Yousefnezhad, Narges; Madhikermi, Manik; Främling, Kary

MeDI

Published in:
Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018

DOI:
10.1109/INDIN.2018.8472080

Published: 27/09/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Yousefnezhad, N., Madhikermi, M., & Främling, K. (2018). MeDI: Measurement-based Device Identification Framework for Internet of Things. In *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018* (pp. 95-100). Article 8472080 IEEE. https://doi.org/10.1109/INDIN.2018.8472080

MeDI: Measurement-based Device Identification Framework for Internet of Things

Narges Yousefnezhad*, Manik Madhikermi*, Kary Främling†
*Department of Computer Science, Aalto University, Espoo, Finland
†Department of Computing Science, Umeå university, Umeå, Sweden
*{firstname.lastname}@aalto.fi, †kary.framling@umu.se

***Abstract*—IoT systems may provide information from different sensors that may reveal potentially confidential data, such as a person's presence or not. The primary question to address is how we can identify the sensors and other devices in a reliable way before receiving data from them and using or sharing it. In other words, we need to verify the identity of sensors and devices. A malicious device could claim that it is the legitimate sensor and trigger security problems. For instance, it might send false data about the environment, harmfully affecting the outputs and behavior of the system. For this purpose, using only primary identity values such as IP address, MAC address, and even the public-key cryptography key pair is not enough since IPs can be dynamic, MACs can be spoofed, and cryptography key pairs can be stolen. Therefore, the server requires supplementary security considerations such as contextual features to verify the device identity. This paper presents a measurement-based method to detect and alert false data reports during the reception process by means of sensor behavior. As a proof of concept, we develop a classification-based methodology for device identification, which can be implemented in a real IoT scenario.**

***Index Terms*—Internet of Things, Device identification, Device profiling, Identity theft, Smart campus**

I. Introduction

Rapid growth of the Internet and online banking forces society to face frauds, especially financial ones. In the past couple of years, frauds (e.g. data breaches and identity thefts) have posed as a major challenge for plenty of IT companies due to the increasing number of online data breaches. Based on statistics released by Identity Theft Resource Center (ITRC) and CyberScout, 791 data breaches were reported in the U.S. by the end of June 2017. In addition, research conducted by the Statistic Brain Research Institute declared that credit card fraud totalled 5.55 billion dollars worldwide in 2016. In fact, credit-card fraud (33% of all frauds) is considered the most common form of identity theft reported [1]. The Federal Trade Commission (FTC) in 2001 reported identity theft as one of the fastest growing crimes in the United States. Identity theft is defined as the appropriation of someone else's personal or financial identity to commit fraud or theft [2]. It is typically initialized by stealing an identity (or creating a fake one) and terminates with illegally using the fake identity to to commit crimes or to gain access to the victim's services [1].

Internet of Things (IoT) as a vision of future Internet, where the barrier between the physical world and digital information will be removed, could confront the same statistics and threats in the near future. The Internet made theft of personal or financial identity easier, because of its ability to accumulate vast amounts of information electronically [2]. In the same vein, IoT technology makes it much easier since beside digital communication among humans, IoT also connects the smart devices owned by humans and aggregates the large volumes of information leaking from each device. Additionally, in IoT, it is more difficult to prevent or detect threats since data in IoT could be under attack not only remotely, but devices may also be physically accessed in many environments.

In IoT environments such as smart cities, large numbers of physically accessible devices are deployed all over the city. As a result, the physical security of these devices is of highest priority since *poor physical security* related e.g. to device easy to disassemble, access by software via USB ports, and removable storage media are considered top security threats [3]. In addition to physical security, providing secure communication sets another important challenge. Currently for secure communication, a pair of private and public keys or certificate is installed on devices to provide device identity. But these certificates have their drawbacks. For instance, unauthorized access of these certificates might result in identity theft and allows to communicate or send false data to other devices. This might impact the overall decision since decisions are made based on the aggregation of all sensor data and any single false data might affect the entire system. In order to cope with such identity theft, the system must be capable of identifying these kinds of threats.

