroke dynamics", *Proceedings of the 4th ACM conf. on Comp. and comm. security*, Apr 1-4, Zurich, Switzerland, pp 48-56.

M. Ranum (1997), "Network Forensics: Network traffic Monitoring", *Tech. Rep.*, NFR Inc.

Kohno, T., Broido, A., and Claffy, K.C. (2005), "Remote physical device fingerprinting", *IEEE Transactions on Dependable and Secure Computing*, 2(2), Apr.-Jun., pp 93-108.