Handling the Internet of Things with Care

Peter Kirstein and Socrates Varakliotis
UCL Computer Science
Gower Street, London WC1E 6BT, UK
{P.Kirstein, S.Varakliotis}@cs.ucl.ac.uk

ABSTRACT

This paper describes how the use of Identifiers with an appropriate system of identifier *storage*, *registration* and identifier *resolution* can greatly extend the flexibility of a system dealing with IoT. The features of the CNRI Handle system are shown to match well the requirements of such a IoT system. In addition to strong technical advantages, the system allows the separation of mechanisms for the management of IoT objects and processes from those of their network addresses – though the two are tied together by the identifier attributes. This in turn eases both the orchestration of independent applications and devices, and the independent usage of the same network end-point from different stakeholders – while the different applications can be managed separately.

We have validated our thesis, by applying the system to a smart office environment, and shown how the properties of the IoT devices can be stored securely in a Handle repository including the characteristics of the device, network addresses and security attributes.

Keywords

Internet of Things; IPv6; Digital Object Architecture; Handle System; IoT security; DTLS; CoAP.

1. INTRODUCTION

The Internet, particularly when combined with the World Wide Web, are clearly the most successful mechanisms for linking the wide variety of computing devices together, with a plethora of applications. There have been many attempts to analyse how and why it has been so successful, and how this can be replicated with the Internet of Things (IoT). While it is an over-simplification, one can assign the success to a few key aspects:

1. There is a single Internet and network layer protocol, below which a variety of network technologies can be accommodated. At the network layer there is a well-defined addressing and packet structure. A routing structure is linked to the addresses.

2. There is a clear Web upper interface, with a uniform naming structure, to which applications can be linked in a uniform way. While the range of application services are becoming more sophisticated, the underlying mechanisms are simple and straight-forward.

3. There are specific bodies with open membership, but tight control, of the mechanisms being standardised: the Internet Engineering Task Force (IETF) for the Internet, and the World Wide Web Consortium (WWWC) for the Web. These have defined the protocols in their own sphere and their interface to each other.

4. There is a Name/Address Resolution System that is replicable, scalable to large numbers, and with tight control on the underlying mechanisms.

For the IoT, it is widely accepted that the Internet will still be used for wide-area communication, and that one would like to replicate as many as possible of the above features in IoT itself. In IoT, one cannot assume that all the underlying network technologies link directly to the Internet layer; it may require more of a description of the characteristics of the device network to meet (1). Web services are widely used in IoT, allowing (2). When some of the recent protocols designed for constrained devices are used, the conditions for (3) can be satisfied.

The main networks all obey a set of protocols defined by the IETF – including those optimised for constrained devices. The number of such devices is becoming so large, and private addressing so cumbersome, that it is moving to IPv6. Key requirements in the Internet are a scalable, distributed set of repositories mapping human recognisable names to IP addresses, and a security infrastructure that keeps edge devices, servers, gateways and their communications reasonably secure. The first is achieved by the DNS; the second is achieved by way of protocols like DNSSEC [1], TLS [2], DTLS [3], SSH [4], IPsec [5], to name a few, and other key distribution mechanisms. Increasingly, information and actuation services are being provided in a standard way by Web Services, which obey the HTTP or CoAP [6] protocols.

The Internet of Things (IoT), should be regarded as an extension of the Internet, which will still be connected via Internet technology, but for a very considerable time there will be a much broader set of edge devices. These edge

IoT Ecosystems 2014, December 02-04, London, Great Britain

DOI 10.4108/icst.mobiquitous.2014.257957

devices will obey Internet protocols to the wide area, but may use a number of domain-specific protocols nearer to the devices. Thus a mechanism that is broader than the DNS, and can also give a description of the device as well as its location is required. There is still a vital need for security, but many of the devices either do not support much security intrinsically (and yet will persist) and/or are not powerful enough to support some of the more sophisticated of the Internet security protocols.

One *Name Resolution system*, with an accompanying set of repositories and registries is the Handle System [7] (hereafter referred to as 'HS' or just 'Handle'). Handle is an embodiment of a complete Digital Object Architecture. Its Name Resolution Service has many of the features of the DNS of scalability and distribution, but also facilitates a much more powerful syntax that describes devices or processes and allows the incorporation of security features. The upcoming version (v8.0) of this system not only supports more services than the DNS, but also links in well with web services. In the IoT6 project [8] we have shown how the use of this system can provide powerful features to aid device descriptions and security features in IoT.