Paper Contribution: To detect IoT devices uniquely, an efficient identity management approach should be defined. We tackle this problem by presenting Measurement-based Device Identification (MeDI) framework based on device behavior or device profile. MeDI does so by monitoring the data packets coming from smart devices to protect the server from receiving and spreading false data. Each device behavior is defined by its features which are characterized by three profiling methods specified in the proposed framework. For proof-of-concept, a smart campus, in the subcategory of smart city, is selected as an IoT environment, in which sensors are available in public areas. This smart campus is called Otaniemi3D [4], which provides information about energy consumption, occupancy, and user comfort by integrating Building Information Models (BIMs) and IoT devices through open standards called Open Messaging Interface (O-MI) and Open Data Format (O-DF). O-MI provides a communication framework between products

and distributed information systems which consume and publish information on a real-time basis. O-DF is defined as a simple ontology, specified as an extensible XML Schema, for representing the payload in IoT applications. [5]

II. Security Management in Otaniemi3D

All devices and software systems involved in IoT application scenarios typically exchange confidential product information with each other. Hence, they are potential subjects to attacks by unauthorized parties that attempt to gain access to such information or to manipulate the information exchange. Therefore, various layers and aspects of security based on our perspective are needed for the O-MI server [1]. From one perspective, security in Otaniemi3D or any other IoT systems can be divided into two layers: client-side and device-side (Fig. 1). The security problems on the client-side are mostly related to managing client user authentication (or registration), and then controlling their level of access to different information. In Otaniemi3D, this problem was resolved by means of a security module [6] using Facebook authentication with OAuth2 and an Access Control List (ACL) approach, where access rules are specified using an O-DF tree as the information structure. Fig. 1 shows the security architecture of Otaniemi3D in which the client-side is secured using authentication and access control module while the device-side might be affected by malicious activities.

In a campus-wide system such as Otaniemi3D, several sensors send their data to a server about real users and spaces. Sensors are installed inside the building without lock or any hardware security and the server code is also available online at Github, which makes it attack-prone. Thus, attackers could access the sensors, reprogram their firmware (e.g. sending sensitive data to a malicious server or sending false data to the server) or find the sensor identity and spoof it. In other words, the attacker is capable of connecting to the server with fake devices but a correct identity of the legitimate device. Since the current system runs no concrete device identification, the server will accept its identity claim. If the system runs the normal identification (e.g. based on the IP address), the problem still exists, since the attacker could simply eavesdrop the IP address of a special device or sensor in the network, even if other identities such as MAC are deployed. Thus, employing only one or two identifiers is not enough for identifying the device; instead, a combination of various identifiers is necessary, determined by device features.

III. Adversary model

Otaniemi3D authenticates sensor devices and the O-MI server by means of the TLS certificate using HTTPS encryption. As certificates are attached to devices, they can be stolen by attackers. This represents a serious source of threats to be considered. In particular, several malicious activities can be performed by an adversary with a stolen certificate. First, all current operational data stored on the certificate are read out, if they are not encrypted. If not protected appropriately, the attacker obtains the encryption and decryption mechanism implemented on the certificate and applies them to eavesdrop messages sent and received by this device. Ultimately, the attacker replicates the device by means of reverse engineering in order to exploit the attacked devices for malicious attacks.

[1]O-MI server: a server that implements the O-MI and O-DF standards and required functionality

