Migrating Monitors + ABE: A Suitable Combination for Secure IoT?

Gordon J. Pace[1], Pablo Picazo-Sanchez[2], and Gerardo Schneider[2⋆]

University of Malta, Malta
{gordon.pace@um.edu.mt}
University of Gothenburg, Sweden
{pablop@chalmers.se, gersch@chalmers.se}

Abstract. The rise of the Internet of Things brings about various challenges concerning safety, reliability and dependability as well as security and privacy. Reliability and safety issues could be addressed by using different verification techniques, both statically and at runtime. In particular, migrating monitors could effectively be used not only for verification purposes, but also as a way to gather information and to enforce certain policies. The addition of monitors, however, might introduce additional security and privacy threats. In this extended abstract we briefly sketch ideas on how to combine migrating monitors with a public cryptographic scheme named Attribute-Based Encryption as a way to ensure monitors are run by the right devices in a secure and private manner.

1 Introduction

The Internet of Things (IoT) is used to refer to the pervasive network of interconnected devices embedded in everyday things —sensors, actuators, devices, and applications for sharing information among them. Usual devices on the IoT include RFID (Radio Frequency IDentification) tags, smartphones, smartwatches, Implantable Medical Devices (IMD), and many other gadgets with communication capabilities.

IoT inherits most of the challenges of distributed systems due the non-locality of data collection and computation. In particular monitoring of such systems presents a wide range of challenges [6, 5, 4, 20] since monitors might need information from other devices in order to duly perform their tasks.

The fact that monitoring cares about what goes on in different locations, it is clear that a monolithic local monitor is not enough. Different monitor instrumentation strategies have been proposed in the literature (e.g., [12]). The approaches can be largely split into two categories: (i) *centralised* or *orchestration approaches* in which the monitor is centrally located, receiving all relevant data and event-notification from the different nodes (e.g., [3]); and (ii) *choreography-based approaches*, in which the monitor is statically split into local parts instrumented in the different locations, and communicates only when

⋆ Corresponding Author.

as required (e.g., [7]). Both approaches, however, pose challenges when applied to IoT environments. The former approach suffers from increased communication (with the central monitoring node), which grows as the number of nodes increases, resulting in slowing down of the overall system and an increase in power consumption. The major challenge with the latter approach is that for many logics, splitting the monitors in an effective manner can be difficult [4, 20]. Furthermore, when nodes might be discovered at runtime, static decomposition of properties can be impossible to perform [12].

Migrating monitors is another approach proposed in the literature [11] based on dynamic choreography —instrumenting monitors locally, but giving them the ability to migrate to other locations when the need to access data or events from elsewhere becomes necessary. This last solution can be particularly suited for IoT environments where most of the correctness can be established locally. This approach avoids a blow-up in the amount of communication of generated data from IoT sensors.

Note that we have so far mentioned monitoring IoT without specifying in detail what the tasks of the monitors are. We should distinguish here three different applications of monitoring: (i) *Proper monitoring*, where the monitor collects data, possibly performing side-effect free computations (e.g., calculate an average during a specific amount of time) other than logging the information or sending it to another device, monitor or node in the network; (ii) *Runtime verification*, where the data is used for verification with respect to properties specifying what the expected behaviour of the system should be. Given the decentralised nature of IoT networks, such properties may be enacted by any of the devices or parties participating in the network, with the monitor usually being automatically generated from the property (e.g., [14]); (iii) *Runtime enforcement* takes this one step further by having the monitors carry code to be executed in the monitored system, send specific commands to control the system, in order to enforce a given property (as mentioned in *runtime verification*) by not allowing the system to act differently than the specified property (e.g., [10]).

The complexity, and degree of intrusion increase with these levels of monitoring. Since monitors can effectively leak information about the state of other entities on the system, we envisage a policy (or policies) which comes with the IoT scenario, and which specifies what types of properties can be enacted by which users e.g., a policy in a hospital context may state that no patient may enact a property that monitors events occurring on another patient's device.

Besides all the above issues, IoT monitoring is challenging due to the nature of the sensors: they are highly constrained in terms of computation, memory, battery and storage capabilities. As a consequence, monitors should be able to run under those constraints. Another challenge is that the IoT topology changes continuously over time because new sensors might be added and others are removed from the network. Migrating monitors might help here since they could automatically migrate to the new nodes when added, and they might eventually be killed when nodes disappear, without affecting the overall monitoring system.

There is, however, a problem when using migrating monitors in both orchestration and choreography-based approaches if deployed in an IoT scenario: security and privacy concerns. Migrating monitors are small software components that travel from one node to another one to either collect data and perform small computations (proper monitoring), verify some properties (runtime verification) or enforce some properties (runtime enforcement). IoT systems are networks composed of subnetworks each containing confidential local information, therefore the migrating monitors should not leak that information nor the architecture to the rest of the system.

Security and privacy concerns on IoT have been considered to be amongst the most challenging open issues nowadays (e.g., [15, 2, 18, 21]), and *Attribute Based Encryption* (ABE) has been identified as one of the more promising cryptographic schemes to secure such systems [1, 22]. ABE is a form of public key encryption where the information is encrypted under a boolean formulae (called *access policy*) which other parties must satisfy in order to decrypt the ciphertext. This cryptographic scheme is particularly useful on IoT since it simultaneously provides fine-grained access control and encryption [17]. Even though many theoretical proposals have been published in this area, only few works have deployed this cryptographic scheme on high-constrained IoT devices [13, 16, 22, 23].

2 Combining Migrating Monitors and ABE for Secure IoT

The use of migrating monitors provides a way of augmenting IoT functionality, side-by-side with ABE which provides guarantees that there are no additional threats (in terms of security and privacy) due to the newly injected functionality.

Our proposed approach to achieve secure and private migrating monitors in IoT would work as explained below:

(i) We provide a monitoring policy specification language, which will specify which users[1] are allowed to enact what type of monitors on the network. This will be used to regulate monitors which will be enacted dynamically.

(ii) We provide a formal language to define migrating monitors integrated with ABE in such a way that it is possible to define which monitors will be executed and where. Monitors can be encrypted under certain access policies (made of attributes and represented as a boolean formulae) such that only those users in the system holding those attributes can satisfy the access policies and thus decrypt the monitors.

(iii) Monitors will be encrypted using a variant of ABE named Multi-Authority Attribute-Based Encryption (MA-ABE) [19]. With this scheme, networks and subnetworks are modelled in the MA-ABE scheme such that we can define the scope of the monitors and thus different subnetworks can share information privately and securely.

[1] Note that in this context, the term *user* may refer to sensors, software components or persons.

(iv) Monitors will statically be checked for the specific purpose they are created and thus identified as proper monitors, runtime verifiers or runtime enforcers. A secure runtime environment to manage monitor control-logic migrating from one IoT device to another is added to the IoT system, which also guarantees that monitors can only be executed following their main purpose. For instance, if a specification is tagged as a proper monitor (and not, for instance, as an enforcer), it will not be allowed to change the state of the devices and actuators, and will be limited to send control-flow messages to other monitor managers.

(v) By allowing users to arbitrarily create new monitors according to the monitoring policies in place, an authentication system must guarantee that only certified monitors can be run in the system.

3 Conclusions

We believe that there is great potential in using migrating monitors on IoT, combined with ABE to guarantee that monitors do not pose new security and privacy issues. In this paper, we have only presented some initial ideas and sketched a general way to achieve an IoT architecture were such monitors may run increasing functionality while not adding new security and privacy concerns. Although here we have not presented a formal argument to show that in this manner we do not introduce any new security and privacy threats, we believe that the cryptographic properties of ABE, and additional measures added at the architectural and monitoring management level, can ensure this to be the case. A more technical presentation of this work would require formal proofs to show that the combination is not vulnerable to attacks. In what concerns the practical side, we are considering the implementation of the above into the tool Larva [9], by extending DATEs [8] (the underlying automata-based specification language of Larva) with primitives from ABE. One aspect of combining migrating monitors and ABE that has not been addressed in our paper, and thus left as future work, is the use of our approach in order to provide additional security and privacy guarantees to the IoT.

Acknowledgements This research has been partially supported by the Swedish Research Council (*Vetenskapsrådet*) under grant Nr. 2015-04154 (*PolUser: Rich User-Controlled Privacy Policies*).

References

1. E. AbuKhousa, N. Mohamed, and J. Al-Jaroodi. e-health cloud: Opportunities and challenges. *Future Internet*, 4(3):621, 2012.
2. L. Atzori, A. Iera, G. Morabito, and M. Nitti. The Social Internet of Things (SIoT) — When social networks meet the Internet of Things: Concept, architecture and network characterization. *Computer Networks*, 56(16):3594 – 3608, 2012.