There have been attempts to embrace multiple device technologies through IPv6 address mappings into the structure of their network addresses, e.g. GLoWBAL [9]. This has several disadvantages. Since the IP address can be read openly, there may be too much revealed about the nature of the device composition. Secondly, allowing application-domain-specific use of IPv6 addresses will complicate the management of network address space. Use of a system like Handle overcomes these drawbacks. Its syntax is much richer than the DNS, allowing the storage and resolution of an arbitrary set of attributes. While Handle is as scalable and distributed as the DNS, its security features are such that one needs authorisation to read the attributes of an identifier. Since one attribute can be an IPv6 address, there can be a direct link between identifier and IPv6 address. Finally, since an end-point may be represented by several identifiers, and its network termination have several IPv6 addresses, different IoT application domains may manage their applications completely independently while using the same edge devices.

There are many ways in which a system like Handle can assist in IoT applications. Because the Digital Objects can be processes, it can ease the composition of systems that include chains of processes. Because of the requirement for strong authorisation to access attributes, it can be used to store attributes that can be accessed only by authorised entities. In IoT, many of the edge devices are too constrained to perform complex authorisation operations; thus Handle can be used to provide proxy security operations.

In a paper like this, only a small set of the possible uses can be considered in detail. Here we discuss the use of Handle in this last role. We have demonstrated in IoT6 how it can be used to provide secured operations, in an environment where the individual components are too constrained to provide the range of security services required.

In Section 2 we describe the relevant features of the Handle Identity Resolution System. In Section 3, we discuss how it can be used to assist security in a particular environment. In Section 4 we describe some aspects of IoT governance in the context of our approach. Section 5 analyses the phases of IoT systems and Section 6 gives a detailed overview of our validation effort. Finally, some conclusions are drawn in Section 7.

2. THE RELEVANT FUNCTIONALITY OF HANDLE

Handle has a number of important and relevant characteristics, with the details described in [7]. A few of them, used in this paper, are provided below.

- It has a managed and globally distributed structure like the DNS, and is globally accessible.
- Its delegated servers (Local Handle Services – LHS) are owned and operated by arbitrary institutions or organisations.
- The Identifier consists of a Prefix and a Suffix. In the Global Handle Service (GHS), the prefix is a globally unique integer assigned to LHS owner and LHS servers can be run standalone.
- The Identifier Suffix can be an arbitrary string. Therefore it can be a URI, URN or IP address if so desired.
- The Handle attributes are arbitrary pairs of *(Type, Value)*. Their Values can be arbitrary strings, allowing them to represent encrypted contents or any other form of identifier.
- The Handle protocol allows for owners of delegated Handle services to further delegate services. However, security risks and policy reasons may render this neither desirable nor permitted at global scale.
- It has been designed to be scalable by allowing its servers to be distributed, and data on one server split onto two if the load becomes excessive.
- It stores Handles as hierarchic Identifiers with secure access to attributes. The access must be authorised. It includes an infrastructure for signing, private/public key operations and integrity checks.
- The Handle system can provide shared caching within a local community; this may be either in a standalone cache server mode or as part of a general caching mechanism. Given a cached resolution result, subsequent queries of the same identifier may be provided from the cache without contacting any Handle service. This permits an application to perform repetitive functions securely while maintaining protection against replay attacks. Since this feature reduces the need to access the Handle system as frequently, it improves both aspects of performance and scalability – albeit with some loss of security.
- Mechanisms for restricting access to Identifier attributes to authorised users, based on a Public key or Secret key technology. The Public/Private key infrastructure can use full X.509 certificates.
- A mechanism for linking multiple network addresses with identifiers for the same physical object or process.
- Facilities for Handle subsystem access via IPv6 networks and the capability of attributes being IPv6 addresses.

- The next release, v8.0, allows programmatic access via a REST interface.

- Software libraries to manage Handles exist, mainly in Java (while some older C library and Python wrappers are less supported nowadays).

3. CONSIDERATIONS CONNECTED WITH SECURITY

We describe in this section how the Handle Service can play key role in providing the security mechanisms required in a IoT architecture. Key components of the security-enabled architecture we envisage are shown in Fig.1.

