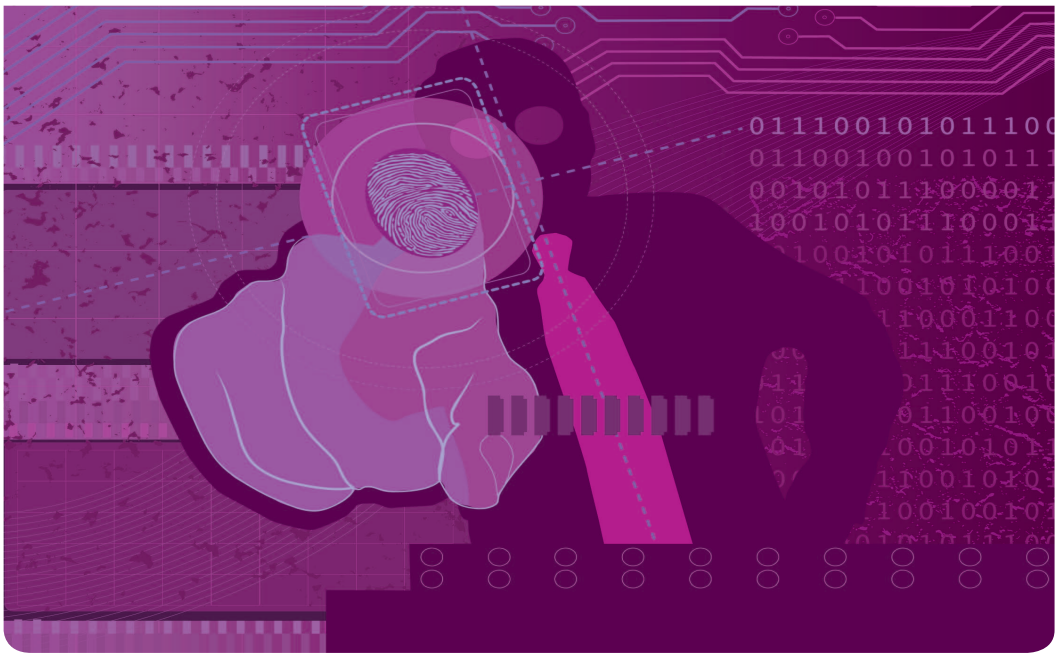
By Shunli Zhang,
Laurence T. Yang,
Liwei Kuang,
Jun Feng,
Jinjun Chen,
and Vincenzo Piuri



©ISTOCKPHOTO.COM/WHA4

A Tensor-Based Forensics Framework for Virtualized Network Functions in the Internet of Things

Utilizing tensor algebra in facilitating more efficient network forensic investigations.

With the ever-increasing network traffic and Internet connectivity of smart devices, more attack events are being reported. As a result, network forensics remains a topic of ongoing research interest in the Internet of Things (IoT). In this article, we present a novel tensor-based forensics approach for virtualized network functions (VNFs). An event tensor model is proposed to formalize the network events, and then, it is used for effectively updating the core event tensor. We then introduce a similarity tensor model to integrate the core event tensors on the orchestration and management layer in the network function virtualization (NFV) framework. Finally, we present an evidence tensor model for network forensics, where we demonstrate how evidence tensors can be merged.

With the rapid development of electronics and communications technologies, the computing power, storage capabilities, and energy capacity of small smart devices [1], [2] have greatly improved. The IoT enables connected things with new capabilities to provide unlimited services for humans. It consists of billions of interconnected cross-domain heterogeneous

Tensor is a relatively new concept, although it has been widely applied in different settings.

entities, in which the entity often refers to a smart sensor, smart actuator, human, or potentially any physical entities or virtual components that can provide/request a service [3]–[5].

Like any other information system, the IoT also depends on the combination of hardware, software, and architectures. The traditional IoT mixes management and processing logic in the same hardware devices, which makes the IoT more complex and tougher to manage. As a new paradigm of the IoT, NFV was proposed to deal with this problem [6] by separating the control plane from the data layer to simplify IoT production. The characteristics and benefits associated with NFV can be broadly categorized into a separation of software from hardware, flexible deployment of network functions, function allocation in hierarchical processing infrastructures, resource allocation, intelligent processing, and dynamic service provisioning.

