Caviglione, L., Gaggero, M., Cambiaso, E., & Aiello, M. (2017). Measuring the energy consumption of cyber security. IEEE Communications Magazine, 55(7), 58-63. DOI: 10.1109/MCOM.2017.1600955
https://ieeexplore.ieee.org/abstract/document/7981525

Measuring the Energy Consumption of Cyber Security

Luca Caviglione*, Mauro Gaggero*, Enrico Cambiaso^, Maurizio Aiello^

** Institute of Intelligent Systems for Automation, National Research Council of Italy, I-16149 Genoa, Italy*

^ Institute of Electronics, Computer and Telecommunication Engineering, National Research Council of Italy, I-16149 Genoa, Italy

Abstract

The Internet is a core tool for developing commercial and social relationships. As a consequence, cyber security must be properly assessed, for instance to face with new and sophisticated threats. To deliver large-scale services, proper countermeasures characterized by a non-negligible energetic impact have to be pursued. In this perspective, this paper proposes to investigate the energy required by the most popular cryptographic algorithms. The collected measures are used to model relationships between power drains and size of the key or offered load via a black-box approach. Results can also be used to prevent classical traffic analysis campaigns.

1. Introduction

Nowadays, the Internet connects a huge amount of entities coordinating and exchanging personal and sensitive data. For instance, the Internet of Things (IoT) approach is used to perform field measurements, cloud platforms offer computing resources as a commodity, and personal mobile devices enable connectivity while on the road. As a consequence, cyber security is definitely a core requirement for mobile, pervasive, and complex network architectures [1]. Unfortunately, delivering such a rich set of services, especially in a trusted and secure manner, does not come for free. In fact, Internet Service Providers (ISPs) or entities operating a datacenter usually face high expenditures, especially in terms of energy bills. Therefore, understanding and optimizing the energy consumption of computing and network appliances have become relevant research topics [2] [3], [4]. However, aspects related to energy consumption of cyber security mechanisms have been often neglected, even if the emerging trend is to explicitly consider their impact as well [5]. In this perspective, a relevant portion of the literature focuses on mobile devices (see, e.g., [6]), mainly due to their battery-operated flavor, limited amount of computing/storage resources, and intrinsic difficulties to measure power drains in a non-invasive manner [7]. This paper tries to fill such gap and proposes to characterize the energy consumption of different standard cryptographic algorithms deployed in modern ISPs, datacenters, and end nodes. In fact, providing security is often an integrated process involving entities placed both in the core and at the border of the network [1]. In more details, enlightening relations among energy requirements and cyber security allows to: (*i*) estimate the energy requirements of security mechanisms to assess their economic impact and perform optimizations; (*ii*) demonstrate how traffic related to security aspects could be measured through an higher-level indicator, such as energy drain, in order to ensure scalability by preventing the need of capturing packets and processing big data-like information; (*iii*) support the idea of using energy consumption as a marker to develop novel cyber security mechanisms, e.g., to early detect attacks.

To this aim, we performed a measurement campaign on several production-quality cryptographic algorithms. Specifically, particular attention was put to select implementations that can be deployed both within end nodes and machineries used in an ISP or in a datacenter. To model data collected in different use-cases, we introduced a black-box approach relying on statistical tools. In particular, our goal is to find a qualitative relationship among cyber security aspects and power requirements, for instance to offer guidelines to engineer and optimize large-scale deployments, or to design novel configurations. To this aim, we used a least squares approach to obtain a polynomial model of the energy costs of cyber security. Results indicate that there is room for developing a more green and secure Internet.

The rest of this paper is structured as follows. Section 2 presents the reference scenario and the considered security technologies, while Section 3 briefly discusses the theoretical background used for modeling the energy consumption. Section 4 deals with the adopted testbed and Section 5 showcases numerical results. Lastly, Section 6 concludes the work by reviewing the most important lessons learned.

2. Reference Scenario and Considered Security Technologies

As previously pointed out, networks are a relevant part of our lives. As a possible example, smartphones are used by about the 65% of the global population to perform financial activities, share data over online social networks, and communicate in real-time. This leads to infrastructures characterized by a high degree of heterogeneity, for instance wireless loops have different security requirements with respect to wired trunks. As a consequence, networks should provide a proper degree of cyber security, as the volume and type of data are of interest for cyber criminals, e.g., to profile users or collect information for large-scale social engineering attacks. Nevertheless, providing a secure environment requires acting on different entities ranging from user devices to remote machineries.