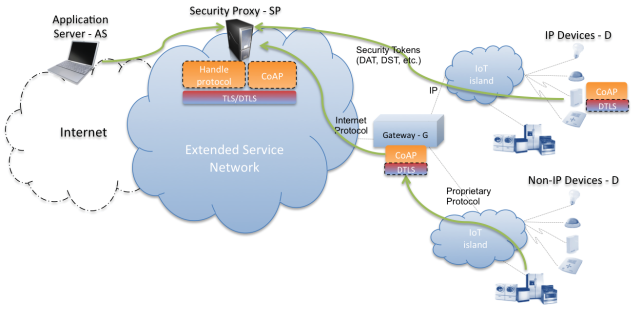

Figure 1: Configuration of IoT applications with security provisions.

Here an application server AS runs IoT applications which include running Sensing and Actuation as Services (SaAaS) on remote devices D. Sometimes the devices are connected directly to the Internet; sometimes they are connected via a gateway G which is itself connected to the Internet. We assume that to do any operation on D requires a security token DST, and to ensure the authenticity of any value returned by D requires an authentication token DAT. Hopefully the device D will be powerful enough to check itself that an operation requested by AS, has been furnished with a valid DST; it will also be able to append a DAT to any value it transmits. If this is not the case, these operations will have to be done in some security proxy SP. Unless the Device is also capable of encrypting and decrypting messages, it will be vulnerable to spoofing and information leakage between D and the nearest SP that can apply the encryption/decryption. For the rest of this paper we assume that the operations are performed on the device; the argument changes little if they are done in an intermediate SP, except that all operations between SP and D will be unprotected. We assume also that D may have constrained capability, so that it is important to reduce the complexity of operations performed by D to a minimum. By contrast, more complex or powerful operations can and should be performed in AS. The basic argument does not change if the more powerful operations are carried out in a different proxy server.

Some basic security operations for which an infrastructure is required are the following: *Confidentiality, Authentication, Authorisation, Integrity, Audit trails.* Confidentiality is always implemented by encrypting/decrypting information with a security key K. Message integrity is indicated by appending a Message Digest MD. The confidence one can have in the authorisation or authentication depends on the strength of the encryption and MD algorithms, and the security protocols used between AS and D. We assume that these algorithms, while as simple as possible, are adequate for the level of protection required. The most complex will probably turn out to be authentication and authorisation, since there will often be many situations to be considered. For example many different entities may be authorised to perform operations on a device. The minimal operations that must be performed in the device is to confirm that the requestor of the operation has a valid security token DST, and to append an authentication token DAT.

In order to keep these operations as simple as possible, we mandate that any message requesting an operation containing DST is so authorised, and any message received by AS containing DAT is authentic. To ensure the authenticity, confidentiality and integrity of messages between AS and D, they use the DTLS [3] protocol.

From Section 2, it is possible to restrict access to Handle attributes to authorised users, create and store security and authentication tokens, have attributes that are other Handles, and include IP addresses. If one is still concerned with unauthorised access to a Handle store, then individual attributes can be stored in encrypted form. In our approach there are unique Handles for both the AS and each device D. The DAT and DST are stored in Handle as attributes of the device. The Handle infrastructure, together with the algorithms in AS, are used to ensure that only authorised messages containing DST are sent to the device. Similarly, AS is authorised to retrieve DAT, and so can check the authenticity of data received from D. The operations between AS and Handle can be as well protected as desired; there is usually little constraint on the complexity of the operations, and data can often be cached rather than requested each time.

The simplest mechanism for providing *confidentiality* is to use *secret key symmetric encryption* between two parties. This key is a shared secret known to the two parties, but not to an outside attacker. A second application of the encryption operation will reveal the original message. Of course providing the *shared secret* of this key to both parties is a challenge; this challenge will be addressed with Handle, but only partially.

Authentication can be confirmed either by using secret key encryption or private/public keys. In public/private technology, one uses an encryption algorithm that is hard to encrypt but easy to decrypt. The private key is known only to the party being authenticated. Public/private key technology is more resource intensive. While it can be used between AS and Handle, most implementations for constrained devices use only shared secret keys. In our validation of Section 6, this is the version that is used there.

4. IOT GOVERNANCE

The subject of governance in IoT is a major area of concern, and has been the object of a whole working group in IoT-A [10] and in the IERC [11]. We treat here the problems of governance in the following particulars:

- Governance of IP-space
- Governance of Name Space
- Governance of Multi-application gateways
- Governance of Security Functions

- Management of Security

While we intend that our approach here would be more generally applicable, we do not advocate it as the only or even the best approach to the different problems raised. Indeed, the application domains of IoT are so diverse, that a single universal approach is very unlikely.

4.1 Governance of IP Space

There is huge international concern on all aspects of the Internet. Indeed, the Internet Governance Forum [12] meets frequently in large international gatherings just to consider this subject. One part of it is the management of IP-space. This used to be managed centrally from the US, but for the last twenty years this has become internationalised. There is a central Internet Registry IANA [13] that allocates large blocks of addresses to a number of regional registries e.g. RIPE [14] for Europe. This system has worked well for IPv4, and is now adopted for IPv6. We are not advocating any change to this system, however some of our activities could threaten this system if carried further than our approach.

With IoT, there are complex physical subsystems being managed. At first glance, one could use the full 128 bits of the IPv6 address space to reflect an application-specific structure of the parts of the IoT world. Such uses are impractical, because the top 64 bits are used for routing purposes in the Internet. The use of the lower 64 bits to represent structure is more defensible, but may still be considered problematic. It would raise the general question of having many more bodies concerned with the management of IP address space. This is a dangerous precedent which would cause endless squabbles in international fora, and may well interfere with the smooth running of the present system. Indeed, technically there would be nothing too problematic in application-specific organisations, e.g. the KNX forum [15] using a particular subset of the address space; indeed, this could not be prevented. However even this modest proposal would raise conflicts between different such bodies. Moreover, IP addresses are often stored in the DNS system. This system is open to outside inspection, thus the IP address might well reveal more information about the underlying physical system than is wise to make generally available. The alternative mentioned below would be superior and would avoid the above risks.

4.2 Governance of Name Space

It is vital to achieve economic scaling to allow a systematic derivation of network addresses and subsystem properties from models of the physical system. Most systems we envisage, be they smart buildings, transportation systems, patient populations or smart cities, have such models for the physical objects. In the preceding section, we have mentioned the proposal to use IPv6 addresses in this context and have deprecated it. An excellent alternative is to adopt a very flexible identifier system, and to associate attributes of the physical world algorithmically to such identifiers. Such a system can be much more flexible and open. If well-designed specifically for this purpose, can meet the IoT needs without the drawbacks of the use of IP space.

Important aspects of such a system would be the following:

- The system should be very extensible – conceptually able to extend to the huge numbers of digital objects envisaged in IoT, with the variety of attributes being considered
- Whereas all identified objects should be globally accessible if required, such access should be able to be completely cut off from the global system if that is desired
- There should be the capability of providing access control to ensure that only authorised users can traverse the identifier space
- There should be fine-grained access to the individual identifiers
- There should be subsets of the identifier space that could be completely managed by arbitrary bodies
- It should be possible to manage the identifier system in a way that would be acceptable to the potential stakeholders
- An identifier should be associated with an arbitrary number of attributes. These can be used to specify its nature or how it can be treated
- It should be possible to embrace other identifier systems

We do not pretend to have a unique solution to the needs of IoT in this regard. Indeed it is a major subject of study in many august bodies including the ITU [16]. However, the approach we adopted in the IoT6 project [8] of using the Handle System [7] from CNRI seems to satisfy most of the above. We have discussed in detail in [17,18] how the system is organised, its characteristics, and why it would at least meet the above needs. A Handle consists of an arbitrary number of $(Type, Value)$ attributes. Since the Type of the attributes can include authentication tokens, authorisation tokens and network addresses, it provides all the services that could be provided by the DNS – but in a more secure manner. There is even an international foundation [19] that governs the use of the system. Of course any universal approach to this problem would scrutinise the properties of candidate systems much more rigorously, and put in much more elaborate safeguards, constraints, and governance of the system itself. Incidentally, a model of how this might work has been shown in Handle in its almost universal use in the media industry and ISBN system. Its practical use in IoT is being demonstrated in its Chinese use for the retail and food production chain. Here local governance, security and extensibility are key features.

Handle has a provision for registering Types globally. This caters for two other types of governance. First one can set up bodies to determine which types are of common enough usage that they should be registered globally. Second, it is possible to register types which embrace a whole other Identifier System. An example is to define the Type *Object Identifier (OID)*; this could allow the whole of the OID space [20] to be used to describe the nature of devices in Handle. In practice, this would not be straightforward or useful, because the OID system has a different security model and method of constructing its identifiers; this subject is currently being studied in the ITU. When we refer to 'Handle' in the rest of this section it is to indicate only that a system

such as Handle could be used, with the proviso that Handle already has the requisite features assumed.

In [17] we show how the system can be used in a in a generic fashion in a specific domain, the 'Buildings Management' domain. We demonstrate a solution that implements a system which addresses the domain model, i.e. one could use it to implement other solutions for Buildings Management (as opposed to a system which addresses a specific Use Case of Management of X Buildings of Y organisation).

4.3 Governance of Multi-application Gateways

At first sight, it would seem that the governance of gateways is a straight-forward task. Some entity owns a gateway, and can therefore be held completely responsible for its management. This is not quite the case. A major reason why IoT may become so prevalent is that the physical world will be increasingly populated with devices that can sense their environment and be made to actuate processes. Often the same devices may require access from different application domains, with their own ecosystem. For example, a smart building may contain a HVAC. The subsystem will need to be set, of course, by a *buildings manager*. However, it may also need to be accessed by a *maintenance engineer* with quite different access requirements and incidentally normally different network address space. A door may need to have its capabilities managed by a *buildings supervisor*, its faults by a *maintenance engineer*, its access by a *user* with an RFID card, its local credentials by a *security manager* and its opening in an emergency by the *municipal fire department*. Some of these require quite different access to its properties, be authorised by different entities, and even different network addresses. In a smart city, a sensor may be used by the traffic lights, police surveillance, bus traffic analysis and street lighting. The individual sensors, or a gateway controlling a collection of them, may require complex and extensible sensing and information authorisation facilities. Not only may these be beyond the capability of a smaller device, but also their management in multi-application environment could become very complex. For this reason, we envisage that use of a system like Handle could have a very important place.

One property of IPv6 technology is that it is possible, and natural, for an IPv6-enabled device to have several such addresses. Thus each application domain could assign its own network address to the device valid in the domain specific to its activity. By defining its own Handle, it could be assigned the network address and other attributes that matter to that application domain. If many application domains require similar attributes of the device, these could be put into another Handle, which is itself an attribute of each of the applications. Access to the gateway may require dual keys, some specific to the owner of the gateway, and one specific to the owner of the application. There may well be further governance desired on specifying the attributes of certain classes of devices, certain classes of applications, and certain standards across applications. However such agreements would only ease the implementation and deployment of the technologies, they are not pre-requisites.

4.4 Management of Security Functions

We must distinguish between the *management* and the *governance* of security. In some cases the governance depends only on single-domain, local, decisions. Many, or even most, of the current deployments have this property. However there may well be Health and Safety or Public Interest considerations which modify this approach. For example, the sensors for car parking may be installed to enable the billing of car owners; however their access for street lighting and police security may be statutory requirements. It could well be that for police security a degree of camera movement is required, while for street lighting much less detail should be available. The Handle approach, with different Handles for the different domains, and several keys allowing access to different attributes, may be highly desirable. The governance may well require constraints on the attributes that are permitted in different application domains using the same sensors or gateways.

Security management has many facets, most beyond the scope of this paper. When identification of sensors is required, it is desirable that each such sensor has an intrinsic token for identification. This might be the equivalent of a manufacturer-supplied MAC address or its unique address in some proprietary building automation systems. The more difficult the token is to spoof, the more faith one can have in the system. For each operation possible on a digital device, there should be a security token that authorises the operation to be performed. It would be possible to have the device hold several security tokens with different privileges. This might require complex operations in remote devices. An alternative is to have the remote device have a single security token. However, each application may require its specific security, and would authenticate the validity of authorisation of an operation. If, and only if, the operation is indeed authorised would it be sent to the remote device with the specific security token required by the device.

The use of a system like Handle would be very helpful in this process. Both the application-specific tokens and the device-specific ones can be stored as attributes in Handle, and used in the authorisation phase. Having different Handles associated with different applications on the same device, and with the same operations on different devices give powerful aids to security management. Thus an application developer may or may not be authorised to put the security attributes into the Handle system; that may be reserved to a Security Administrator. An unauthorised person may not even be able to traverse the identifier space of Handle. Thus if the identifier space gives details of the architecture of a subsystem, these details could be hidden from unauthorised users. In the same way, the network manager may be authorised to put in the network address attributes from a model of the application. Here the purpose of the authorisation may be more to ensure that the translation is done correctly in view of the network characteristics. The fact that network attributes are associated with Handles may have nothing to do with who is authorised to access the Handle.

Fundamental to all the above is that individual entities are authorised by strong public/private key authorisation. This in its turn is predicated on a security infrastructure that is trusted to identify the entities and to furnish them with the appropriate public/private key pair, and to do the correct key renewal and revocation. These functions are fundamental to most of the current security environments. Many IoT applications are carried out over radio networks which are notoriously insecure. For this reason in the validation described in Section 6 and in [17], all transactions

between an application and a remote device are carried out using DTLS [3]. This is a datagram-based protocol standardised in the IETF for this class of transactions. Of course, how strong the security of DTLS is depends on the encryption algorithms used – which may be not as strong as one would like if the devices are processing-challenged.

5. THE PHASES OF IOT

In most IoT applications there are at least two phases: a *set-up phase* and an *operational phase*. Whenever there is a change in the application, configuration or authorisation, one may need to enter the set-up phase. However, to consider what is needed in that phase, it is desirable to consider first the operational phase, and then consider what pre-configuration that requires. It is difficult to generalise both phases, but we will describe them below where appropriate as typical operational and set-up phases for smart buildings applications (which is the validation example in Section 6.)

5.1 Operational Phase

Operations may be initiated by an application server (AS) or by an outside event. In an application-initiated operation, each operation is applied to a Handle of a digital object, which could be a process or device. This Identifier is the Handle ID (HID) of the device. Using the *application authorisation token* (AAT) of the application, the HID and the attribute of the operation, the application tries to obtain the *Device Security Token* (DST) of the operation. Assuming the AAT is authorised to access the Handle of the digital object, the authorisation infrastructure of the HS also checks that it is authorised to download the DST, and any other parameters required for the operation – e.g. its network address. The application then accesses the device with the DST. The device need verify only that the DST provided is a correct security token for that operation. The operation is then carried out. The operations between the application and the HS are carried out using the Handle protocol; those between the application and the device use secure datagrams with DTLS and CoAP for the Application Data Units. The application may be requesting data from the device. In this case the above operation will be followed by one from the device returning the data requested. This must be authenticated with a *Device Authentication Token* (DAT) authenticating the device. The DAT is a combination of a property of the device itself (MAC address or serial number) and of the network address; this is because the network address may be an indication of other devices associated with that one, and because the network address assignment may have occurred through some sort of auto-registration.

It may be that the operation is not being applied to an end-device, because it is a legacy component or requires a complex process. Then the access to the HS may return not only the DST and device network IPv6 address, but also a security token and identifier for an intermediate process. In that case the relevant parameters are sent to that intermediate process.

In a device-initiated operation, an outside event causes the device to send a message to the AS containing the DAT via DTLS. On receiving the message, AS associates the message with a HID. It then queries the Handle System (securely) to verify the authenticity of the message, and whether the sending entity had the authorisation to make the operation request. Thereafter the procedure is identical to that followed by the application-initiated operation.

5.2 Set-up Phase

In order for the operational phase actions to be carried out, there must be a set-up phase – which may indeed be entered many times.

We assume that there is a Master Management System (MMS), which can call many generic applications (A). These are stored in a database, with a Handle to the application stored. During the building installation phase, several databases are set up. One is the Model Configuration (MC) database. This is database of the building, based on some model of the building. For each room, exact devices are stated and an inventory is created. For each device, its exact description is given, including possibly authentication information built into the device. From this authentication information, and other such as the network IPv6 address, the Device Authentication Token (DAT) is constructed. It should be noted that some of the databases may still reference others. For example, the configuration database of the building may just state that each room has a door sensor and door actuator. A further database gives the actual model identifier and serial number of each component. The model identifier may actually be the Electronic Products Code [21] of the component. If there is a fault, and a component is replaced, the repairer would be obliged to put a new EPC into this database. During the model configuration phase, a Handle registration phase is entered which assigns a Handle to each room and to each device; this Handle may be algorithmically constructed in a manner similar to the GLoWBAL system [9]. There are two differences: First, the construction is of an Identifier, not a network address; thus it is not stored in an openly searchable DNS but only a Handle store that requires authorisation. Secondly, the Handle stores data about the device or room – possibly including authentication tokens. During this phase, one associates authorisation rights to each Handle. One may also associate processing Handles. For example, one may define a technology-dependent process to construct the data to be sent to a device; this can be defined in the Handle of the Device as of type *nextprocess*, which in turn is a complete process – including requiring an authorisation token to be activated. The Handle for each device will also include a *network address*. While this may again be defined algorithmically, this algorithm need take account only some network topology information – not device characteristics.

In constructing the application, one may use also the HS to steer the operations. For example, if the device is a *doorsensor*, then it may be associated with a *dooropener*. The Handle for the dooropener may be an attribute of the doorsensor. Another use is that the doorsensor may be associated with a Handle for the room, so that an event in the dooropener sensor may point to all the devices in the room.

Another aspect of this phase is that one can use the *group member* provision in Handle to define group operations. For example one could define all temperature gauges in a building (or on a floor) as belonging to a group. If one is using IPv6 for devices, then the group operation translates into an operation on a multicast group. One important aspect of this, and in which IPv6 is different from IPv4, is that the device may have several Handles with different properties and quite different Handle and network addresses. Thus,

for example, the municipal fire department may have the need, and authority, to open all doors in a building; the need for individual actuation requests would be obviated. A door opener could be provided independently with a Handle (and network address) in the address space defined by each of two different organisations, e.g. the municipal fire department and the building supervisor. This allows each to have complete freedom in their applications.

From the above, it is clear that the set-up phase will normally be key to IoT applications, and that more than one *role* may be needed. Thus in our examples, the *Buildings Supervisor* and the *Fire Department* must have separate authorisation to set up their different Handles. A *Buildings Contractor* may set up the Handles for the different devices in a building, but it is a *Security Manager* who decides who has access to which resources. An individual *Installation Engineer* may have authorisation to define the Handle for a particular device – but no other authorisation. The fine-grained control of Handle over authorisation is ideal for this.

6. VALIDATION: SECURITY OPERATIONS IN A SMART OFFICE WITH HANDLE

6.1 Analysis

Our work on this activity has not been a mere paper exercise. We have defined a set of operations in a smart office environment that are realistic, and show how Handle could be used to assist the IoT. We have implemented enough of the full scenario to illustrate all phases of the application. For lack of space, we describe only the part of the total scenario implemented that illustrates the interaction with Handle for security functions. The IoT6 project full Smart Office scenario, including the non-security-specific interactions as well as auxiliary operations unrelated to security, are described in detail in [8].

In the scenario for the smart office, a person wants to enter a room. If he is authorised to enter, the system will allow the door to be opened, will put on the lights and send a default setting to the HVAC. The system will monitor the office temperature; if it goes too high, it will change the HVAC settings. A presence sensor checks if there is anyone in the room. When this indicates that there is nobody present, it will shut down lights, etc., and return to the room entry condition.

In the operational phase all Handle operations are read-only. In the validation, we simulated the smart office scenario without using the real devices (HVAC, door, etc.), but using the full real Handle Internet and IoT infrastructures that we built. We simplified the granularity of the authorisation used, and the complexity of setting up the office equipment Handles. However all operations on devices were based on authorised operations, and all sensor readings on authenticated sensors.

In the Setup Phase we assume that the Handle system administrator has configured various *Building Administrators* with additional *create* and *delete* Handle permissions. The administrators would also have separate security tokens and possibly different authorisation levels. These administrators have also set the authorisation tokens for the building application to access Handle. Normally, these tasks are done prior to the operations phase, while the office building is being prepared and its spaces are allocated to various occupants and functions. The process involves a significant amount of data entry, and assignments of roles, settings, permissions, access cards, access rights, control of equipment, etc., following processes of the relevant departments (Estates and Facilities, Security Dept., HR and Personnel Administration, etc.). While the actual processes are outside the scope of this paper, the requirements of these processes are quite crucial, they require secure operation and have significant impact on the management of the IoT system.

6.2 Implementation

Based on the analysis of sub-section 6.1, we have actually carried out a complete implementation that is described in [17] of all the parts concerned with Handle. The different devices like sensors and door openers are represented by small motes [22] running Contiki, but we use a real RFID reader [23] to simulate the 'Request to Enter'. Each of our sensors have a mechanism to provide the authentication token DAT. Any attempt to actuate lights require the security token DST. There are mechanisms for auto-configuration of the sensors, and for generating the DAT based on a MAC equivalent on the sensor combined with the IPv6 address of the mote. All communication between the motes and the application are via CoAP/DTLS – using symmetric encryption.

The application runs on an AS, which is a PC running Linux. The communication between the AS and the motes is via 6LoWPAN. The application uses private/public algorithms when communicating with the Local Handle Server (LHS), which runs also on a Linux PC. The communication between the AS and the LHS uses the REST protocols of CoAP/DTLS which are supported on the LHS.

During the Operational Phase, when a data packet comes in from a mote, its Handle is found on the AS. Then its authenticity is checked with the DAT that is stored in the LHS. If found authentic, the Application carries out the relevant operation, which may include sending data to a Web-based management server. If an actuation is required, the DST is obtained from the LHS, and the data including the DST is sent via DTLS to a mote. The mote checks the validity of the DST, and if valid performs the operation. Repeated access to the LHS is largely avoided by obtaining the relevant parameter from a cache held in the user process accessing the LHS.

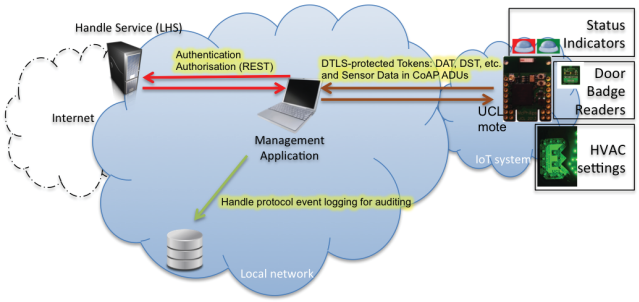

Figure 2: Implementation overview of Handle-aided secure IoT scenarios.

During the Operational Phase, the various accesses had required only read access. During a Set-up Phase, the different Handles required are set up in the LHS and the AS. For these there are different credentials allowing the creation

and update of the credentials on the LHS. During the Set-up Phase, some algorithms are run to generate the Handle Identifiers from the assumed room number and device. The Handles and their attributes are then set up with the appropriate credentials. Then various parameters like IPv6 address, *DAT* and *DST* are inserted. All these operations are carried out securely requiring authentication and authorisation at each stage using private/public key pair operations.

In the implementation carried out, neither the security policies nor the algorithmic generation of the identifiers were very sophisticated. This would be different, of course, in a serious deployment.

7. CONCLUSION

While a system like Handle can provide assistance to IoT in many areas, this paper has concentrated on the aids to security. The use of the Handle server has considerable advantages over DNS. While it shares the global reach and scalability of the DNS, it allows also the storage of the characteristics of end-devices, security attributes and network addresses, requiring authorisation to access these attributes. While algorithmic computation of IPv6 network addresses to reflect device characteristics has shown advantage in IoT deployment, application of the same technique to identifiers is even more powerful and flexible. It prevents the revelation of the properties to unauthorised users, and allows different stakeholders to manage the algorithms quite independently. Even when the remote devices have constrained resources, it allows proxy security servers to carry out complex authorisation and authentication operations which are beyond the capability of the devices by themselves. Finally, although the validation described is limited in the range of applications and devices considered, the security properties have been demonstrated fairly fully. Moreover, we have shown that all the above activities can be implemented in full compliance with existing Internet protocols.

Acknowledgment

We acknowledge the financial support of the European Union under the IoT6 project 288445, and that of CNRI in giving us early access to the Beta Release of the Handle System v8.0.

8. REFERENCES

[1] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose."DNS Security Introduction and Requirements", RFC 4033, March 2005.

[2] T. Dierks, E. Rescorla. "The Transport Layer Security (TLS) Protocol Version 1.2". RFC 5246, August 2008.

[3] E. Rescorla and N. Modadugu. "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[4] T. Ylonen and C. Lonvick. "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[5] S. Kent and R. Atkinson. "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[6] Zach Shelby, Klaus Hartke, and Carsten Bormann. "Constrained application protocol (CoAP)" (2013).

[7] Sam Sun, Larry Lannom, and Brian Boesch. "Handle system overview". RFC3650, November, (2003).

[8] IoT6 European Project. "Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability" (2012). `http://www.iot6.eu`

[9] A. J. