Proceedings of the 51^{st} Hawaii International Conference on System Sciences | 2018

On Software Standards and Solutions for a Trusted Internet of Things

David P. Maher
Intertrust Technologies
dpm@intertrust.com

Abstract

We discuss a high-level model for software applications and services that can support a minimal set of human-centric trust management capabilities. We outline the unique set of challenges we must address if we are to attain a level of trust that will be required for a robust deployment of an IoT. We discuss the role of standards and how we can maximize the effectiveness of standards and device and service certification. We suggest a set of solutions for trust management that can support the unique security, safety, and privacy requirements of a robust IoT. Prominent among these solutions is the use of an older approach for access control, viz. the reference monitor, and blockchain technologies that can record trust and policy graphs and trust-related attributes for IoT devices and supporting services. An open, but governed trust blockchain can serve as a universal trusted oracle.

1. Introduction

More than any technology that has ever emerged so far, the Internet of Things (IoT) represents an evolution of computing that challenges our ability to manage the safety and security of our environment. As technologies are developed and introduced into the world, we rarely understand the full implications of their use, or even how dominant they may be come. This was true of the mobile communications industry [1], and the computing industry itself [2]. In the past we usually have had time to introduce standardized security solutions to deal with new threats as they emerge, though it has often been painful. The software industry has a long history of successfully supporting standards as computing has evolved [3]. However, we are now faced with the fact that computing technology is evolving faster than our ability to contend with the threats we can see clearly, much less the threats that will only emerge after new technologies are fully ensconced. The fact that IoT computing is ubiquitous, pervasive, and is now invading the innards of (what were once) the simplest of commonplace things, means that the number of stakeholders in any standard has increased in ways that make establishing consensus much more challenging. It may be more prudent to advocate a mixture of flexible standard approaches and cooperative strategies that address the most imposing requirements. This paper nominates some of those. In the past computing standards were typically designed in response to a desire for interoperability among multiple implementations of protocols or software components. Now, it is more important that software standards for IoT additionally address the need for humans to be able to contend with the complexity that IoT thrusts upon us, especially where it pertains to the ability of people to trust and manage the safety and security of IoT devices as well as the impact of potential accidental use or nefarious abuse of those devices on our lives.

Some previous works [4-7] address the need for new IoT trust models in line with what we propose, specifically recognizing the need for a more generally effective access control strategy and the usefulness of social graphs to accommodate less rigid policies.

We will continue the discussion as follows:

- Enumerate the drivers for a new informal trust model for the IoT
- Outline the components of a distributed trust model for the IoT that use trust graphs and policy graphs
- Describe solutions based on Blockchain technologies that indicate how this model can work
- Describe how to introduce minimal standards and cooperative strategies for establishing trust in an evolving IoT

2. Drivers for a new IoT Trust Model

We believe that any trust model for IoT needs to accommodate the main drivers of its evolution in the context of distributed computing. We enumerate them here:

1. Ubiquity and Scale
2. Hyper-connectivity

URI: http://hdl.handle.net/10125/50599
ISBN: 978-0-9981331-1-9

3. Cyber-physical integration
4. New interaction models

2.1 Ubiquity and scale

Numerous projections have been made of the expected growth of the number of Internet connected devices [8]. As the numbers progress through the tens of billions (and the often-quoted trillions of sensors), it becomes obvious that new strategies are needed to be able to manage and personalize all of these devices. People will own or be asked to interact with and /or manage hundreds of IoT devices and in industrial settings, often more. We need to find ways to tame the complexity that could overwhelm us if every device or software designer chooses their own unique interaction model.

Never before has the amount of data about individual people and their behavior been produced on this scale. New strategies for the governance and secure management of these many devices and the vast amounts of data they produce will be required to ensure that people are not overwhelmed by the IoT's complexity and scale.

2.2 Hyper-connectivity

We need to recognize that IoT devices are not just connected to the Internet, but rather they are connected to one another in myriad ways. This has profound implications for trust and security models. Individuals who may bring a device into their home or need to manage them in an enterprise may have little idea of the number of different network connections a device can make, and the number of different protocols they may use. Additionally, network connectivity is constantly changing as individuals with their personal area networks move throughout their homes, other people's homes, offices, and public places. As automobiles, with their burgeoning networks encounter the networks of the highway and urban infrastructure, and other automobiles, connectivity changes rapidly using many modalities, and using many different network layers and protocols.