In modern scenarios, cyber security is provided through a vast set of techniques, which can also interact in a very complex manner. In fact, guaranteeing trusted and secure features encompasses a rich variety of protocols (e.g., to deliver the information via Transport Layer Security) and machineries (e.g., to distribute and manage credentials or certificates). For instance, a server devoted to implement Authentication, Authorization and Accounting can be complex, especially if scalability is needed. Unfortunately, the precise understanding of how the different hardware and software components contribute to energy drains is still an open research problem, especially due to heterogeneity of implementations [3], [5], [7]. Assessments on the energy used by tools like antivirus, spam filters, and firewalls have been already partially addressed (see, e.g., [8] and references therein). However, a clear understanding of what and how deplete energy is still missing. Therefore, we decided to solely focus on basic security services, and in particular to evaluate the energy requirements of cryptographic algorithms. In fact, functionalities of tools used to enforce cyber security can be decomposed into simple mechanisms to guarantee communication integrity, non-repudiation of a message, authentication, as well as consistence of a generic fragment of information. Specifically, we considered the following classes of algorithms.

- *Encryption algorithms*: they take plaintext and a key as inputs and provide ciphered text as output. This operation can be performed by many different methods, e.g., by means of text blocks expansion and reduction, data permutation, or substitution-boxes [8]. The algorithms considered in this paper are AES-CBC, AES-ECB, Blowfish, DES-ECB, 3DES, and RC4;

- *Hashing algorithms*: they are primarily used to check the data integrity by computing a fixed-length string against a text provided as input. Hash functions are usually engineered to provide a unique output difficult to invert [9]. The considered algorithms are MD2, MD4, MD5, RIPEMD-160, SHA-1, SHA-2, Tiger, and Whirlpool;

- *Keyed-Hash Message Authentication Code* (HMAC): primarily used to avoid message and hash tampering, they offer a mechanism for message authentication by using cryptographic hash functions in combination with a secret shared key [10]. The considered algorithms are the same tested for the hashing case.

Table I briefly describes the different cyber security algorithms taken into account. We point out that some of them are outdated (e.g., the SHA-1 has been considered insecure from 2010 and its support was dropped in 2016 from Google Chrome) or "flawed", i.e., known vulnerabilities have been disclosed (e.g., DES-ECB and 3DES) [11]. However, since such methods are still widely adopted, especially in legacy devices, they have been considered for the sake of completeness.

Table I: Considered cyber security algorithms.

Algorithm	Acronym	Brief Description
AES-CBC	Advanced Encryption Standard - Cipher Block Chaining	The encryption is based on a substitution-permutation network. In this case, each block of plaintext is XORed with the previous cyphered block before being encrypted.
AES-ECB	Advanced Encryption Standard – Electronic Code Book	Simpler variation of the AES. In this case, the original message is divided into blocks, and each one is encrypted separately.
Blowfish	-	A Feistel network-based block cipher.
DES-ECB	Data Encryption Standard – Electronic Code Book	It takes a fixed-length string of plaintext and transforms it through a Feistel network.
3DES	Triple DES	3DES applies three times the DES to increase robustness.
RC4	Rivest Cipher 4	It generates a pseudorandom stream of bits via permutation and pointers.
MD2, MD4 and MD5	Message Digest (2, 4 and 5)	Hashing algorithms using different functions (4 in the case of MD5).
RIPEMD-160	RACE[1] Integrity Primitives Evaluation Message Digest	It is similar to the MD, but it is considered more secure.
SHA-1 and SHA-2	Secure Hash Algorithm (1 and 2)	A family of hashing functions using different architectures during the years to increase robustness.
Tiger	-	A collision resistant hashing function based on the Merkle–Damgård principle.
Whirlpool	-	A more secure modification of the AES.

[1] RIPEMD was developed within the European Union Project RACE Integrity Primitives Evaluation (RIPE) 1988 - 1992 supported by the EU RACE Program - Research and Development in Advanced Communications Technologies.