NFV introduces great granularity, flexibility, elasticity, and visibility to the IoT, but it also brings new security and privacy challenges [6]. For instance, in essence, decoupling the data plane and the control plane in NFV equivalently leaves a door to the attackers for exploiting the vulnerabilities of NFV controllers, application programming interfaces, networking protocols and applications, and further damages the trust relations. Therefore, the security and privacy issues of the IoT need to be studied more and improved, and network forensics is essential in this perspective. While the various network forensic domains, such as email, web, and multimedia, and network traffic analysis [e.g., Internet Protocol (IP) traceback] have been widely studied, NFV has not yet been explored enough [7].

In this article, we focus on NFV forensics by employing a tensor model. Tensor has been broadly utilized in applications. This is not surprising as tensor is known to be efficient and effective for data representation, where high-quality core data can be appropriately extracted by employing tensor decomposition [8] methods. The latter has also been used to analyze and mine data in diverse research fields, such as face recognition and information retrieval. We observe that a tensor model can also be used to efficiently represent network events and forensic evidence. By integrating the tensor model with NFV, network forensic effects can be greatly enhanced. Specifically, in this article, we demonstrate how tensor algebra can be utilized in NFV to facilitate forensic investigations.

EXISTING WORKS

NFV allows one to decouple the network function from the specific hardware [9] and can be used to support VNFs on virtual machines. The VNFs can also be located in the cloud to allow for greater flexibility, scalability, and efficiency. Tensor is a relatively new concept, although it has been widely applied in different settings. For instance, the extensible order tensor model has been used to represent heterogeneous data as a unified model. Tensor decomposition [8] is a powerful data analysis tool in data-driven applications (e.g., big data), such as trend estimation and multiclustering. High-order singular value decomposition (HOSVD) is a popular tensor decomposition method and can be utilized for data mining domains, such as data reduction and tag recommendations. While there have been separate studies focusing on network forensics, the tensor model, and NFVs, no study has examined the potential to integrate a tensor model and NFV to facilitate more efficient network forensic investigations. This is the gap we deal with in this article.

PROPOSED TENSOR-BASED FORENSICS MODEL

HARDWARE RESOURCES AND VIRTUALIZATION LAYERS

The hardware resources layer in a forensics model (Figure 1) consists of computing devices, storage equipment, and network equipment. All of these resources are employed to provide the underlying capabilities to support high-level functions, and they can be considered as the infrastructure. These resources are managed by the virtualization layer, which changes the physical devices to a logical view to provide a uniform resource pool.

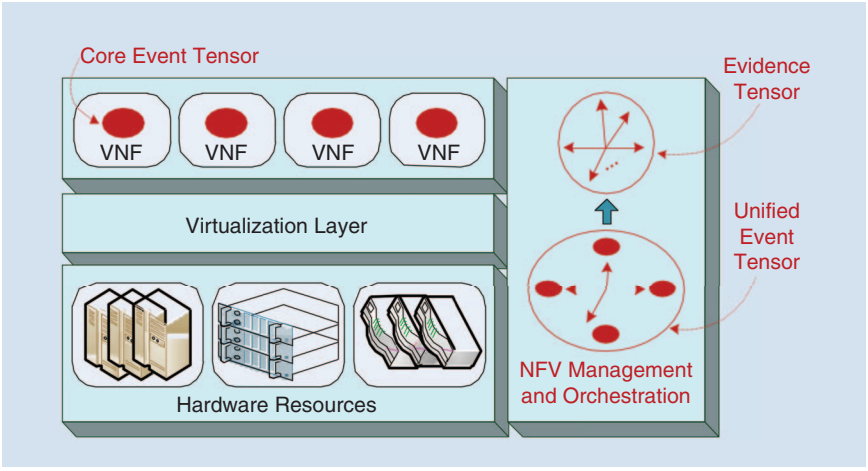


FIGURE 1. The proposed tensor-based forensics model.

VNF AND A CORE EVENT TENSOR MODEL

The VNFs execute on the virtualization layer. In our approach, we construct the evident tensor model for each VNF. The HOSVD method is used for extracting the core event tensor, which is then uploaded to the management and orchestration layer for integration and construction of the evidence model. The

incremental method is employed to dynamically update the core event tensor with the evolving data stream.

MANAGEMENT AND ORCHESTRATION LAYER

This module is in charge of unifying the collective core event tensor models and the construction of the evidence tensor model. In our approach, we also present a similarity tensor to measure the similarities of the core event tensor and fuse them as a loosely integrated event tensor model. Then, the probabilities are computed to construct the evidence tensor model for network forensic investigations.

CONSTRUCTION OF AN EVENT TENSOR MODEL

EVENT TENSOR MODEL

Let $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \ldots I_P}$ denote a Pth-order tensor. The data characteristics can be represented as tensor orders $I_1, I_2, \ldots, I_p$. This article formalizes the network events in network devices to a tensor model. We can build a seventh-order event tensor model $\mathcal{T}$ event$\in \mathbb{R}^{I_t \times I_{\text{sip}} \times I_{\text{tip}} \times I_{\text{sport}} \times I_{\text{dporp}} \times I_{\text{dev}} \times I_{\text{des}}}$, and the tensor orders $I_t, I_{\text{sip}}, I_{\text{tip}}, I_{\text{sport}}, I_{\text{dporp}}, I_{\text{dev}}$, and I_{des} denote time, source IP, destination IP, source port, destination port, devices, and event description, respectively. All network events detected in the networking devices are represented as event tensor models.

CORE EVENT TENSOR AND INCREMENTAL UPDATING METHOD

A tensor decomposition method is used for extracting core event information from the proposed event tensor model. To a primitive tensor $\mathcal{T}$, the core tensor $\mathcal{S}$ and approximation tensor $\mathcal{S}'$ can be computed by utilizing the HOSVD method (for the detailed calculation process, refer to [8]).

In this article, an incremental method is utilized for effectively updating the core event tensor of the primitive event tensor model. Assume that matrix $\mathrm{T}_1 = \mathrm{U}_1 \Sigma_1 \mathrm{V}_1^T$, T_2 is a new arrived matrix, then the new matrix $[\mathrm{T}_1, \mathrm{T}_2]$ can be incrementally and dynamically broken by projecting the arrived columns of matrix T_2 to the truncated matrix T_1. The linear operations are performed when the added blocks reach. And then the updated truncated matrix and singular values are uploaded to the management and orchestration layer. Leveraging the incremental method, the network devices can effectively update the core event tensors. Accordingly, the similarity tensor model and the evidence tensor model will be changed based on the updated core event tensor models.

SIMILARITY TENSOR AND EVIDENCE TENSOR

SIMILARITY TENSOR MODEL

Figure 2 displays the sixth-order similarity tensor model $\mathcal{T}_{\text{sim}} \in \mathbb{R}^{I_{\text{time}} \times I_{\text{location}} \times I_{\text{port}} \times I_{\text{des}} \times I_{\text{event}} \times I_{\text{event}}}$, where $I_{\text{time}}, I_{\text{location}}, I_{\text{port}}$, and I_{des} refer to the time, location, port, and description, respectively. Tensor order I_{event} denotes the network event. We used four dimensions in the port order to measure the port similarity. For example, in the right-hand table of Figure 2, the

An incremental method is utilized for effectively updating the core event tensor of the primitive event tensor model.

dimension value three denotes that the two network events have the same destination port but different source ports. The tensor subspace $\mathcal{T}(:,:,0,0,:,0)$ is used for representation of a network event, while the subspace $\mathcal{T}(0,0,:,0,:,:)$ is used to measure the port similarity of two network events. For instance, in Figure 3, the tensor element $t(j,k,0,0,i,0)$ reveals that a network event i is inspected at time j and in location k. The tensor element $t(0,0,1,0,m,n)$ denotes that the network event m and n have the same source port and destination port. The tensor order I_{des} is employed for the description of a network event. The description content of a network event can be coded as ASCII and represented in the tensor subspace $(0,0,0,:,:,0)$. We use the description order to represent some special similar characteristics of network events.

EVIDENCE TENSOR MODEL

We now describe our evidence tensor model for network forensics. The fourth-order evidence tensor is defined as $\mathcal{T}_{\text{evidence}} \in \mathbb{R}^{I_{\text{node}} \times I_{\text{node}} \times I_{\text{nattr}} \times I_{\text{eattr}}}$, where the tensor orders I_{node}, I_{nattr}, and I_{eattr} denote network nodes, node attributes, and edge attributes, respectively. For instance, in Figure 3, suppose the node Node_1 remotely installs NetSpy on Node_2, then the tensor element $\mathrm{t}(1,0,0,0)$ is equal to 0.4 and tensor

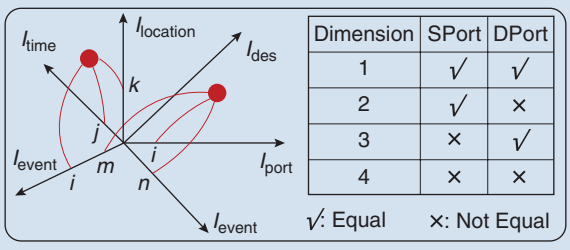


FIGURE 2. A sixth-order similarity tensor model.