This has profound implications for both trust and security models. It means we cannot rely on network security in order to protect ourselves. The weak link principle will overwhelm us. There will be no reasonable way to tell what device is connected to which network. Worse, as we need to worry about contamination, we won't know what device *has been* connected to what network, and what applications running on a device may have interacted with what applications on a network that may have been invaded by hostile entities. This all implies that devices and their applications will need to increasingly be responsible for self-protection. It is the only strategy that scales and deals with the issues of hyper-connectivity.

While device and application self-defense will be increasingly necessary, that does not mean that devices cannot seek help from trusted sources. We will discuss that strategy later.

2.3 Cyber-physical integration

This is perhaps the most obvious driver of a trust model for IoT, yet it is not clear that it is fully appreciated. Resources in IoT devices do not just include processors and memory, but rather they include components that can easily be turned into weapons and even weapons of mass destruction when used in scale. Currently we see ransomware being applied to office computers and controllers for remote locks, but of course there is no reason this cannot go much further as the cyber world integrates with critical infrastructure controlling dams, electrical grids, etc. This point has been made repeatedly, and the threat has been clearly instantiated for years (see the stories of the Stuxnet virus [9], where there were successful attacks on devices that were not even directly network connected). Yet, we see deployments of IoT devices where little thought has been given regarding potential misuse and disruption. That may be all well and good for the moment, but as we have seen when technologies become fully ensconced in our culture and we become increasingly dependent on them, the emergence of misuse and abuse inevitably ensues.

2.4 New modes of interaction

As computing evolved, we saw new modes of interaction introduced, and sometimes it was not until the next stage of evolution that we saw those new modes become commonplace. For example, when networked computing arrived, mobile code was introduced, but when networked computing evolved to mobile computing, mobile code (in the form of apps) became more dominant and that refined model of software distribution and maintenance is even being retrofitted to networked desktop computing. As an aside, this is a good example of where security design kept up (at least somewhat) with the evolution of technology. The dominant mobile operating systems (IoS sand Android) took the mobile code security problem head-on. In fact, their approach can evolve to a good approach for IoT security, as discussed later.

The IoT, as we see it currently evolving, includes many relatively new modes of interaction that we should model, as they are critical to the trust model. IoT human interaction modes have far greater variety as humans interact with things through both physical and virtual interfaces. Machine to machine interactions while not entirely new are introduced in a more commonplace way, and devices are linked together to form more complex composite devices where events on one device trigger events on other devices (think the IFTTT service [10], but in an IoT context where devices expose services used by other devices). Furthermore, IoT devices will increasingly come equipped with virtual cloud images and cloud services that extend both the capabilities of those devices as well as their interaction models. Cloud services provide or amplify the "intelligent" capabilities of the devices. We see this as another class of IoT device interaction.

The IoT trust model thus needs to address security, safety, and privacy aspects of all three distinguished classes of interaction: 1) human interaction, 2) machine-to-machine (M2M) interaction, and 3) cloud services interaction.

2.5 Trust model drivers and "security by design"

"Security by design", "safety and security first" and other slogans and aphorisms are of course shorthand descriptors that are not well-defined. However, if we place them in a trust model context, we can begin to develop a process that makes these sentiments actionable. Recognizing the drivers, we can begin to more comprehensively list threats and hazards and prioritize them. In fact, that might be the first aspect of a IoT device security standard. The device manufacturer will need to show how the software on the device provides safety, security, and privacy given considerations of scale, connectivity, and interaction modes.

We also must take these sentiments seriously. As we have outlined above, IoT is an evolution of distributed computing where the scale is enormous, the misuse can involve virtually everything, and the pathways for spreading mischief and nefarious behavior are many.

The discussion below is meant to allow us to focus on IoT device and system design that focuses on security, safety, and privacy first.

3. Standard components of an IoT trust model

Trust in our context means reliance. Here we ask what do we need to rely on in order to enjoy safety, security, and privacy when we introduce an IoT device into our environment. A standard approach to establishing these attributes is to completely list the device's ***resources***. Here are classes of resources that we want to see:

- Device controls (anything that can make the device do something)
- State information
- Sensor information
- Computing resources (processors, memory, embedded programs)

Once we have these enumerated, we ask who can have access to them, and how. In the case of a hyper-connected IoT device the assumption must be that everyone has access to all resources, unless there are governance mechanisms for the resources. The challenge is immediately clear: How to make all of this comprehensible to the end user or person responsible for deployment and maintenance of the device? How do we make it easy for someone to rely on the governance mechanisms? This introduces the next class of the trust model components, namely ***trusted attributes***. Some entity can evaluate and test the device and determine that the resources are protected, or that they do not pose a hazard, or any number of assertions that can relieve a device user or manager of responsibility or worry. We will need to standardize on both the substance and nomenclature for these attributes in ways that ordinary people can understand, and we will need to allow policy to determine whether to trust some entity that makes such assertions. In order to properly deal with the complexity thrust upon us by IoT, standardizing on trusted attributes will be essential.