3. S. Azzopardi, C. Colombo, J. Ebejer, E. Mallia, and G. J. Pace. Runtime verification using VALOUR. In *RV-CuBES*, volume 3 of *Kalpa Publications in Computing*, pages 10–18. EasyChair, 2017.
4. A. Bauer and Y. Falcone. Decentralised LTL monitoring. *Formal Methods in System Design*, 48(1-2):46–93, 2016.
5. A. Bauer, M. Leucker, and C. Schallhart. Model-based runtime analysis of distributed reactive systems. In *17th Australian Software Engineering Conference (ASWEC 2006), 18-21 April 2006, Sydney, Australia*, pages 243–252, 2006.
6. B. Bonakdarpour, P. Fraigniaud, S. Rajsbaum, and C. Travers. Challenges in fault-tolerant distributed runtime verification. In *Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part II*, pages 363–370, 2016.
7. C. Colombo and Y. Falcone. Organising LTL monitors over distributed systems with a global clock. In *Runtime Verification - 5th International Conference, RV 2014, Toronto, ON, Canada, September 22-25, 2014. Proceedings*, pages 140–155, 2014.
8. C. Colombo, G. J. Pace, and G. Schneider. Dynamic event-based runtime monitoring of real-time and contextual properties. In *13th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'08)*, volume 5596 of *LNCS*, pages 135–149. Springer-Verlag, 2009.
9. C. Colombo, G. J. Pace, and G. Schneider. LARVA — Safer Monitoring of Real-Time Java Programs (Tool Paper). In *7th IEEE International Conference on Software Engineering and Formal Methods (SEFM'09)*, pages 33–37. IEEE Computer Society, 2009.
10. Y. Falcone, L. Mariani, A. Rollet, and S. Saha. Runtime failure prevention and reaction. In *Lectures on Runtime Verification - Introductory and Advanced Topics*, volume 10457 of *LNCS*, pages 103–134. Springer, 2018.
11. A. Francalanza, A. Gauci, and G. J. Pace. Distributed system contract monitoring. *J. Log. Algebr. Program.*, 82(5-7):186–215, 2013.
12. A. Francalanza, J. A. Pérez, and C. Sánchez. Runtime verification for decentralised and distributed systems. In *Lectures on Runtime Verification - Introductory and Advanced Topics*, pages 176–210. 2018.
13. L. Guo, C. Zhang, J. Sun, and Y. Fang. A privacy-preserving attribute-based authentication system for mobile health networks. *IEEE Transactions on Mobile Computing*, 13(9):1927–1941, Sept 2014.
14. K. Havelund and G. Roşu. Runtime verification. In *Computer Aided Verification (CAV'01) satellite workshop*, volume 55 of *ENTCS*, 2001.
15. C. M. Medaglia and A. Serbanati. *An Overview of Privacy and Security Issues in the Internet of Things*, pages 389–395. Springer New York, New York, NY, 2010.
16. P. Picazo-Sanchez, J. E. Tapiador, P. Peris-Lopez, and G. Suarez-Tangil. Secure publish-subscribe protocols for heterogeneous medical wireless body area networks. *Sensors*, 14(12):22619, 2014.
17. Z. Qiao, S. Liang, S. Davis, and H. Jiang. Survey of attribute based encryption. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on*, pages 1–6, June 2014.
18. R. Roman, J. Zhou, and J. Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266 – 2279, 2013.

19. Y. Rouselakis and B. Waters. *Efficient Statically-Secure Large-Universe Multi-Authority Attribute-Based Encryption*, pages 315–332. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
20. K. Sen, A. Vardhan, G. Agha, and G. Rosu. Efficient decentralized monitoring of safety in distributed systems. In *26th International Conference on Software Engineering (ICSE 2004), 23-28 May 2004, Edinburgh, United Kingdom*, pages 418–427, 2004.
21. S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. *Computer Networks*, 76:146 – 164, 2015.
22. X. Wang, J. Zhang, E. M. Schooler, and M. Ion. Performance evaluation of attribute-based encryption: Toward data privacy in the iot. In *2014 IEEE International Conference on Communications (ICC)*, pages 725–730, June 2014.
23. D. J. Wu, A. Taly, A. Shankar, and D. Boneh. *Privacy, Discovery, and Authentication for the Internet of Things*, pages 301–319. Springer International Publishing, Cham, 2016.