To appear:
in IEEE IT Professional Vol: 9 Issue:5, 2017

Locally Centralized, Globally Distributed Authentication and Authorization for the Internet of Things

Hokeun Kim and Edward A. Lee, *University of California, Berkeley*

Abstract— Authentication and authorization are essential parts of basic security processes and are sorely needed in the Internet of Things (IoT). The emergence of edge and fog computing creates new opportunities for security and trust management in the IoT. In this paper, we discuss some existing solutions to establish and manage trust in networked systems and argue that these solutions face daunting challenges when scaled to the IoT. We give a vision of efficient and scalable trust management for the IoT based on locally centralized, globally distributed trust management using an open-source infrastructure with local authentication and authorization entities to be deployed on edge devices.

Key Words—Internet of Things, Network-level security, Access control, Authorization, Authentication, Centralization/decentralization

SECURITY CHALLENGES

On October 21, 2016, domain name service provider Dyn suffered a distributed denial-of-service (DDoS) attack leading to a significant collapse of fundamental infrastructures comprising the Internet. In a DDoS attack, a number of compromised or zombie computers, forming a botnet, send a flood of traffic to the target server, causing a denial of service by exhausting computation and/or communication resources. What made this incident remarkable was the fact that many of compromised computers launching the attack were relatively small devices including printers, webcams, residential gateways, and baby monitors, i.e., the Internet of Things (IoT).

The IoT benefits from connectivity that facilitates collaboration among a variety of computing systems, ranging from sensor nodes and mobile devices to large control systems and cloud computers. The IoT closely interacts with human beings and physical systems such as medical devices or smart power grids. However, as seen in the Dyn incident, the IoT also brings about security challenges, especially when the "things" lack security.

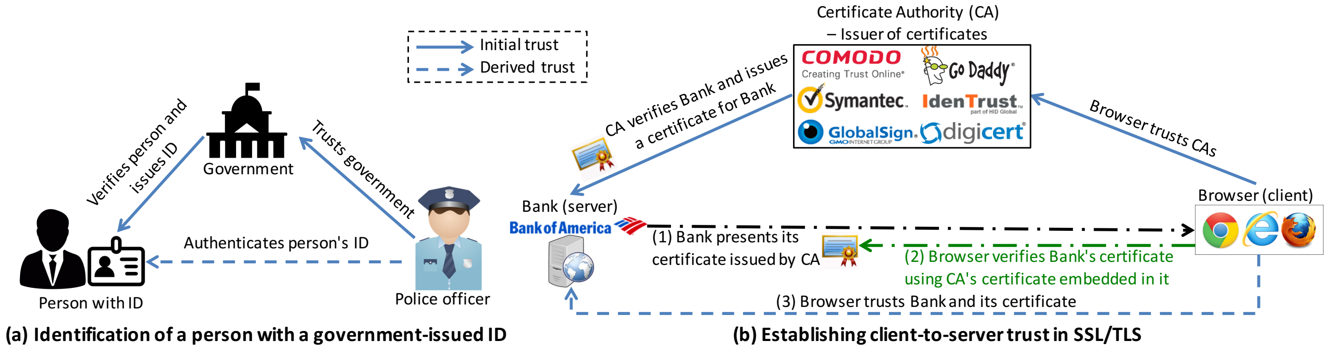
Another incident that showed the threat of the connectivity was the cyberattack on Ukrainian power grid on December 23, 2015 [1]. The attackers gained control over the SCADA (Supervisory Control and Data Acquisition) system of the Ukrainian power grid and caused blackout for several hours in the large area of Ukraine's Ivano-Frankivsk region populated by 1.4 million residents. This case showed that, unlike cyberattacks in the past, the consequences from attacks on the IoT can be more devastating than information theft or financial loss. The consequences can be life-threatening.

In this paper, we propose locally centralized, globally distributed authentication and authorization for the IoT to address these problems.

AUTHORIZATION, AUTHENTICATION AND TRUST

So why did the IoT fall under the attacker's control in two aforementioned incidents? First, things that did not used to be connected are now connected to the Internet, a wilderness full of potential adversaries. Second, the things were not robust enough to be exposed to the wilderness in part because they lacked proper mechanisms for access control.

Access control, or *authorization*, is the process of determining whether an entity (a device or a user) can access resources, e.g., read or write data, execute programs, and control actuators. Authorization also includes denying or revoking access, especially for someone or something malicious. *Authentication*, a process of identifying an entity, is a prerequisite for authorization. In most cases, authorization is not even possible without proper authentication. How can we grant or deny access to someone or something that we do not know about?


Initial trust
Derived trust
Government
Verifies person and issues ID
Trusts government
Authenticates person's ID
Person with ID
Police officer
Certificate Authority (CA) – Issuer of certificates
COMODO Creating Trust Online
Go Daddy
Symantec.
IdenTrust
GlobalSign.
digicert
CA verifies Bank and issues a certificate for Bank
Browser trusts CAs
Bank (server)
Bank of America
Browser (client)
(1) Bank presents its certificate issued by CA
(2) Browser verifies Bank's certificate using CA's certificate embedded in it
(3) Browser trusts Bank and its certificate

(a) Identification of a person with a government-issued ID

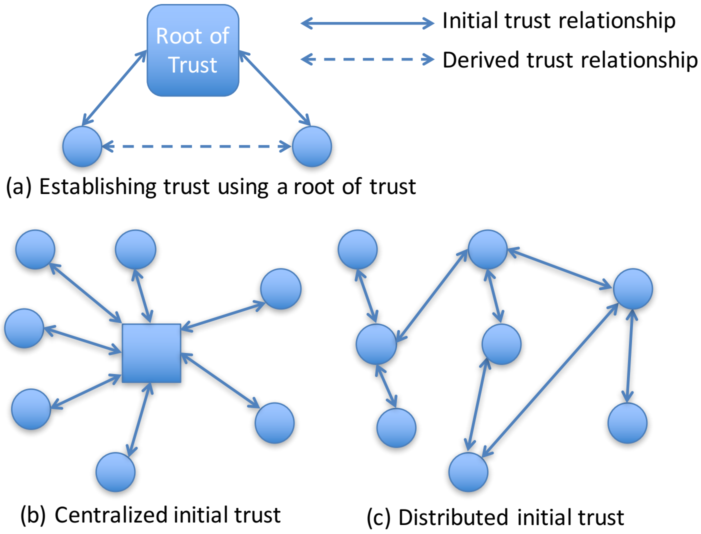
(b) Establishing client-to-server trust in SSL/TLS

Figure 1. Trust and Authentication

Authentication is intrinsically based on *trust*. For example, when we check someone's ID, we first have to trust the issuer of the ID, such as a national government, as shown in Figure 1 (a). The same analogy applies to networked computers. Figure 1 (b) illustrates the process of establishing trust between a browser (client) and a bank's website (server). A number of modern websites use HTTPS, a secure version of HTTP running over SSL/TLS (Secure Socket Layer/Transport Layer Security), a widely used protocol providing channel security guarantees. SSL/TLS uses public-key cryptography for channel establishment; thus, the authenticity of server's public key is critical. A (digital) certificate, a token for authentication, includes the public key of the server. A certificate is issued and digitally signed by a certificate authority (CA). Initially, the browser trusts CA and the CA has issued a certificate for the bank's website. When the browser connects, the web server presents its certificate (Figure 1 (b)-(1)). The browser verifies the bank's certificate using CA's certificate (Figure 1 (b)-(2)). If the verification succeeds, the browser trusts the bank's web server (Figure 1 (b)-(3)). Public Key Infrastructure (PKI) is a set of roles and policies for issuing and managing these certificates.

Indeed, there is a variety of ways to implement tokens for authentication in computer systems. Passwords are the most common ways to authenticate human users. For additional security, we often use two-factor authentication using what we have (e.g., phones) and what we are (e.g., fingerprints) in addition to what we know, the passwords.

In machine-to-machine communications, cryptography is a powerful tool for providing security guarantees. In such crypto systems, cryptographic keys are commonly used as tokens for authentication and also for authorization. Therefore, in many cases, authentication, authorization, and trust management come down to the problems of generating and managing cryptographic keys.



(a) Establishing trust using a root of trust

(b) Centralized initial trust

(c) Distributed initial trust

Figure 2. Ways of building trust in networked systems

Ways of Building Trust

Establishing trust in computing usually starts with a *root of trust*, which constructs an initial trust relationship. Using the root of trust, further trust relationships are derived as shown in Figure 2 (a). A root of trust can be a special hardware component such as a TPM (Trusted Platform Module). Another example is the *root certificate* in PKI.

For networked computers, there are two broad classes of ways to set up the initial trust relationship. One is to ask a centralized authority as in Figure 2 (b), and the other is to use distributed trusted fellows as in Figure 2 (c).

Asking a Centralized Trusted Authority

In centralized trust schemes, the centralized authority is often called a trusted third party because it does not participate in the communication. SSL/TLS based on PKI is one of the most widely used approaches using a centralized CA for authentication. Although there are multiple trusted CAs in PKI, the trust management is still centralized in the same sense that multiple national governments are centralized authorities.

Another widely used approach using a centralized trusted third party is the Kerberos authentication system [2]. Kerberos uses temporary access tokens called *tickets* to authenticate clients and servers and to grant access to the services. Thus, Kerberos provides authorization as well as authentication.

Wireless sensor networks (WSNs), predecessors to the IoT, have their own security solutions. Many WSNs use a base station with plentiful resources for coordination of the battery-powered sensor nodes. Sometimes a base station also works as a root of trust for sensor nodes, especially as a key distributor. One representative example is SNEP (Sensor Network Encryption Protocol) as part of SPINS (Security Protocols for Sensor Networks) [3]. In SNEP, each sensor node shares a secret key called *master key* with its base station. Further keys between sensor nodes are derived using master keys, in other words, based on trust with the base station, a centralized authority.

A pitfall of centralized trust management is that failure of the centralized authority can result in failure of the whole system. One example of this is the WoSign incident that occurred in 2016 [4]. A Chinese certificate authority WoSign mistakenly issued certificates to false subjects; for instance, if you control foobar.github.com, WoSign issued a certificate for *.github.com. This incident showed how the entire security can be broken when the root of trust gets broken.

Using Distributed and Trusted Participants

Distributed trust schemes can avoid the problem of the centralized authority being a single point of failure. In distributed schemes, there is no centralized trusted third party and the participants coordinate autonomously to build further trust.

The concept of a *web of trust* used by OpenPGP [5], an encryption standard widely used for email encryption, leverages trust between participants. OpenPGP uses public keys for encrypting messages. The association of a public key with a recipient is as critical as in PKI. Therefore, OpenPGP also uses certificates, but in this case the certificates are signed by other trusted users rather than a centralized authority. Unlike in PKI, even if a single user gets compromised and has the private key stolen, the effect of the attack would not be catastrophic as in the WoSign incident.

Bitcoin [6], a cryptography-based digital currency, also uses distributed trust. There is no single authority validating Bitcoin transactions. Whenever a Bitcoin transaction occurs, the Bitcoin client broadcasts the transaction to the entire Bitcoin network. Other clients verify the transaction and attach it to a public ledger called a *blockchain*. The blockchain is shared by the Bitcoin clients, therefore it is infeasible for a single malicious Bitcoin client to forge transactions.

LEAP+ [7] is an example security protocol for WSNs, based on distributed trust. LEAP+ also uses base stations, but their involvement is limited to certain tasks such as node initialization. For pairwise key creation and management, the sensor nodes collaborate with neighboring nodes. In this way, LEAP+ can reduce the management overhead of a base station and communication overhead of direct communication between a sensor node and a base station which is usually more costly than communication between neighboring nodes.

In general, security schemes based on distributed trust are more resilient than centralized schemes. And the overhead of authentication and authorization can be distributed to participants, leading to better scalability. However, distributed schemes are more vulnerable to collusion attacks, it is harder to manage and keep track of the whole system, and overhead of individual entities tends to be higher than centralized schemes.

Network Architectures and Trust

There are various aspects that need to be considered for the IoT, including scalability, context-awareness, and ease of deployment, in addition to security and privacy, as Pal [8] points out. The amount of data generated by the IoT is increasing rapidly [9], and demand for real-time processing is surging for cyber-physical systems [10] such as autonomous vehicles or factory floors. To satisfy these needs, Bonomi et al. [11] introduce *fog computing* for the IoT. Fog computing utilizes edge-computing devices which include mobile phones, smart gateways (wireless routers with plentiful computational power) and laptop computers, which can act as a gateway to the Internet.

Edge computing has some advantages compared to cloud computing, which is prevalent in recent days. Lopez et al. [12] make the following points relevant to security and privacy:

1. Private and sensitive data are kept within the edge rather than sent to the centralized cloud, enhancing privacy.
2. The edge has more context information related to the security and privacy, reducing the overhead for the cloud and better serving the heterogeneous entities.
3. Proximity and intelligence at the edge enable real-time interaction with the IoT, predictable latency, and clock synchronization.

Many computing devices with enough resources can play a role as an edge device. An example is the *SwarmBox* that is proposed as a hardware platform in the TerraSwarm project (https://terraswarm.org/). The version 2.0 SwarmBox is an edge computing server hosted in an IMB-186-4300U manufactured by ASRock Inc. SwarmBox 2.0 has the following useful features as an edge device. It supports both WiFi and BLE (Bluetooth Low Energy) for wireless communication, functioning as a smart gateway for devices that connect to the Internet through the SwarmBox. It has dual Ethernet ports, one for the Internet and the other for local networks. The one for the local network is also equipped with hardware support for the IEEE 1588 Precision Time Protocol (PTP), enabling nanoseconds-scale clock synchronization, a desirable property to support real-time systems.

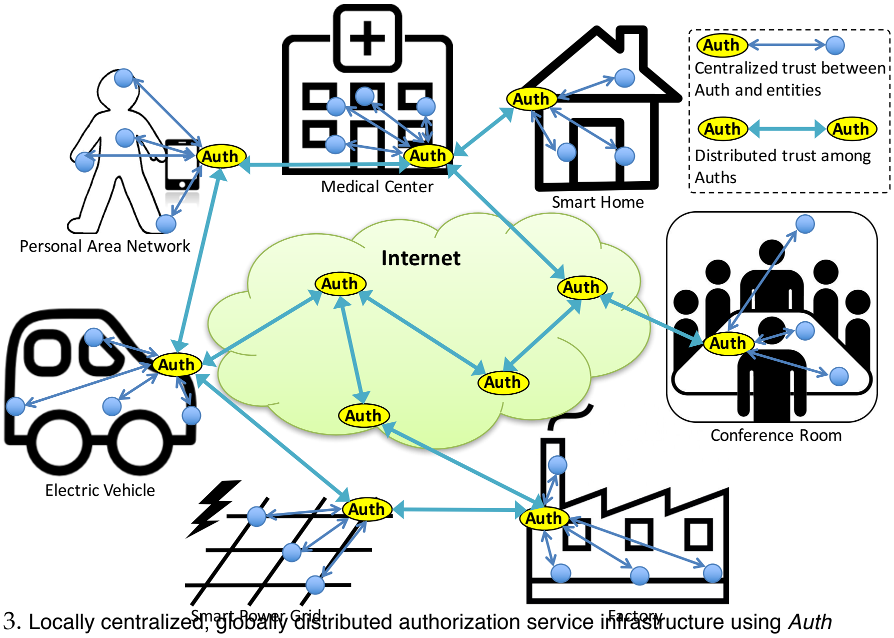


Figure 3. Locally centralized, globally distributed authorization service infrastructure using *Auth*

Locally Centralized, Globally Distributed Authentication and Authorization

We introduce a network architecture shown in Figure 3 using local authentication/authorization entities that we call *Auth* [13] to be deployed on edge devices of any kind. Auth's open-source software implementation in Java is available on Github (https://github.com/iotauth). Auth provides authorization services for locally registered entities (IoT devices), while managing trust relationships with other Auths globally. We call this a *locally centralized* and *globally distributed* infrastructure. Auth is now part of a toolkit for constructing an authorization infrastructure called the Secure Swarm Toolkit [14]. In [14], we have shown through formal analysis that Auth and the *Secure Swarm Toolkit* satisfy fundamental security requirements.

Locally Centralized

As noted in the previous section, the locality of edge computing has advantages over globally centralized cloud computing in terms of security and privacy. Auth keeps credentials of registered devices locally because we expect local domain experts can better manage Auth and registered devices. We assume that the network granularity of local IoT devices may vary depending on the network's nature; it can be a personal area network, a vehicle, or a building, for example. The Secure Swarm Toolkit also includes software building blocks to program IoT applications accessing Auth and IoT services. These software building blocks encapsulate cryptographic keys and operations to help local system designers with only moderate knowledge in security.

Auth stores credentials and access policies of its locally registered entities in its database. The authorization process is achieved by distributing *session keys* which are cryptographic keys valid only for specific access activities. To serve different contexts composed of heterogeneous IoT devices, Auth provides a variety of security alternatives. For example, it can support multiple underlying communication protocols, including TCP and UDP over WiFi, and BLE. Auth allows resource-constrained devices to use longer-term, cached session keys with less power-hungry cryptography. Auth also supports secure one-to-many communication such as broadcasting or publish-subscribe by distributing the same session keys to more than two entities. The effect of different security configurations on energy consumption is demonstrated in our Secure Swarm Toolkit paper [14].

Globally Distributed

The trust relationship between Auths is globally distributed. The current implementation of Auth uses HTTPS for communication between Auths, based on certificates. These certificates are managed in a way that is similar to the web of trust in OpenPGP, trusting other Auths for signing certificates. When an entity needs to access other entities registered with another Auth, then the two Auths collaborate for authorization of their entities, leveraging the distributed trust. With this, we expect to achieve better scalability. A more detailed analysis of Auth's scalability is discussed in [14]. Unlike certificates in PKI, Auths do not need to have a domain name or fixed network address, allowing edge devices without fixed network addresses to run Auth.

Distributed trust can make the entire system more resilient, limiting the impact of attacks even when an Auth gets compromised. The globally distributed architecture also provides ways to protect the internal network from the Internet, particularly by firewalling or physically disconnecting the external connection. This is possible because the local authorization service does not depend on an external authority. Even when an Auth becomes unavailable due to an attack or a failure, the resulting effect should be limited to the local authorization service. For further resilience, Auth can back up the authorization information of its registered entities to other trusted Auths and let the entities migrate to the trusted Auths for authorization in case of the Auth's failure.

Remaining Challenges and Future Work

There are still challenges that need to be addressed for further secure authorization infrastructure. Auth is fully automated once entity registration is completed. During registration, Auth and the entity to be registered set up credentials, security configurations, and access control policies. In many existing security solutions, this initialization process is quite costly. However, to cope with scalability and dynamically added and removed IoT devices, there should be an automated or semi-automated registration process. One possibility is to exploit physical proximity, where the ability to place a device within a few centimeters of an edge computing server establishes a trust relationship. Another remaining challenge is dealing with authorization for mobile devices. We envision this can be done in a similar way as the current cellular network, which deals with cellular handoff (changing cell towers as a mobile phone moves).

To enhance availability of Auth under a denial-of-service attack and avoid being a single point of failure, a distributed implementation of Auth will be required. To provide further security guarantees, we can use Auth as a point of intrusion detection, since it can see access-related activities of local devices. Auth also can serve as a software attestation center to guarantee that the IoT device is running a legitimate program not tampered with by adversaries. Further studies need to be carried out on usability of the Secure Swarm Toolkit's software building blocks for accessing Auth and IoT services.

Conclusions

Security schemes solely based on centralized trust do not take advantage of emerging edge-computing devices and may face problems of a single point of failure. Fully distributed solutions may not be practical for the IoT due to the overhead on individual IoT devices, especially resource-constrained devices. We envision the authentication and authorization infrastructure for the IoT to be locally centralized and globally distributed, which is achievable by local authorization entities based on globally distributed trust among these authorization entities.

Acknowledgment

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

[1] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the cyber attack on the Ukrainian power grid," *SANS Industrial Control Systems*, 2016.

[2] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos network authentication service (V5)," RFC 4120, IETF, Jul. 2005.

[3] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[4] L. Tung, "Mozilla to China's WoSign: We'll kill Firefox trust in you after mis-issued GitHub certs," ZDNet, Sep. 2016. [Online]. Available: http://www.zdnet.com/article/mozilla-to-chinas-wosign-well-kill-firefox-trust-in-you-after-mis-issued-github-certs/

[5] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, "OpenPGP message format," RFC 4880, IETF, Nov. 2007.

[6] G. Hurlburt, "Might the Blockchain Outlive Bitcoin?" IT Professional, vol. 18, no. 2, pp. 12–16, Mar. 2016.

[7] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 4, pp. 500–528, Nov. 2006.

[8] A. Pal, "Internet of Things: Making the Hype a Reality," *IT Professional*, vol. 17, no. 3, pp. 2–4, May 2015.

[9] "Cisco global cloud index: Forecast and methodology, 2015–2020," White Paper, Cisco Public, Tech. Rep., 2016.

[10] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-Physical Systems Security – A Survey," *arXiv:1701.04525[cs]*, Jan. 2017.
[11] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop*. New York, NY, USA: ACM, 2012, pp. 13–16.
[12] P. G. Lopez et al., "Edge-centric Computing: Vision and Challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, Sep. 2015.
[13] H. Kim, A. Wasicek, B. Mehne, and E. A. Lee, "A Secure Network Architecture for the Internet of Things Based on Local Authorization Entities," in *Proceedings of the IEEE 4th International Conference on Future Internet of Things and Cloud*, Aug. 2016, pp. 114–122.
[14] H. Kim, E. Kang, E. A. Lee, and D. Broman, "A toolkit for construction of authorization service infrastructure for the internet of things," in *Proceedings of the 2nd ACM/IEEE International Conference on Internet-of-Things Design and Implementation*, Apr. 2017, pp. 147–158.

Hokeun Kim is a Ph.D. candidate in the department of Electrical Engineering and Computer Sciences (EECS) at University of California, Berkeley. He is a researcher in the Ptolemy project at UC Berkeley. His research interests include system-level security for the Internet of Things (IoT), computer architecture for real-time embedded systems, and modeling and simulation of cyber-physical systems. He received his B.S. (2010) in Computer Science Engineering, and M.S. (2012) in EECS from Seoul National University. He was a research associate at HP Labs in Palo Alto, CA. He was a software engineer at ESTsoft Corp. and YoungWoo CnI Inc. in Seoul, Korea. Contact him at hokeunkim@eecs.berkeley.edu.

Edward A. Lee is the Robert S. Pepper Distinguished Professor in EECS at the University of California at Berkeley, where he has been on the faculty since 1986. He is the author of several books and more than 300 papers and has delivered more than 170 keynote and other invited talks at venues worldwide. Lee's research focuses on cyber-physical systems, which integrate physical dynamics with software and networks. His focus is on the use of deterministic models as a central part of the engineering toolkit for such systems. He is the director of the nine-university TerraSwarm Research Center, a director of iCyPhy, the Berkeley Industrial Cyber-Physical Systems Research Center, and the director of the Berkeley Ptolemy project. From 2005-2008, he served as chair of the EE Division and then chair of the EECS Department at UC Berkeley. He led the development of several influential open-source software packages, notably Ptolemy and its spinoffs. From 1979 to 1982 he was a member of technical staff at Bell Labs in Holmdel, New Jersey. He is a co-founder of BDTI, Inc. and has consulted for a number of other companies. He is a Fellow of the IEEE, was an NSF Presidential Young Investigator, won the 1997 Frederick Emmons Terman Award for Engineering Education, and received the 2016 Outstanding Technical Achievement and Leadership Award from the IEEE Technical Committee on Real-Time Systems (TCRTS). Contact him at eal@eecs.berkeley.edu.