Access to resources on a device will not always be static, and so part of the trust model will necessarily be means for ***delegation***. If I have an electronic lock, I might establish sole control over its state, but I may want to give selected others a permission. How permissions are established and enforced is part of a delegation model that needs to be explicitly described, and again should be made understandable to ordinary people. This can be very tricky however, since delegation mechanisms that are truly explicit and don't rely on faulty assumptions of trust transitivity are difficult. The implications of delegation need to be accommodated. For example, it may not always be apparent that giving a youngster access to your home automation system also gives them access to safety related mechanisms that they may not understand.

The previous paragraph serves to introduce us to another part of an effective trust model, namely ***performance aids*** that can help us understand the consequences of action, give us guidance, and tame complexity. We cannot rely on (i.e. trust) effective governance that is neither understandable to an average user or overly complex. Trusted performance aids (often in the form of web services) will be an essential part of an IoT trust model. They will help with scale and complexity, especially when we use cryptographic key management mechanisms to implement a permissions model for resource governance across many devices.

In the course of delegation as well as other tasks associated with IoT management, we will need to identify things and the entities we want to trust with access to their resources. Thus, ***identity management*** will be an essential high-level component of a trust model. This is an area that we know has been problematic in the past. It will require a more intuitive system for identifying people and other entities as well as their attributes. Secure approaches to this using hierarchical X.509 and SAML certs have already proven inadequate. Below, we describe an approach that has a greater chance of dealing with the scale and complexity of the IoT.

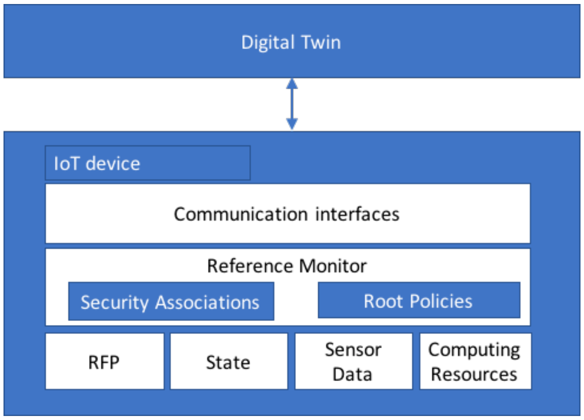
Composite devices need to be part of the trust model. These can be arrays of physically separate but similar devices, or they can be heterogeneous assemblies of devices whose union is defined by software, often running on the cloud. Such composites will become increasingly commonplace in order to tame the complexity of device management. We will need to recognize the fact that while IoT devices are most often designed and configured by manufacturers, composite devices will more often be defined and configured by end-users who may not be subject to the same regulation and subject their creations to the same testing and certification schemes as device manufacturers. This will be a challenge for any IoT trust model.

4. Direction toward some solutions

We cannot possibly describe here adequate solutions to the challenges posed by a comprehensive IoT trust model. Indeed, we only gave a very high level outline of a trust model above. However, we do want to point to some relevant approaches to the issues identified above. Some approaches are directly derived from prior research and experience in computer security and others are more speculative and will require extensive, forward-looking research and development. We have seen that our challenges and requirements are practically overwhelming, yet we are attempting to find simpler approaches to address them.

4.1 The reference monitor concept

The trust model discussion above began with the ability to enumerate resources and then define means for governing access to them. The concept of a Reference Monitor [11-12]] was first designed in the 1970s to perform this governance task. One can think of a reference monitor as a firewall between the “outside world” and the resources provided by a device. It is typically implemented as a core (or kernel) process that checks each command or request against a list of ***security associations*** (see below) for permissions to take an action or access some resource. The idea was used in the Windows NT kernel (and later editions), but reference monitors have gone out of fashion due to the fact that implementation becomes highly difficult when device interaction models are complex. Nonetheless, we imagine that at least a very simple reference monitor should be a part of any IoT device. In fact, the simpler the better, and we will need to make sure that device interaction models are compact. We fundamentally believe that in large scale hyper-connected deployments of cyber-physical devices we cannot afford to support arbitrary multi-tasking, multi-user interaction at the end-points. We generally do not need to in the IoT. Let’s consider a simple model of an IoT device in the following figure:



The distinguished resources in the device include a remote front panel RFP (allowing remote control of the devices functions), device state, sensor data, and general computing resources. The reference monitor intercepts every request to access these resources through every communication interface. The requests are authenticated using a message authentication code referencing an authentication key that is part of a security association record. The key is typically a symmetric cryptographic key shared between the device and the entity making the request. The security association will also include a set of permissions for the requesting entity. If the message is authenticated, the requesting entity's identity will also be confirmed and the permissions together with the device's root policy will determine access to the requested resource. In IoT, we make the assumption that general purpose computing resources will rarely be referenced except by a small number of entities for maintenance purpose. The RFP, state, and sensors will often be involved in P2P entity interaction with a limited class of entities, and the security association and policy data bases will be referenced by an even more restricted set of entities. Under this device interaction model, the use of a reference monitor seems quite reasonable, and we can make the direct to device interaction model even simpler by using the "digital twin" concept [13] whereby many interactions with the IoT device can be performed via an interaction with a virtual digital copy stored in the cloud or at a service point accessible to both the cloud and the device (gateway). The twin can then serve as a distinguished entity with a distinguished security association.

Now a number of issues come to mind: how do we establish this explicit security association, and isn't this overkill for simple devices? We want to address these questions before continuing further.

A ***security association*** is also a well-worn concept in computer security. As mentioned above, an IoT device can maintain a list of entities that may have permissions to access the resources of a device. This will be a table of entries where each entry consists of an identity of a remote entity, a symmetric cryptographic key shared by the device and the remote entity, a list of permissions the remote entity has regarding the device's resources, and some other info to help thwart replay attacks. When a command comes through an RFP interface, the reference monitor will check the message authentication code provided by the remote entity. This is a code that can only be properly constructed using the shared secret key. Unless I have arranged for an entity to have a security association with this device, no matter how much that entity may have invaded my other network devices, or even has access to the LAN, this device will not be directly accessible by that entity. Now the reader may observe that the device could still be attacked if a peer with a proper security association has been commandeered. However, the damage may be limited by the scope of the permissions granted to that peer in the security association data base, and of course we presume that peers are likewise protected by similar security monitors.

We advocate that a reference monitor be a *standard, foundational aspect* of all IoT devices, and that the governance of device resources be traced to a robust reference monitor implementation. This can be done while keeping simple things simple. However, the skeptic will rightfully observe that we have a dependency on cryptographic key management, which even in simple systems is hard to properly implement. In fact, systems secured using cryptography are often successfully attacked through the key management system and not by breaking the crypto. In addition, permissions management is even more complex, with transitive trust issues limiting the effectiveness of straightforward solutions. So, we next discuss how security associations and key management systems can be implemented in a scalable way using cloud-based performance aids which are part of our trust model. Then we discuss more general trust management solutions that can aid us in overall IoT device management.

4.2 Security Associations and IoT key management

Much of the complexity in the management of security associations derives from the complexity of security association management protocols. This starts with bootstrapping trust and root key management. ***Cloud-based trusted performance aids*** can help us simplify key management from the point of view of the end user and also the device designer. The IoT trust model must include robust processes for cloud service interaction as we mentioned above (see the discussion on digital twin). This is a distinguished kind of M2M interaction distinctly different from the M2M interaction characterized by devices acting in concert as composite devices. The trust relationship is quite different, and often involves trust roots embodied by asymmetric keys whose secret parts can be managed by cloud services that can afford the resources to properly protect them. We expect that humans can interact through a helpful UI with web services that can construct virtual images of security association tables used by device reference monitors, again using the digital twin concept.

Devices will need to be deployed with root keys that can be used to authenticate these web-based digital twins and or other trusted performance aids. Other than the relatively rare use of an authentication key for device renewability, heavy duty cryptographic protocols can be avoided in most devices, certainly the devices whose IoT interfaces only expose an RFP, state and sensor data, and a secure update protocol. It is beyond the scope of this paper, but we have designed scalable systems with high grades of security that place little burden on the device implementation. There are a number of considerations not discussed here, but we must say that the trust modeling is actually not that simple and requires assurances that require restrictions that are not always easy to accommodate. An example involves protection of root keys. In cases where we can severely constrain access to all computing resources in a device, and physical invasion of the innards of a device is not part of the attack model, protecting root keys in IoT devices is not difficult. However, in other cases we must employ more sophisticated and robust software self-defense methods, and hardware security methods. We mention, as an aside, that hardware security when available, can seem to be a simple solution but it is rarely adequate by itself mostly because the interface to a specialized secure subsystem needs to be carefully implemented and protected. Software security and careful design cannot be abdicated.

For many applications, IoT key management, security association management, and delegation of authority will require much more sophisticated trust management. We will need to have mechanisms that identify entities that make trusted statements instructing devices or advising people to rely on various assertions. While we claim that a simple concept of a reference monitor, properly implemented can solve our problems. We have just deferred the problem to the age-old issue of who CAN we trust for what purposes? The network security approach to this involves X.509 identity certificates and associated SAML certificates for trusted assertions [14]. Given the trust model drivers discussed above, we cannot rely on this technology which even in the hierarchical network security world is hard to maintain. With the kinds of IoT interaction models we already see emerging, hierarchical trust management will not work and does not scale well. We do, however, see a solution, discussed in the next section.

4.3 Assertion Blockchains and their use in a trust management system for IoT

There are numerous proposals for the use of Blockchains in IoT applications [15-17]. These generally relate to their use for workflows, "smart contracts," and immutable transaction ledgers for peer-to-peer IoT interactions, and applications such as supply chain management. But, here, we want to use blockchains for very basic assertions in trust management.

When introducing the use of blockchains in basic distributed computing applications, we hasten to disassociate the approach from blockchains designed to be used for Internet currencies or distributed ledgers designed for other kinds of value exchange. We use the term ***assertion blockchain*** to denote an application using a combination of Merkle hash trees and hash chains in combination with an appropriately chosen Byzantine Consensus Protocol [18] to robustly and immutably record assertions. In this case, the assertions will typically correspond to edges in a ***trust graph*** or ***policy graph***, where a trust graph consists of identity nodes, and an edge between nodes A and B is labeled with a set of conditions. An edge connecting A and B with those conditions means that when those conditions are satisfied, A trusts B, or sometimes A delegates trust to B.

A policy graph is used in a decision process, and an edge relates a set of conditions to a set of permissible actions that can then determine conditions that relate to another set of permissible actions.

Governance of IoT resources can be administered with the aid of an assertion blockchain, where various assertions are referenced iteratively. An IoT assertion blockchain will have permissions that are determined within the blockchain itself. However, it will also incorporate blockchain write permissions for certain root assertion types. Here is an example in a traditional context that illustrates this:

I may get an email from someone whose email address is abc@xyz.com with an attachment that includes a number of statements that say something like (PK_- indicates a public key):

"PK_A at time/date said PK_B ia validator for email from abc@xyz.com"

"PK_C at time/date said PK_A is a validator for email from the domain xyz.com"

"Pk_D at time/date said PK_C is a validator for all assertions made through the domain xyz.com

My email client can then use a policy graph with policy statements that 1) guide me to use the blockchain to validate all of the statements above, and 2) validate each step in a policy graph up to compliance with a root policy. This would be in lieu of using X.509 cert chains. The policy graph may require me to check whether any of the statements have been controverted by later statements (amounting to revocation).

Trust graphs can be embedded into a block chain as can policy graphs, that determine under what conditions I can trust an assertion, however the blockchain embedded policy graph will eventually lead my client to evaluate a locally embedded root policy with statements that are not embedded in the blockchain. It is this root policy that needs to be protected from alteration by other means, such as locally managed software integrity protection that is also part of the trust model.

There are a number of considerations to be made for an open but governable assertion blockchain that can function, practically, as a universal oracle that can help us determine what to believe. One such consideration is confidentiality. We believe that a publicly accessible blockchain can be used for storing authenticators of confidential statements. This can be done by using hashes of public keys and hashes of the assertions, and supplying the full keys and assertion documents out of band, and under private governance, but in a way that authentication of provenance is still preserved. We will need to take care that correlations don't make things easy to track. We are currently working on a number of approaches for that.

Currently deployed systems that use blockchains like Bitcoin to immutably record assertions do not scale for use for IoT applications, where we need ultra- low cost, low latency, and large capacity in terms of transactions per second. The example above cited a familiar trust management task that we believe can be implemented more flexibly and effectively using blockchain embedded trust graphs and policy graphs. However, the absolute need for a new approach is shown by the following IoT example.

4.4 Example: Trusting sensors

Suppose that we want to forensically validate a photograph made with a digital camera, and sent anonymously to a news organization. Can we trust the image the photograph represents? A policy graph could guide us through a number of steps each one of which would require trust validation. Some assertions that will need to be checked are relatively static, while others will be part of a high-volume stream of assertion blockchain entries. This is because a trust decision would require an assertion from the raw photographic sensor in the device, and perhaps an assertion from the GPS device. Assertions from each of the software modules both within the original device containing the photographic sensor and from postprocessor modules that may lie outside of it. In cases where we might expect even a small probability that the authenticity and provenance of the sensor recording might be questioned, we could record these assertions in the block chain. We summarize just some of the assertions that might be required to verify chain of handling and control in the table below.

Entity	Blockchain recording
Photo Sensor	Sensor credentials, calibrations
Raw output	Hash of main sensor output file, other readings (time, GPS…)
Device SW	Software credentials
Mobile Device	Device credentials
Output Photo	Hash of main sensor file, process IDs for transformations from Device SW,
Workstation Software	Software credentials
Final Photo	Hash of Photo with post-processing software IDs

The specifics may vary, depending on the way processes are composed within trusted modules.

There are billions of photographic sensors in existence already, deployed by hundreds if not thousands of different device manufacturers, and while even in the future checking the authenticity and provenance of a photograph may not be required very often, we don't know when and for what photograph we may need validation. Thus, we will need to record trust assertions from many sources: individual sensors, devices that house those sensors, manufacturers of the devices, originators of software post-processing apps, etc.

This example involves the authentication of just one type of sensor, however there are many more types that will become increasingly relevant as we rely on sensor information for all kinds of decisions, including real-time decisions involved with the governance and operation of critical infrastructure. Contributions to a blockchain embedded trust graph, policy graph, in combination with provenance assertions for recordings and recording transformations for an increasing number of relevant sensors will need scale that no system has currently achieved. Permission-less blockchains that rely on byzantine consensus protocols that are based on proof of work or proof of stake will not scale. A blockchain approach that uses consensus approaches for writing to the blockchain that also includes contextual permissions may well do the trick.

4.5 Assertion Blockchains with contextual write permissions

Although verification of the right to enter assertions into the blockchain can be done by referencing rights that have been asserted previously using the Blockchain, expediency and prioritization of assertions to be included and which will achieve consensus will require protocols that recognize the right to enter a new block by entities dependent on context. In currency transaction blockchains, miners can prioritize transactions even choosing according to transaction fee bidding. In addition, the protocol permits a new block only every ten minutes, and block size is fixed. These restrictions and approaches do not scale properly for many IoT assertion blockchain applications. Thus, a governance model for an assertion blockchain is required that allows flexible policy to determine who and when an entity has priority to write a block. The policy graph for governance will also be embeddable in the blockchain. A robust and well-defined governance model, and standards and rules for maintaining a policy graph for write permissions and consensus regarding the next block to be written and when, will be required. While an explicit governance model means the assertion blockchain is not completely open, it will negate the need for the massive overhead entailed by proof of work approaches and the artificial valuations required by proof of stake. We will require standards for governance, and certification of policy and the contextual permissions themselves.

4.6 Other supportive cloud services

Those who are owners of or are responsible for IoT devices can benefit from a number of cloud-based services that we need to accommodate in the interaction part of the trust model, and may benefit from some standardization. The first class of these we will call ***Virtualization and Visualization*** services. Such services will allow virtual images of devices to appear in graphical depictions made available through cloud services. As a device is deployed it can appear in one or more depictions that provide its state, and allow a user to interact with the device through a richer UI. One of the benefits of this approach is that the web service can also simulate for the user the consequences of any configuration including how changes may affect safety, security, and privacy regarding a specific device that might be the subject of focus, but also other devices that might interact with that device. These services can also allow graphically aided configuration of composite devices, and simulate how these composite devices will work. Again, this will be tremendously useful for end-users who need to be informed of the safety, security, and privacy consequences of interactions among IoT devices. These simulations will be more robust and discernable if IoT device interactions are standardized to some extent.

Since the virtualization and visualization services may do more than simulate possible interactions, rather they will be privy to actual configurations of devices, and change those configurations, we need to accommodate these services in the trust model, and there will need to be a well-understood and ideally standardized method of interaction between users and the V&V services and between those services and the actual devices.

Another class of web service that will be helpful and which will have a big impact on the trust model will be ***data analytics services*** for data collected from device sensors, but also from event streams generated by IoT devices. These services can be granted permissions to collect a stream directly from the devices, or they could be given permissions to collect data from a web cache that collects data directly. In the latter case, a web service could get information from derived data, allowing perhaps only relevant data to be accessed. In any case the trust model here could become fairly complicated given the variety of policies that the web service may follow regarding the disposition of data, and their claims of ownership. We have seen examples of data collection where a manufacturer monitors internal sensors of industrial devices, yet will not share that information with the end user or device owner. One can see that use of IoT data for forensic reasons (what caused a failure) and for predictive maintenance will be valuable. Data ownership issues will need to be worked out, but from our point of view, we want to at least make data flow and data access explicit.

One class of web service featuring data analytics that can be very supportive involves the analysis of event data to find security anomalies that could point to imminent attacks. This is a sensitive area, and an example where sharing data could benefit the common interest, even if it might reveal some information that might be considered proprietary. We believe such services will become fairly commonplace, as IoT devices will be vulnerable to systemic attacks and services that detect anomalous behavior will be useful for defending against such attacks. The trust model here will need to recognize the benefits and privileged access of these services, and the services will need to provide information regarding how data is protected, especially when some analytical approaches may involve comingling of different entity's data.

5. More on standards and governance

As mentioned above, it is harder to agree on standards when so many stakeholders are involved, and technology is moving fast. But, considering the end-user point of view as in [19], we realize that in order to contend with the scale and complexity of IoT, we will need to standardize on certain aspects of user interaction, nomenclature and other areas where lack of understanding could lead to safety and security hazards. We need to make choices about what to standardize and while in the software field we often are more concerned about interoperability, in this new world of IoT where software design can create physical hazards and lack of understanding by users can result in serious safety and security faults, standards will need to be motivated by user experience, and the need to limit complexity brought about by lack of standardization. While this is not completely new to the software world, we are going to need to deal with these issues more systematically. Consider standards for automotive operator UI. Some things are not standardized, like the location of light switches, but the brake pedal can be found in the same place on every car. There are higher level standards that deal with complexity of the automotive UI. See for example the overview published by the ITU-T focus group on driver distraction [20].

We have been using a trust model to organize our thoughts and form the basis of reasoning about safety, security, and privacy. This led us far beyond a focus on devices themselves, and helped us consider their entire interactive context. We will review the components described above and discuss implications for standardization.

Resources – we will need standards that encourage IoT device manufacturers to enumerate all accessible resources and demonstrate how access is protected using a reference monitor approach. We believe it will be helpful for resource nomenclature to be standardized so that users are less likely to be confused and overwhelmed when they need to manage IoT devices from different sources.

A **reference monitor** should be a standard part of all IoT devices, and simply implemented in simple devices. Cloud services can help maintain the requisite security associations in more complex devices.

Another standard aspect of IoT devices should be a documented **interaction model** that can be used when the device is evaluated to determine safety, security, and privacy properties. This model should show how resources are accessible by default, and how the reference monitor (or some interface to it) and root and dynamic policies affect interaction. This model should cover user, M2M, and cloud service interactions. We believe that interactions with other devices and services that collect data from IoT devices need to be explicitly highlighted with explanations of what data is collected, and what the service may do with it. We believe that the interaction model can be simplified using device virtualization (digital twin) approaches.

Trusted attributes are labels that are applied to devices by trusted third parties. Nomenclature should be standardized, and standard ways of authenticating attribution should allow machine resident policies that reference those attributes to be properly executed.

We have recommended that a **universal standard assertion blockchain** be used for immutably recording trusted attributes, indexed in a way that will allow them to be revoked or amended.

Identity management schemes will need standardized formats for machine reference. An assertion blockchain should be used to record and authenticate identity relationships including relationships with public keys.

Delegation schemes will need some aspect of standardization in order to avoid user confusion. Specifically, we will need standards that help ensure that delegation consequences are explicit.

Performance aids, usually in the form of web services, that help expose delegation consequences and problems with composite device configurations, will need standard modalities for warnings and notifications. Standard policies for persistently providing certain types of warnings (say, related to safety or cyberattack) will need to be considered.

6. Summary and related work

We have analyzed the problems relating to IoT trust, safety, privacy, and security, and argued that traditional computer security approaches will not be effective. We have identified what we believe to be the most powerful and useful solutions based on an old and largely abandoned computer security approach (reference monitor), together with a system architecture approach making the reference monitor concept viable (categorization of resources and device virtualization), and a universal approach to trust management based on blockchains. Our goal has been to find a set of prescriptions for IoT trust that can be standardized without being overly constraining, and which can be used for IoT devices of all types.

We have argued that the key to effective governance of IoT device resources is to simplify interaction models to the point that the reference monitor approach is feasible for large distributed systems, and while we point out that device self-defense is the most scalable strategy, we have discussed the use of cloud services and device virtualization (digital twins) that can provide performance aids for the maintenance of security associations and key management, relieving devices of the responsibility for the complex tasks those systems can involve, while simplifying the interaction model.

Important related work is being done applying social graph concepts for trust management of IoT devices [21], especially for devices that operate more autonomously, making greater demands on our ability to reason about trust and automate its management.

Additional related work is aimed at monitoring IoT device behaviors [22] in order to strengthen system security by identifying behavioral anomalies in device interactions. These technologies can be used in conjunction with dynamic policy management that can be administered using some of the infrastructure we advocate.

To make trust management more effective, we introduced the possibility of using assertion blockchains to replace the use of the hierarchical X.509 certification schemes. Such an approach can be "open, but with governance." That is, we can imagine a blockchain with embedded trust and policy graphs that anyone can use and rely on for virtually any kind of assertion relating to IoT devices, users, manufacturers and attributes, yet has trust anchors that can be used for root policies in applications. This blockchain will also feature governance policies that can ensure the scalability, proper functioning, and relatively smooth evolution of the blockchain.

We wish to accelerate the debate on how much to standardize, and how strictly. The drivers of IoT we mentioned above make it necessary for open, public oversight of IoT system evolution to emerge.

The concept of a universally accessible trust management oriented assertion blockchain that is subject to explicit governance will be controversial and a challenge to properly implement, but it seems so potentially comprehensive and adaptable that we will continue to better define and implement this approach.

7. References

[1] Angel Lozano, www.dtic.upf.edu/~alozano/innovation/

[2]www.pcworld.com/article/155984/worst_tech_predictions.html

[3] en.wikipedia.org/wiki/Open_standard

[4] S. Sicari, A. Rizzardi, L.A. Grieco, A. Coen-Porisini, "Security,privacy and trust in Internet of Things: The road ahead", Computer Networks, Vol. 76, pp. 146-164, Jan. 2015.

[5] Bernabe, J. B., Ramos, J. L. H., & Gómez-Skarmeta, A. F. (2016). TACIoT: multidimensional trust-aware access control system for the Internet of Things. Soft Comput., 20(5), 1763-1779.

[6] X.Xu, N.Bessis, J.Cao "An Autonomic Agent Trust Model for IoT systems", Procedia Computer Science 21 (2013) 107 – 113.

[7] F.Bao, I-R Chen, "Dynamic trust management for internet of things applications", Proceedings of the 2012 international workshop on Self-aware internet of things Pages 1-6.

[8] M.O'Neill, "Insecurity by Design: Today's IoT Device Security Problem", Engineering 2 (2016) 48–49.

[9] K. Zetter, *Countdown to Zero Day*, Crown publishers, NY, NY, 2014

[10] en.wikipedia.org/wiki/IFTTT

[11] T. Jaeger, *Encyclopedia of Cryptography and Security (2nd Ed.)*, Springer, NY, NY, 2011, pp. 1038-1040.

[12] J. Anderson, 'Computer Security Technology Planning Study', ESD-TR-73-51, US Air Force Electronic Systems Division(1973).Section4.1.1 http://csrc.nist.gov/publications/history/ande72.pdf

[13] M.Grieves, J.Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems", in *Transdisciplinary Perspectives on Complex Systems*, F-J Kahlen, S. Flumerfelt, A. Alves, Editors, Springer International Publishing, Switzerland, 2017

[14] W.B. Bradley, W.B., D.P. Maher, "The NEMO P2P service orchestration framework." In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on* (pp. 10-pp). IEEE.

[15] K.Christidis, M.Devetsikiotis "Blockchains and Smart Contracts for the Internet of Things", IEEE Access, 2016, Vol. 4, Pages: 2292 - 2303

[16] H.M. Kim, M. Laskowski,, Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance (August 23, 2016). Available at SSRN: https://ssrn.com/abstract=2828369 or http://dx.doi.org/10.2139/ssrn.2828369

[17] Y. Zhang, J. Wen, "The IoT electric business model: Using blockchain technology for the internet of things", Peer-to-Peer Networking and Applications July 2017, Volume 10, Issue 4, pp 983–994.

[18] G. Bracha, Asynchronous Byzantine Agreement Protocols, *Information and Computation* vol. 75,2, Nov. 1987; Elsevier, Amsterdam, NL. Pp. 130-143

[19] D. Maher, https://www.oreilly.com/learning/a-human-centric-trust-model-for-the-internet-of-things

[20] ITU-T Focus group on Driver Distraction, Report on User Interface Requirements for Automotive Applications, www.itu.int/en/ITUT/focusgroups/distraction/Documents/deliverables/ , 2013.

[21] M.Nitti, R.Girau, L.Atzori, "Trustworthiness Management in the Social Internet of Things", IEEE Transactions on Knowledge and Data Engineering Year: 2014, Volume: 26, Issue: 5 Pages: 1253 – 1266

[22] T.R.Fuller, G.E.Deane, "IoT applications in an adaptive intelligent system with responsive anomaly detection", IEEE 2016 Future Technologies Conference(FTC), Page(s):754 - 762