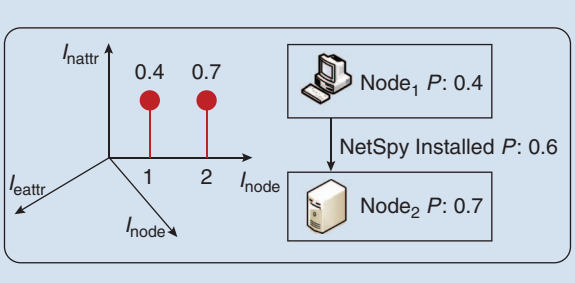


FIGURE 3. The fourth-order evidence tensor.

element t(2, 0, 0, 0) is 0.7. The edge is denoted as a tensor element t(1, 2, 0, 0), whose value is 0.6. Here, we can refer to the calculation of probabilities for tensor elements [10]. The description NetSpy Installed for this edge is represented along the tensor order I_{eattr}.

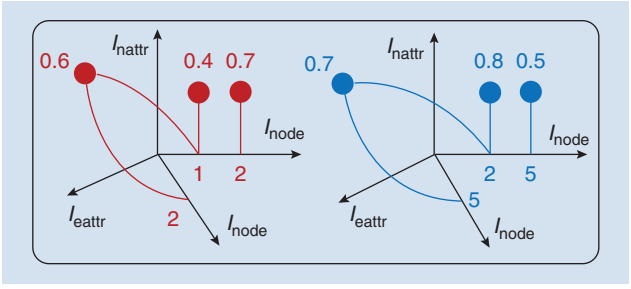


FIGURE 4. Merging the two evidence tensor models.

MERGING EVIDENCE TENSOR MODELS

Inspired by the approaches in [10], we propose merging the evidence tensor models to generate a new evidence tensor model. Upon conclusion of the merging process, new tensor elements are generated, and the probabilities for the network nodes and edges are updated. For example, in Figure 4, the value of tensor element $t_1(1, 0, 0, 0)$ is equal to 0.4, $t_1(2, 0, 0, 0)$ is 0.7, $t_1(1, 2, 0, 0)$ is 0.6, the value of tensor element $t_2(2, 0, 0, 0)$ is equal to 0.8, $t_2(5, 0, 0, 0)$ is 0.5, and $t_2(2, 5, 0, 0)$ is 0.7. After the merging process, the values of tensor elements change to 0.7, 0.94, and 0.88, respectively.

In the proposed tensor-based forensics approach, the merging of the evidence tensor models is executed at the management and orchestration layer in the NFV framework. This newly generated evidence tensor model can then be used for network forensic investigations.