3. Modeling

The crucial issue of constructing a qualitative model of the energy required by cyber security mechanisms is the poor granularity of the available data. This is a direct consequence of the adopted standard solutions. For instance, the length of the key used for encrypting information in a real scenario does not vary in a “continuous” way since only well-defined, discrete values are considered. Accordingly, we decided to use a data set containing only the energy consumption of “feasible” configurations adopted in production-quality environments.

A direct consequence of the “quantization” of the available configurations is the need of creating a model of energy consumption that is simple but at the same time robust to noises characterizing the collected data. Toward this end, the relationship between the power consumption and the different cyber security methods was modeled through polynomials. In particular, we used a least squares technique to tune the coefficients by minimizing the mean squared error between the available measurements and the output of the models (i.e., the so-called residuals). In order to limit the impact of noises, the degree of the polynomials had to be chosen much smaller than the number of available measures. This avoids the overfitting phenomenon, i.e., the obtained models interpolate a random error (the noise) instead of approximating the real underlying relationship between the considered quantities.

More specifically, our goal is to approximate the relationships either between the size of the key for data encryption and the energy consumption or between the amount of processed data and the power drain. Using least squares, the resulting models are very robust to noises, and the unknown parameters can be obtained by using simple algebraic equations, i.e., there is no need of applying complex optimization procedures.

4. Measurement Methodology

To build the dataset used to model the energy consumption of cryptographic algorithms running both in end nodes and network devices, we conducted an extensive set of trials. In more detail, tests were performed to capture a mixed set of use cases, especially to understand the impact of the “strength” of security algorithms on the energy footprint. Thus, we performed experiments by varying the following parameters:

- *Algorithm*: for each algorithm, we evaluated its energetic impact to understand whether the complexity or the implementation play a role;

- *Size of the key*: we tested how varying the size of the key influences the required energy. As said, the sizes of the key have been selected to reflect “production-quality” requirements;

- *Load* (*or Volume*): all the permutations of the aforementioned configurations were stressed with different traffic. In this way, we tried to assess the energetic scalability of the algorithms, also to enlighten some critical aspects of the implementation. The offered load was considered of increasing sizes, i.e., 1 KB, 10 KB, 100 KB, 1 MB, 10 MB, 100 MB, and 1 GB, to account for usages ranging from a user device sending a small amount of data to a core/border appliance processing traffic groomed from a high-speed link.

To effectively measure the energetic footprint of cyber security, it is mandatory to remove possible overheads introduced by the hosting machinery. For instance, many algorithms require the access to L2 interfaces or to interact with the buffering architectures of devices. This is a critical issue, as precisely identifying and quantifying the energetic requirements of network components, protocols, and specific hardware subsystems are still mostly open problems [3]. Therefore, after preliminary evaluations of different configurations, i.e., emulated devices, virtual machine-based nodes, and real components, we decided to use a controlled framework built from scratch. Such testbed was created by using Linux (Ubuntu 14.04.2 LTS, GNU/Linux 3.16.0-20-generic x86/64 kernel) running on an Intel Dual Core E2160 CPU at 1.80 GHz with 8 GB of RAM. To collect and process data, we implemented ad-hoc Java and Python modules together with bash scripts. As regards the cyber security algorithms, we used Java/Linux implementations, which is the choice commonly used in production-quality settings and Android-based mobile devices. In particular we adopted the Bouncy Castle Cryptographic API Libraries[2] (release 1.55 of August 2016).

As commonly done in the literature, we estimated the required energy by exploiting the tight relation between the CPU usage and the consumed power [3], [6]. In fact, many works shows that the power used for the computation is the predominant part of the energy consumed within a device (see, e.g., [8]). Therefore, along the lines of [12], we measured the used computing resources without considering overheads due to test conditions or other competing processes. In other words, we assumed that the consumed energy is proportional to the amount of CPU used for the entire processing, where the proportionality coefficient depends on the specific hardware/software technology. The amount of used CPU was measured for each configuration (i.e., type of algorithm, length of the key, and offered load), and samples were stored into a database for further processing. To model data, we used Matlab on a PC equipped with an Intel Core2 Duo CPU at 1.8 GHz and 2 GB of RAM.

5. Numerical Results

In this section, we present the results obtained through an extensive measurement campaign. Specifically, Table II showcases the considered algorithms jointly with all the different keys used for our investigation. Each configuration was tested with different loads, ranging from 1 KB to 1 GB. As regards the polynomials used for modeling consumption, we fixed the degree to 2 to limit the impact of coarse-grained measurements. We point out that this limit is due to the fact that the length of the keys cannot vary "continuously", rather it must adhere to standard values (see the discussion in Section 3). As said in Section 4, trials focused on modeling a "qualitative" behavior. In fact, the precise understanding of how the technologies used for networking or computing contributes to energy drains has been an important topic for at least a decade [3], [6], and it is still part of ongoing research (see, e.g., [13] for the case of IoT-based scenarios). As an example, consumptions are highly influenced by the hosting hardware (e.g., commodity hardware *vs* ad-hoc FPGA-based implementations). Therefore, identifying the proper value for the proportionality coefficient between energy consumption and amount of used CPU is challenging and outside the scope of this work. Thus, we just focused on the CPU used by a given algorithm to complete a task, which is a more general abstraction of the energy consumed by an appliance to run the functionalities implementing the security layer. The reported results are "relative" values, i.e., they are not absolute quantities, but scaled against the maximum measured CPU usages (taken equal to 1). With a little abuse of terminology, in the following we will refer to "CPU usage", "energy", and "power" consumption interchangeably, as they are proportional.

[2] https://www.bouncycastle.org/java.html (Last Accessed: November 2016)

Table II: Algorithms and key sizes considered for our tests.

Algorithm	Key Size (bits)								
	64	**128**	**160**	**192**	**198**	**256**	**384**	**512**	**1024**
AES-CBC		•		•		•			
AES-ECB		•		•		•			
Blowfish	•	•		•		•		•	•
DES-ECB	•								
3DES		•		•					
RC4	•	•		•		•		•	•
MD2	•			•	•	•			
MD4	•			•	•	•			
MD5	•			•	•	•			
RIPEMD-160		•	•	•	•	•	•		
SHA-1			•	•	•	•	•	•	
SHA-2			•	•	•	•	•	•	
Tiger				•		•			
Whirlpool					•	•	•		

Figure 1 depicts the CPU used by different encryption algorithms to process various traffic volumes using a 128 bit key. Similar results were obtained for other lengths of the keys, but they are not reported for the sake of brevity. The main finding is that all the considered algorithms exhibit two different consumption profiles: (*i*) an almost constant consumption trend for loads smaller than 10^7 bytes and (*ii*) a linear increasing one for higher volumes of traffic (notice the logarithmic scale in the *x*-axis). The major exceptions are the RC4 and AES-ECB methods, which appear to be almost insensible to the amount of data to be processed. This also suggests the presence of major optimizations within the software implementation. In general, the ISP, datacenter engineers, or software developers should prefer more robust solutions having the same energy footprint. For instance, it turns out that using load-insensitive mechanisms is preferable if some form of load distribution is not possible (e.g., having distributed architectures processing in parallel smaller fractions of the overall traffic). As another example, results indicate that, when in the presence of a mobile population with limited power sources, it would be possible to trade energy for security, for instance by using simpler or more efficient cyber security solutions at the price of a reduced energy consumption.

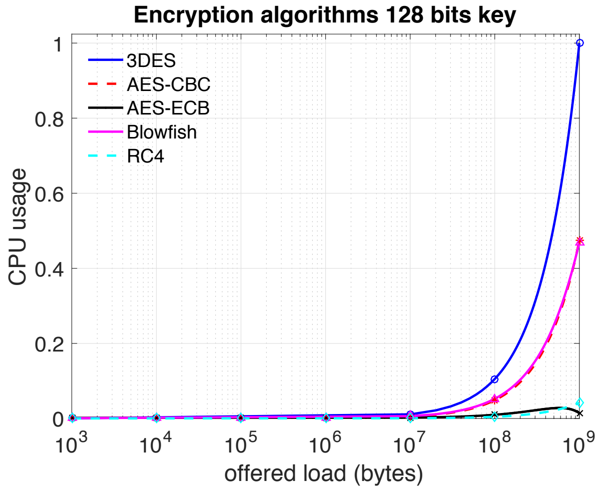

Figure 1: Relationship between CPU usage and offered load for different encryption algorithms with a 128 bit key.

Figure 2 shows the results of encryption algorithms when varying the length of the key. For the sake of compactness, we report only the results related to the Blowfish and RC4. Such methods appear to be insensitive to the used key, thus making preferable to adopt more robust solutions since they do not account for additional energy requirements or battery drains. Similar considerations could be done also for the remaining encryption algorithms, which have trends similar to the Blowfish one. The presence of some energy-insensitive techniques suggests that reducing the length of the key to pursue economic and energy savings could be useless.

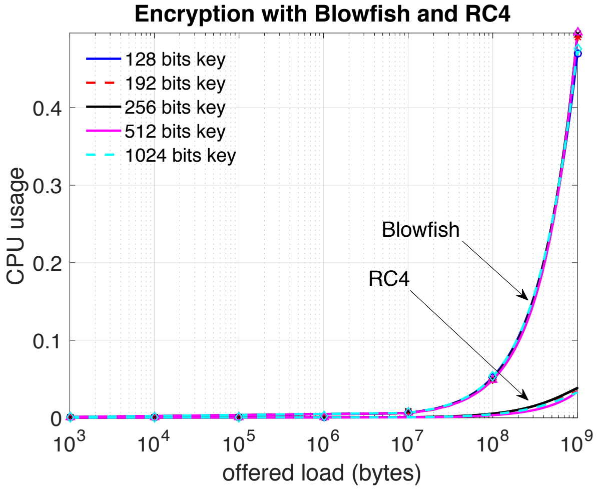

Figure 2: Relationship between CPU usage and offered load for Blowfish and RC4 by varying the length of the key.

Figure 3 displays the results obtained by investigating different hashing algorithms. Also in this case, two different zones characterize consumptions, i.e., the required CPU is almost constant for loads less than 10^7 bytes, while it increases linearly for larger loads. The Whirlpool algorithm reveals to be quite power-hungry, hence it is not suitable for mobile devices or to pursue energy efficiency. Instead, the remaining algorithms have similar consumptions, and therefore the ISP/datacenter operator could select the most suitable techniques without paying too much attention to the energy.

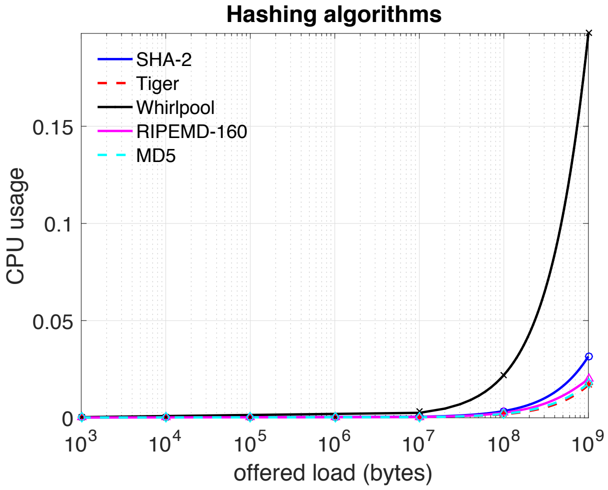

Figure 3: Relationship between CPU usage and offered load for hashing algorithms.

Figure 4 presents the results for the case of HMAC algorithms with key length of 256 bit. The other key lengths have comparable results but have been omitted for the sake of brevity. The reported trends are similar to the ones of Figure 3. However, this is not surprising since HMAC offers message authentication by means of cryptographic hash functions. Therefore, the considerations regarding the energetic requirements are the same of the case of the hashing algorithms. It is worth noting that, for the MD2 case, the low degree of sophistication does not match with its high energetic requirements. By performing additional investigations, we found that this is due to a poor software implementation of the algorithm. As a consequence, this showcases that code optimization can make a relevant difference in terms of economic expenditure for the energy bill and the quality of experience of end users, e.g., by avoiding reducing the lifetime of mobile devices due to excessive power depletions.

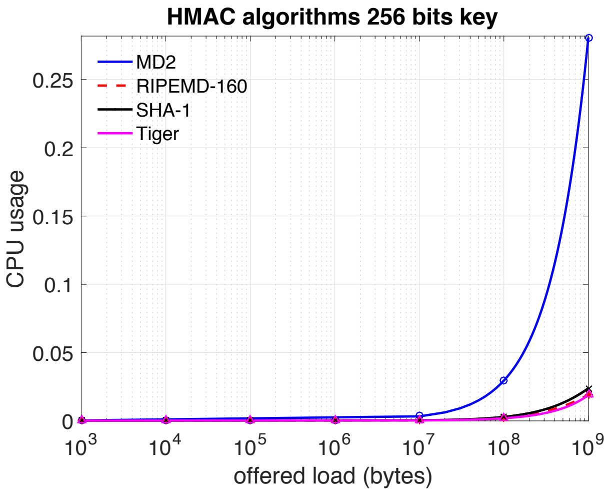

Figure 4: Relationship between CPU usage and offered load for HMAC algorithms with a 256 bit key.

6. Lessons Learned and Conclusions

As discussed, understanding the energetic requirements of cyber security techniques is fundamental for the development of green and secure network environments. The main lessons learned and possible future research directions are the following.

Refrain from pursuing economic and energy savings at the price of cyber security. Our results indicate that algorithms like MD4, SHA-1, and SHA-2 have small energy footprints. However, as they are considered highly insecure, their adoptions should be avoided even if energetically favorable.

Optimize the code. The comparison of different cryptographic algorithms hints that software optimization could play a major role in terms of economic savings. For instance, the excessive consumption of MD2 reported in Figure 4 is not fully justified by its computational requirements. In this vein, the next-generation of mobile, trusted, and secure networks should not only be secure by-design, but also energy-efficient. Besides, code optimization could be also an early and effective countermeasure to prevent energy-draining attacks [14].

Offload and fragment if needed. The obtained models of power consumption show two different behaviors characterizing cyber security techniques, i.e., constant *vs* linear for low *vs* high loads. Therefore, load fragmentation may be favorable owing to simpler and more efficient entities working in parallel. This could be also a benefit for nodes with limited capabilities, for instance by offloading some security operations via a cloud-based paradigm. For the specific case of mobility provided by cellular networks, some security features could be implemented through a Cloud Radio Access Network model. However, the delegation "outside" the device makes the access to the cloud an additional point of fragility, which should be carefully assessed.

There is room for run-time optimizations. Since algorithms with similar security degree have different consumptions, some optimizations could be performed within the ISP or the datacenter. For instance, switching the security mechanisms to more energy-efficient ones if there are no foreseen risks. In other words, a choral coordination among firewalls, network probes, and anticipatory security systems could allow to trade between security and energy efficiency, if needed.

Precisely knowing the energy required by security mechanisms can be used as a novel marker to perform anomaly detection and to prevent non-scalable or computationally intensive traffic analysis. For instance, a growth in the power consumption could reveal the presence of some form of Denial of Service (DoS) or Distributed DoS (DDoS) attacks (see, e.g., [5] and the references therein). However, the detailed investigation of these topics, including long-lasting DDoS attacks, is let to future works.

Lastly, as a part of future developments, the approach proposed in this paper could be used to improve high-level models such as the one reported in [15] to provide an online estimation of the used power. This can lead to additional benefits: (*i*) refine the estimation of the energy drained in mobile nodes to help the optimization of architectures, (*ii*) enhance models used to quantify the energy used by Internet-scale service providers by explicitly considering the contribution of security-related algorithms, and (*iii*) develop novel traffic analysis techniques able to correlate consumption with loads, for instance to early detect attacks.

Acknowledgement

This work has been partially supported by the research project STARTECO, funded by the Italian Ministry of University and Research (MIUR), project id DM62554.

References

[1] J. Jang-Jaccard, S. Nepal, "A Survey of Emerging Threats in Cybersecurity", Journal of Computer and System Sciences, Vol. 80, No. 5, Aug. 2014, pp. 973-993.

[2] S. Subramanya, Z. Mustafa, D. Irwin, P. Shenoy, "Beyond Energy-Efficiency: Evaluating Green Datacenter Applications for Energy-Agility", in Proc. of the 7th ACM/SPEC on International Conference on Performance Engineering, pp. 185-196, March 2106.

[3] A. Bianzino, C. Chaudet, D. Rossi, J. Rougier, "A Survey of Green Networking Research", IEEE Communications Surveys & Tutorials, Vol. 14, No. 1, First Quarter 2012, pp. 3-20.