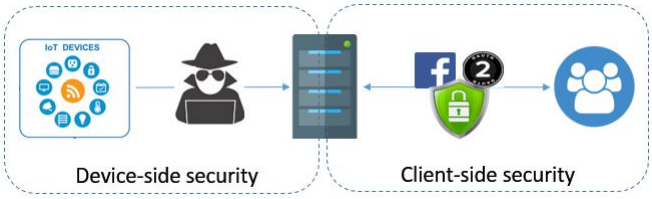

Fig. 1: Security architecture in Otaniemi3D

A. Attack model:

An attacker attaches an extra device with the same identity (or certificate) of an authorized device to the system and employs this identity to send messages to the IoT server on behalf of the legitimate device. This attack, also known as *object emulation attack* [7], is performed in order to send false data (or falsify data). The attacker physically accesses device certificates from a legitimate device and installs them on another sensor device, after which the malicious device begins sending false data to the server. In a critical situation, false data can trigger a false alarm, after which false alarms (e.g. fire alarm) can induce disasters. This is a type of physical security which has two feasible defense solutions [8]: placing a barrier around the network or security control at network layer. Since placing a barrier in large-space and open environments like smart campuses is impossible, we explore security control at the network layer.

IV. IoT Device Identification

Recently, researchers attempted to employ various types of features for device identifications in communication networks. Some feasible features (or dimensions) for this purpose could be as follows: location extracted from GPS and WiFi [9], [10], familiarity of devices derived from Bluetooth, time [10], identity (IP, MAC, or RFID tag) [10], unique hardware-specific characteristics like clock skew [11]–[13], device fingerprinting which uses MAC and properties of packets received from a device such as address and port of client/server [14], padding, packet size, destination IP counter [15], and inter-arrival time [16]. In order to avoid false alarms, environmental and real-time factors also have to be considered during attack detection [7]. Considering these features, it is necessary to have a continuous identity verification system. When the server receives the data, it will verify the sensor data and its behavior by comparing it with earlier values available in the feature database.

Fig. 2 shows a high-level overview of a framework that takes device identity decisions by performing automatic classification of the devices. The classification works based on device

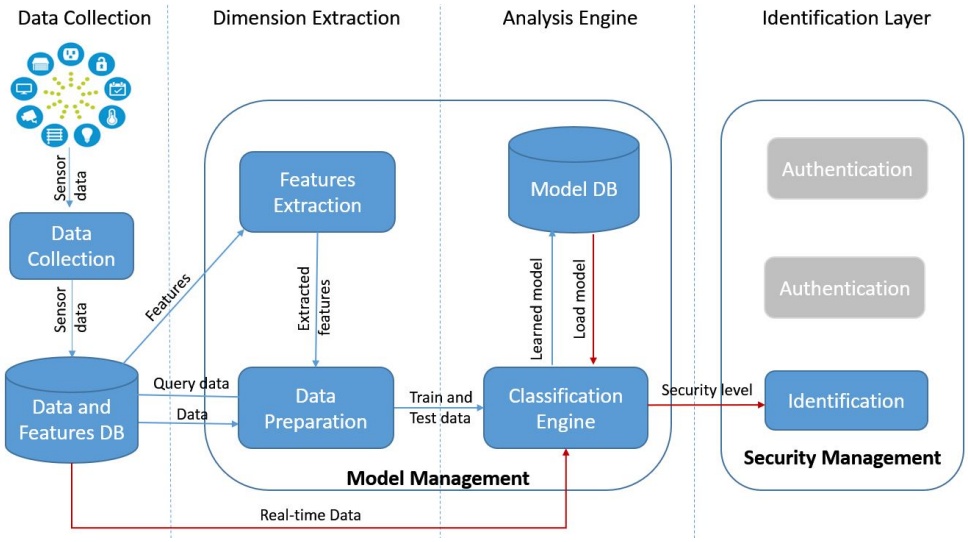

Fig. 2: Device Identification MeDI framework

features and a classifier model. This system has four layers in total: Data Collection, Dimension Extraction, Analysis Engine and Identification layer. Additionally, it has two main modules: *Model Management* and *Security Management*. *Model Management* is responsible for detecting the sufficient features and then analyzing them through machine learning methods. Once the dimensions have been extracted and the machine learning model of the current device is identified, *Security Management* makes the security decision (authenticated or not) and provides the enforcement support. In addition to the modules in the system, two databases (DBs) are also available to manage the data. The first one is the *Data and Features DB*, which includes the sensor data and dimension name. Once the required dimensions have been exposed from the first DB, the value of each dimension will be extracted from the observation and the learned model will be stored in *Model DB*.

The framework is driven by data which can be observed with the sensors of the devices through Bluetooth, WiFi, and GPS in the *Data Collection* step. The features are fed to the *Features Extraction* module to calculate features (or feature vector) describing the current observation and according to the extracted features value, the *Data Preparation* module sends a data query to *Data and Features DB* to find the required sensor data. If the device is undergoing its learning phase, then after extracting features and sensor data, they are employed to learn the model for the device by producing the training and test set. The model learned by the *Classification Engine*, is stored in *Model DB*. After learning phase comes the next phase of model processing, i.e. prediction phase (red lines in Fig. 2). Once the features have been extracted from new data, the *Classification Engine* adopts them plus the classifier model loaded from *Model DB* to classify new observations. Based on the classification result, a security level (e.g. binary value) will be assigned as output which is forwarded to the *Identification* layer. This security layer considers the security level while it is verifying the device identity. The *Identification* module verifies the identity, labels the device features of current observation as malicious or legitimate.

V. Use Case Description

Device Profiling. The main challenge is to authenticate the origin of the received messages on the server since there is no comprehensive restrictive authentication process or other effective method for detecting identity thefts. We apply device profiling (device behavior) for authentication, which has been adopted for user authentication. The researchers seek for behavioral features for continuous user authentication to overcome the weaknesses in earlier authentication solutions. For instance, the password-based solution suffered from password theft. Similarly, the certificate-based approach for device identification, suffered from certificate theft. Device profiling is extracted from messages received from the device. The message arriving to server has two parts [7]: *data part* measurement of the sensor and *fingerprint part* object unique fingerprint. In previous research, the fingerprint could be extracted from network traffic features such as inter-arrival time or from device-specific features like Radio Frequency (RF)-based signals. We believe that sensor measurement can also be a useful value for creating the fingerprint. For this purpose, in this paper, the device behavior is learned by combining both parts of messages, considering these feature sets: measurement-based and statistical features. According to these sets, three profiling methods are appointed for each device. Measurement-based features make the *measurement-only* method, statistical features construct *statistic-only* method, and the combination of these features establishes the *aggregation* method.

Feature Extraction. Since not enough data from the Otaniemi3D server could be extracted for our research purposes, we found a similar dataset to Otaniemi3D: Intel Lab Dataset [2] includes six columns (timestamp, epoch, humidity,

[2]http://db.csail.mit.edu/labdata/labdata.html

temperature, light, and voltage) collected from 54 sensors deployed in the Intel Berkeley Research lab. In addition to four sensor measurements (*humidity*, *temperature*, *light*, and *voltage*), we created one more feature which was also used by previous papers for fingerprinting. The *inter-arrival time* is computed for each packet (or data record) according to the time interval between two consecutive packets (current and previous packets). The statistical features must be calculated over a flow or sequence of packets from the source. In this paper, 12 consecutive packets are considered the *packet flow*. Based on the analysis, 12 packets is a good trade-off for flow length; the same number was used by [15] for making fingerprints. Based upon timestamp, epoch (monotonically increasing sequence number from each device), and inter-arrival time, five statistical attributes are calculated: *flow duration*, *inter-arrival average*, *number of expected packets*, *number of missing packets*, and *idleTime*. In the same vein, the sensor measurements are replaced with their average values in each flow of 12 packets (see TABLE I).

Device Model. We construct a binary classifier model for each device using fixed-length fingerprints. Each classifier classifies the data captured from its corresponding device, i.e. $Device_i$, as legitimate class and data from other devices as malicious class. Finally, a single classifier which is the ensemble of all the classifiers, is used to recognize an unknown fingerprint as belonging to legitimate or malicious class. When a new device emerges, after collecting enough data from the device, a new classifier is trained and added to the pool of classifiers without making any modification to the existing classifiers. Our analysis showed that the data are cluttered, i.e. data from legitimate class are not linearly separable from other class. Thus, nonlinear classifiers such as tree-based algorithms are reasonable methods. Random forest is amongst the top performer tree-based algorithm in terms of prediction accuracy [17].

TABLE I: Weights of values and statistics attributes

Attribute type	Attribute name	Weight
Sensor measurements	voltage average	2.6431
	light average	2.5460
	humidity average	1.6467
	temperature average	1.3752
Device statistics	flow duration	0.9120
	inter-arrival average	0.8976
	number of expected packets	0.5714
	number of missed packets	0.5668
	idle time	0.3174

VI. Evaluation

Based upon two categories of features (statistical and measurement-based), three profiling methods are established: *statistic-only*, *measurement-only*, and *aggregation*. The *statistic-only* method includes only the device statistics features (last five features in TABLE I). *Measurement-only* contains only the sensor measurements (first four features in TABLE I). The *aggregation* method combines all these nine features to one model.

A. Classification Results

The average value of performance metrics (accuracy and F-measure) is calculated for 53 classifications according to 53 sensor devices. One device in Intel lab data is ignored in analysis since it has empty measurement values. Experimental results (see TABLE II) show that the average accuracy for the *statistic-only*, the *measurement-only* and the *aggregation* model equal to 65.21%, 71.18% and 76.15% respectively. The *aggregation* model obtains a 11% accuracy improvement over the *statistic-only* model. However, the results achieved by the sensor *measurement-only* and the *aggregation* models are close. The *aggregation* model provides a 4% improvement to the *measurement-only* method. Following the same trend, as the results display, f-measure, precision, and specificity are improved by aggregating the measurement-based attributes and statistic-based attributes. Thus, the *aggregation* model achieves the best average result in all metrics.

TABLE II: Classification performance with three methods

	Accuracy	F-measure	Precision	Specificity
Statistic-only	65.21%	0.583	0.556	0.652
Measurement-only	71.18%	0.481	0.728	0.921
Aggregation	**76.15%**	**0.625**	**0.781**	**0.901**

By Random Forest algorithm, the importance for each predictor variable (or feature) is also computed based on prediction error, i.e. out-of bag error [18]. The larger this value, the more important the variable is for predicting the device identity. TABLE I lists the attributes in each method, sorted by their importance rate. The important weights show that sensor variables are more important that statistical variables.

B. Statistical Analysis

To verify that there is enough evidence to support our findings about the results, we run a statistical test. In [19], Friedman test is employed to compare various machine learning algorithms over multiple datasets. We follow a similar procedure of statistical analysis by considering data of each device a separate dataset and by comparing the three profiling methods concerning these devices. The Friedman test ranks the algorithms for each dataset separately so that the best performing model ranks fist, the second ranks second and so on [19]. Table III lists the accuracies and their ranks for 53 devices concerning three methods.

The Friedman test compares the average ranks of algorithms $R_j = \frac{1}{N}\sum_i r_i^j$ where r_i is the rank of the j-th of models on the i-th dataset. The null hypothesis is that all three methods are equivalent. The Friedman statistic is computed as follows:

$$X_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \quad (1)$$

$$X_F^2 = \frac{12*53}{3*4}\left[(1.255^2+2.160^2+2.585^2) - \frac{3*4^2}{4}\right] = 48.934$$

TABLE III: Accuracy and ranking for each device

	Aggregation	**Measurement-only**	**Statistic-only**
$Device_1$	**76.90%** (1)	71.74% (2)	55.16% (3)
$Device_2$	**68.66%** (1)	66.10% (2)	56.70% (3)
$Device_3$	**68.28%** (1)	62.10% (2)	54.03% (3)
$Device_4$	**73.65%** (1.5)	73.65% (1.5)	54.96% (3)
$Device_5$	79.62% (2)	**81.50%** (1)	61.76% (3)
$Device_6$	58.78% (2)	53.41% (3)	**69.51**% (1)
$Device_7$	**64.60%** (1)	62.24% (2)	61.36% (3)
$Device_8$	**74.72**% (1)	72.16% (2)	54.83% (3)
$Device_9$	62.18% (2)	**64.43%** (1)	56.86% (3)
$Device_{10}$	**71.92**% (1)	7184% (2)	57.59% (3)
⋮	⋮	⋮	⋮
$Device_{53}$	**75.73**% (1)	71.52% (2)	70.23% (3)
Average rank (R_j)	**1.255**	2.160	2.585

Variables k and N are the total number of methods and the total number of datasets, respectively. With three methods ($k = 3$) and 53 datasets ($N = 53$), the computed Friedman statistic for our experiment is 48.934. Friedman statistic, being conservative, was substituted by better statistic which is distributed according to the F-distribution with $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2} \tag{2}$$

$$F_F = \frac{(53-1) * 48.934}{53 * (3-1) - 48.934} = 44.59$$

Based on the number of methods and data, F_F is distributed according to the F distribution with $3-1 = 2$ and $(3-1)(53-1) = 104$ degrees of freedom. The critical value of $F(3, 104)$ for $\alpha = 0.05$ is 2.691 which is less than the F_F. Therefore, the null hypothesis is rejected, which means that the three methods are statistically different in accuracy.

Whenever a significant difference between three or more sample means has been revealed by analyses like Friedman, a post-hoc test can be used to recognize sample means that are significantly different from each other. Nemenyi test [20], a post-hoc test, considers the performance of two methods significantly different if the corresponding average ranks differ by at least the Critical Difference (CD) with q_α as the critical value. Based on two-tailed Nemenyi test, the critical value $q_0.05$ for 3 models is 2.343, so the CD can be:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} = 2.343 \sqrt{\frac{3*4}{6*53}} = 0.455 \tag{3}$$

Comparison of this CD with each average rank proves that the *aggregation* method performs significantly better than *statistic-only* ($|1.255 - 2.585| \geq 0.455$) and also better than the *measurement-only* method ($|1.255 - 2.160| \geq 0.455$). Therefore, considering the two test results, we can claim that exploiting the sensor measurements can amend the process of device identification.

VII. Related Work

As multiple users and devices need to authenticate each other through trustable services, it is essential to manage identity authentication in the IoT [21]. The idea of device identification to handle the privacy of data was first introduced in 2009 by Sarma and Girao [22] while most of Service Oriented Architecture (SOA)-based identity management previously proposed in IoT like Liberty Alliance, Card-Space, and Shibboleth did not consider device identity in the framework [23]. Recently, identities are adopted as representations of entities of all kinds as the end points of communication. An IoT device can identify itself using its identity or its specific features [24]. Cloud-based IoT solutions like that introduced in [25] exploit the former method presenting a framework for identity management with basic functions such as Registration of sensors and receiver device to the cloud, Identification of hosted services, and Authentication of sensors and receiver device.

with the increasing number of security attacks in the world, device basic identity is not enough for device authentication since it can be forged easily. Therefore, specific features require to be extracted from the device. The most common method employs device traffic data for device-type identification. Network traffic classification can be based on different major attributes: Payload-based attributes according to signatures of the traffic at the application layer level; Statistical-based attributes related to traffic statistical characteristics [26]. Statistical-based attributes could be mostly extracted from the device header. Depending on the IoT devices and their environment, these features can be calculated from specific features of device. For example, for mobile devices, signal noise from microphone [27] has been nominated as a good identifier. In addition, some external database like Alexa Rank and Geo IP could be accompanied to the traffic data from different network layers [26].

The proposed method in this paper can be considered a traffic analysis method with certain differences. The payload information is ignored in traffic analysis since most of the payloads are encrypted and fingerprints can not be extracted from encrypted traffic [15] while in our environment, due to a new messaging standard (O-MI), the server authenticates the device before storing the decrypted data in database; it has access to the unencrypted payload data. In addition to exploiting payload data, the proposed method operates during the entire life-cycle of devices although some of the previous traffic analyzer denote for specific stage of life. For example, IoT Sentinel [15] verifies the identification for new devices only during their registration process but several attacks could occur during device life-cycle. Furthermore, some papers [7] consider different features for various object types, while in our method, features assumed to be the same for all IoT devices.