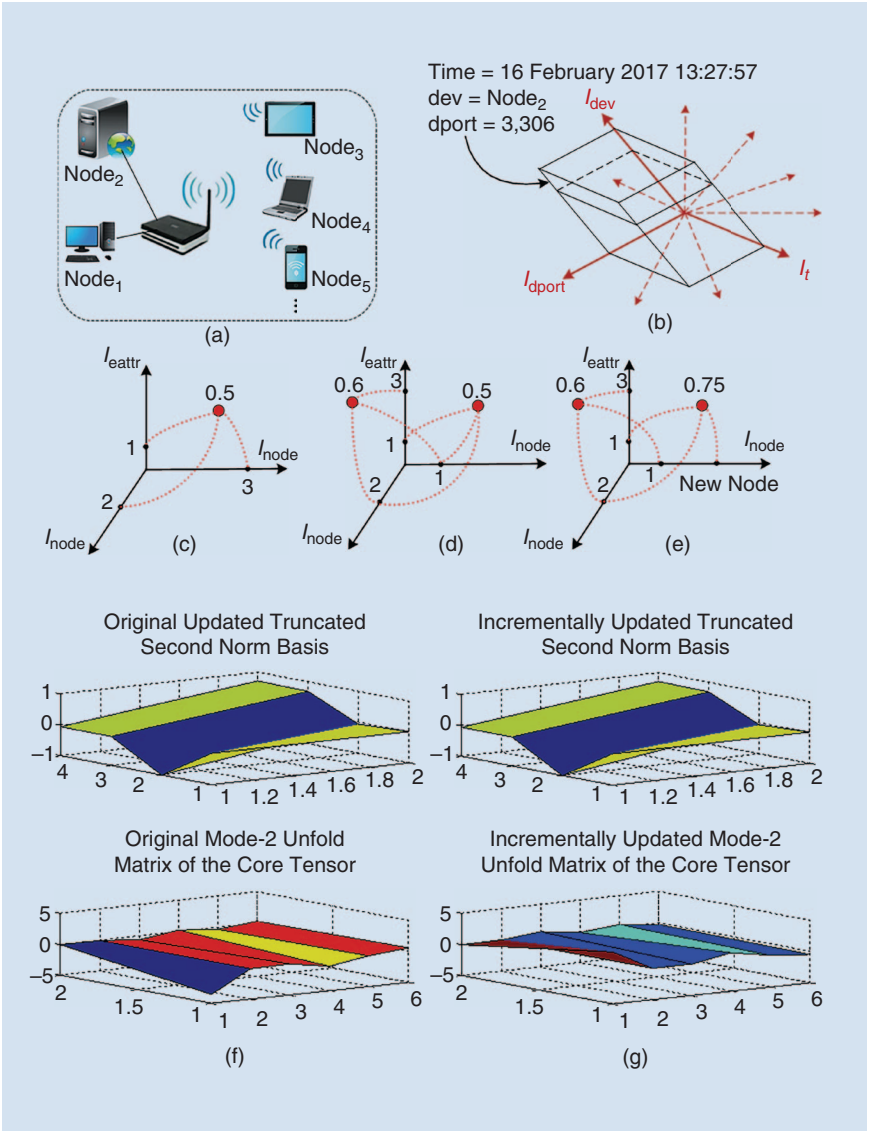


FIGURE 5. A case study for the tensor-based forensics approach. (a) An experiment network, (b) an event tensor, (c) and (d) two separated evidence tensors, (e) the integrated evidence tensor, and (f) and (g) a visualization of the HOSVD and the incremental HOSVD method, respectively.

A CASE STUDY

Data security is one of the most serious problems in the IoT because intruders have a large scope for attack possibilities. The intuitive examples are traffic jams and malicious activities. Here, we explain how the proposed tensor-based forensics method can be used to construct a network attack model comprising commonly seen attacks, such as port scans, repeated login attempts and others related attacks in IoT.

THE NETWORK STRUCTURE

The experiments were carried out on a platform constructed with Ubuntu 14.04, OpenStack Kilo. The VNF consists of a virtual firewall, virtual intrusion detection system, and virtual deep packet inspection. As shown in Figure 5(a), port scanning and denial-of-services attack tools were installed in the attacker computer ($Node_1$), and the target victim computers were in the same network as the attacker-controlled computer.

DESCRIPTION OF NETWORK ATTACKS

In our case study, the attacker-controlled computer utilized the port scanning tool to collect the Transmission Control Protocol (TCP) and User Datagram Protocol port information from the targeted computers, and it knew that a particular computer node, $Node_2$, opened a mass of TCP port. Thus, the attacker could select a specific port and initiate a

network attack. Here, we assume that the login attempt attack to $Node_2$ was conducted by the attacker-controlled $Node_1$. Utilizing the port 3306 vulnerability, the attacker sent the user name and password to the My Structured Query Language (MySQL) database server on $Node_2$ and successfully logged in to $Node_2$'s database server.

EVENT TENSOR AND EVIDENCE TENSOR

Figure 5(b) details the event tensor model constructed with the port scanning and login attempt information. The subtensor space, consisting of the tensor orders I_{time} and I_{dev}, and I_{dport} describes the login attempt event. At 13:27:57 on 26 February 2017, the TCP port 3306 was remotely connected to $Node_2$. This network event reveals that the attacker ($Node_1$) attempts to login to the MySQL server ($Node_2$). Using the event tensor, we first construct a third-order evidence subtensor, $\mathcal{T}_{evidence} \in \mathbb{R}^{I_{node} \times I_{node} \times I_{eattr}}$, where the tensor orders I_{node} and I_{eattr} denote network nodes and edge attributes, respectively. In this scenario, we assume that there are four computer hosts and three network attack methods as previously discussed. Three types of network attack can generate 2^3 attack combinations. Hence, $\mathcal{T}_{evidence} \in \mathbb{R}^{4 \times 4 \times 8}$. Figure 5(c) and (d) shows two different third-order evidence tensors, and Figure 5(e) is the merged evidence tensor.

REDUCTION AND INCREMENTALLY UPDATING OF EVIDENCE TENSOR

After obtaining the integrated evidence tensor, the original tensor data are processed for data reduction using both the HOSVD and the incremental HOSVD methods. Then, we obtain the most valuable core tensor and truncated matrixes. The $4 \times 4 \times 8$ primitive evidence tensor can be decomposed into three $4 \times 2, 4 \times 2, 8 \times 3$ truncated matrices and a $2 \times 2 \times 3$ core tensor. Here, the data reduction rate is 40.625%, thus achieving a savings of 59.375% in storage space. This, consequently, results in an improved computational efficiency. When the tensor elements are updating, employing the incremental method computing the truncated matrices and core tensor can be efficiently improved. Figure 5(f) and (g) presents the findings from using the HOSVD and the incremental HOSVD method, respectively.

CONCLUSION

Network forensics are increasingly important in our interconnected digital society. Thus, it is important for forensic techniques to keep pace with technological advancements. We demonstrated the potential of utilizing tensor algebra in facilitating more efficient network forensic investigations. More specifically, we proposed a tensor-based forensics model for VNFs. An event tensor model was then used to represent network events, and the incremental method was utilized to efficiently update the generated event tensor. The updated core event tensors were then submitted to the management and orchestration layer for integration. We also introduced a similarity tensor model to fuse the event tensors and an evidence tensor model to facilitate network forensic investigations. Finally, we demonstrated the practicality of this approach using a case study in the IoT.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Plan of China (2017YFB0801804) and Shenzhen Fundamental Research Program (JCYJ20170307172200714).

ABOUT THE AUTHORS

Shunli Zhang (shunlizh2018@gmail.com) is a lecturer in Information Science and Technology at Jiujiang University, China.

Laurence T. Yang (ltyang@ieee.org) is a professor of computer science at Huazhong University of Science and Technology, China; Shenzhen Huazhong University of Science and Technology Research Institute, China; and St. Francis Xavier University, Canada.

Liwei Kuang (kuanglw@139.com) is a software engineer with Fiber Home Technologies Group.

Jun Feng (junfeng989@gmail.com) is a Ph.D. degree student in computer science and technology at Huazhong University of Science and Technology, Wuhan, China.

Jinjun Chen (jinjun.chen@gmail.com) is a professor in the Faculty of Science, Engineering and Technology at Swinburne University of Technology, Australia.

Vincenzo Piuri (vincenzo.piuri@unimi.it) is a professor at the University of Milan, Italy, and a visiting professor at the University of Texas at Austin and George Mason University, Fairfax, Virginia.

REFERENCES

[1] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of Things is the backbone," *IEEE Consum. Electron. Mag.*, vol. 5, no. 3, pp. 60–70, 2016.

[2] H. Thapliyal, R. K. Nath, and S. P. Mohanty, "Smart home environment for mild cognitive impairment population: Solutions to improve care and quality of life," *IEEE Consum. Electron. Mag.*, vol. 7, no. 1, pp. 68–76, 2018.

[3] C. Zhu, V. C. M. Leung, L. Shu, and C. H. Ngai, "Green Internet of Things for smart world," *IEEE Access*, vol. 3, pp. 2151–2162, Jan. 2015.

[4] J.-H. Lee and H. Kim, "Security and privacy challenges in the Internet of Things [security and privacy matters]," *IEEE Consum. Electron. Mag.*, vol. 6, no. 3, pp. 134–136, 2017.

[5] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *Proc. IEEE World Forum on Internet of Things*, 2014, pp. 287–292.

[6] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 2016.

[7] M. D. Leom, K.-K. R. Choo, and R. Hunt, "Remote wiping and secure deletion on mobile devices: A review," *J. Forensic Sci.*, vol. 61, no. 6, pp. 1473, 2016.

[8] J. Feng, L. T. Yang, Q. Zhu, and K. K. R. Choo, "Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment," *IEEE Trans. Dependable Secure Comput.*, to be published. doi: 10.1109/TDSC.2018.2881452.

[9] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, 2015.

[10] A. C. Liu, A. Singhal, and D. Wijesekera, "Creating integrated evidence graphs for network forensics," in *Proc. 9th IFIP Int. Conf. Digital Forensics*, 2013, pp. 227–241.