[4] M. Gaggero, L. Caviglione, "Predictive Control for Energy-Aware Consolidation in Cloud Datacenters", IEEE Transactions on Control Systems Technology, Vol. 24, No. 2, March 2016, pp. 461-474.

[5] A. Merlo, M. Migliardi, L. Caviglione, “A Survey on Energy-Aware Security Mechanisms”, Pervasive and Mobile Computing, Vol. 24, Dec. 2015, pp. 77-90.

[6] N. R. Potlapally, S. Ravi, A. Raghunathan, N. K. Jha, “A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols”, IEEE Transactions on Mobile Computing, Vol. 5, No. 2, Feb. 2006, pp. 128-143.

[7] N. Vallina-Rodriguez, J. Crowcroft, “Energy Management Techniques in Modern Mobile Handsets”, IEEE Communications Surveys & Tutorials, Vol. 15, No. 1, First Quarter 2013, pp. 179-198.

[8] X. Li, F. T. Chong, “A Case for Energy-Aware Security Mechanisms”, in Proc. of the 27th International Conference on Advanced Information Networking and Applications, March 2013, pp. 1541-1546.

[9] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, L. Uhsadel, “A Survey of Lightweight-Cryptography Implementations”, IEEE Design & Test of Computers, vol. 24, no. 6, pp. 522-533, 2006.

[10] H. Krawczyk, M. Bellare, R. Canetti, “HMAC: Keyed-Hashing for Message Authentication”, RFC 2104, Network Working Group, Feb. 1997, IETF.

[11] H. Alanazi, B. B. Zaidan, A. A. Zaidan, H. A. Jalab, M. Shabbir, Y. Al-Nabhani, “New Comparative Study Between DES, 3DES and AES Within nine Factors”, Journal of Computing, vol. 2, no. 3, pp. 152-157, March 2010.

[12] L. Caviglione, M. Gaggero, J.-F. Lalande, W. Mazurczyk, M. Urbanski, “Seeing the Unseen: Revealing Mobile Malware Hidden Communication via Energy Consumption and Artificial Intelligence”, IEEE Transactions on Information Forensics and Security, vol. 11, no. 4, pp. 799-810, April 2016.

[13] W. Trappe, R. Howard, R. S. Moore, “Low-Energy Security: Limits and Opportunities in the Internet of Things”, in IEEE Security & Privacy, vol. 13, no. 1, pp. 14-21, Jan.-Feb. 2015.

[14] F. Palmieri, S. Ricciardi, U. Fiore, “Evaluating Network-Based DoS Attacks under the Energy Consumption Perspective: New Security Issues in the Coming Green ICT Area”, in Proc. of the International Conference on Broadband and Wireless Computing, Communication and Applications, Oct. 2011, pp. 374-379.

[15] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, L. Yang, “Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones”, in Proc. of the IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Oct. 2010, pp. 105-114.

Biographies

Luca Caviglione received the Ph.D. degree in electronics and computer engineering from the University of Genoa, Italy. Since 2007 he has been a Researcher with the Institute of Intelligent Systems for Automation, National Research Council of Italy. His research interests include P2P systems, wireless communications, cloud architectures, and network security. He is an Associate Editor of the Transactions on Emerging Telecommunications Technologies.

Mauro Gaggero received the B.Sc. and M.Sc. degrees in electronics engineering and the Ph.D. degree in mathematical engineering from the University of Genoa, Italy, in 2003, 2005, and 2010, respectively. Since 2011, he has been a Researcher with the Institute of Intelligent Systems for Automation, National Research Council of Italy. His research interests include control, optimization, and learning from data. He is an Associate Editor of the IEEE Control Systems Society Conference Editorial Board.

Enrico Cambiaso received the Ph.D. in 2016 in Computer Science from the University of Genoa, Italy, with a thesis titled “Design and Development of Slow DoS Attacks”. His scientific interests are related to computer and network security, communication protocols, cyber-attacks, intrusion detection systems, covert channels, and cloud computing.

Maurizio Aiello graduated in 1994, worked as free-lance consultant for universities, research centers, and private industries. From August 2001, he is responsible of the network infrastructure of the National Research Council of Italy. He is a teacher at the University of Genoa and University College of Dublin. He is a student coordinator, and manages fellowships and EU projects in the computer security field. His research interests are on network security and protocols.