VIII. Conclusion and Future Work

In this paper, we proposed an IoT device identification framework. We also presented three device identification meth-

ods, according to their profiles. Two methods are based on a single data category (the sensor *measurement-only* and the *statistic-only*), while the third one (the *aggregate* model) is constructed by the merger of all features provided by the first two methods. We evaluated the performance of these models by adopting a lab dataset. Our results show a significant accuracy improvement of the *aggregate* model in comparison to the *statistic-only* model. However, the gain between the *aggregate* model and the sensor *measurement-only* model is slight. Overall, measurement features are more informatics than packet statistics.

This work can be extended by analyzing the performance of header-based models of identification and comparing them with the proposed *measurement-based* model in the same IoT platform. The performance of the proposed method can also be analyzed by running an intrusion environment. In addition to a physical attack, there are several attack scenarios including remote hacking, which requires further investigation. This can be inspected as the next step of this research.

IX. Acknowledgment

The research leading to this publication is supported by the European Union's Horizon 2020 research and innovation program (grant 688203) and Academy of Finland (Open Messaging Interface; grant 296096).

References

[1] W. Wang, Y. Yuan, and N. P. Archer, "A contextual framework for combating identity theft," *IEEE Security & Privacy*, vol. 4, no. 2, pp. 30–38, 2006. [Online]. Available: https://doi.org/10.1109/MSP.2006.31

[2] G. R. Milne, A. J. Rohm, and S. Bahl, "Consumers' protection of online privacy and identity," *Journal of Consumer Affairs*, vol. 38, no. 2, pp. 217–232, 2004.

[3] E. Bertino and N. Islam, "Botnets and internet of things security," *IEEE Computer*, vol. 50, no. 2, pp. 76–79, 2017. [Online]. Available: https://doi.org/10.1109/MC.2017.62

[4] A. Buda, T. Kinnunen, B. Dave, and K. Främling, "Developing a campus wide building information system based on open standards," in *Joint Conference on computing in Construction, JC3, Heraklio, Greece, July 4-7, 2017*, 2017, pp. 733–740.

[5] T. O. Group, *Open Data Format (O-DF), an Open Group Internet of Things (IoT) Standard*, 2014, http://www.opengroup.org/iot/odf/.

[6] N. Yousefnezhad, R. Filippo, A. Javed, B. Andrea, M. Madhikermi, and K. Främling, "Authentication and access control for open messaging interface standard," in *Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS), 2017 14th International Conference on*. ACM, 2017.

[7] Y. Sharaf-Dabbagh and W. Saad, "On the authentication of devices in the internet of things," in *17th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM 2016, Coimbra, Portugal, June 21-24, 2016*, 2016, pp. 1–3. [Online]. Available: https://doi.org/10.1109/WoWMoM.2016.7523532

[8] L. SolarWinds Worldwide, *Detecting and Preventing Rogue Devices*, 2017. [Online]. Available: http://web.swcdn.net/creative/pdf/Whitepapers/UDT_WP_Detect_Prevent_Rogue_Devices.pdf

[9] M. Miettinen, S. Heuser, W. Kronz, A. Sadeghi, and N. Asokan, "Conxsense: automated context classification for context-aware access control," in *9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '14, Kyoto, Japan - June 03 - 06, 2014*, 2014, pp. 293–304. [Online]. Available: http://doi.acm.org/10.1145/2590296.2590337

[10] C. Perera, A. B. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 414–454, 2014. [Online]. Available: https://doi.org/10.1109/SURV.2013.042313.00197

[11] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," in *2005 IEEE Symposium on Security and Privacy (S&P 2005), 8-11 May 2005, Oakland, CA, USA*, 2005, pp. 211–225. [Online]. Available: https://doi.org/10.1109/SP.2005.18

[12] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, "On the reliability of wireless fingerprinting using clock skews," in *Proceedings of the Third ACM Conference on Wireless Network Security, WISEC 2010, Hoboken, New Jersey, USA, March 22-24, 2010*, 2010, pp. 169–174. [Online]. Available: http://doi.acm.org/10.1145/1741866.1741894

[13] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE Trans. Mob. Comput.*, vol. 9, no. 3, pp. 449–462, 2010. [Online]. Available: https://doi.org/10.1109/TMC.2009.145

[14] R. R. R. Barbosa, R. Sadre, and A. Pras, "Flow whitelisting in SCADA networks," *IJCIP*, vol. 6, no. 3-4, pp. 150–158, 2013. [Online]. Available: https://doi.org/10.1016/j.ijcip.2013.08.003

[15] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma, "Iot SENTINEL: automated device-type identification for security enforcement in iot," in *37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017*, 2017, pp. 2177–2184. [Online]. Available: https://doi.org/10.1109/ICDCS.2017.283

[16] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "Gtid: A technique for physical device and device type fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 519–532, 2015.

[17] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: A classification and regression tool for compound classification and QSAR modeling," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1947–1958, 2003. [Online]. Available: https://doi.org/10.1021/ci034160g

[18] C. Fuchs, *Predicting outcomes of bariatric surgery using datamining techniques*, Eindhoven University of Technology, Department of Industrial Engineering and Innovation Sciences, Information Systems Research Group, 2017.

[19] J. S. Sartakhti, H. Afrabandpey, and M. Saraee, "Simulated annealing least squares twin support vector machine (SA-LSTSVM) for pattern classification," *Soft Comput.*, vol. 21, no. 15, pp. 4361–4373, 2017. [Online]. Available: https://doi.org/10.1007/s00500-016-2067-4

[20] P. Nemenyi, "Distribution-free multiple comparisons," in *Biometrics*, vol. 18, no. 2. INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1962, p. 263.

[21] M. Abomhara and G. M. Køien, "Security and privacy in the internet of things: Current status and open issues," in *2014 International Conference on Privacy and Security in Mobile Systems, PRISMS 2014, Aalborg, Denmark, May 11-14, 2014*, 2014, pp. 1–8. [Online]. Available: https://doi.org/10.1109/PRISMS.2014.6970594

[22] A. C. Sarma and J. Girão, "Identities in the future internet of things," *Wireless personal communications*, vol. 49, no. 3, pp. 353–363, 2009.

[23] P. Mahalle, S. Babar, N. R. Prasad, and R. Prasad, "Identity management framework towards internet of things (iot): Roadmap and key challenges," in *Recent Trends in Network Security and Applications - Third International Conference, CNSA 2010, Chennai, India, July 23-25, 2010. Proceedings*, 2010, pp. 430–439. [Online]. Available: https://doi.org/10.1007/978-3-642-14478-3_43

[24] R. Roman, P. Najera, and J. López, "Securing the internet of things," *IEEE Computer*, vol. 44, no. 9, pp. 51–58, 2011. [Online]. Available: https://doi.org/10.1109/MC.2011.291

[25] S. Horrow and A. Sardana, "Identity management framework for cloud based internet of things," in *First International Conference on Security of Internet of Things, SECURIT '12, Kollam, India - August 17 - 19, 2012*, 2012, pp. 200–203. [Online]. Available: http://doi.acm.org/10.1145/2490428.2490456

[26] D. Bekerman, B. Shapira, L. Rokach, and A. Bar, "Unknown malware detection using network traffic classification," in *2015 IEEE Conference on Communications and Network Security, CNS 2015, Florence, Italy, September 28-30, 2015*, 2015, pp. 134–142. [Online]. Available: https://doi.org/10.1109/CNS.2015.7346821

[27] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, "Mobile device identification via sensor fingerprinting," *CoRR*, vol. abs/1408.1416, 2014. [Online]. Available: http://arxiv.org/abs/1408.1416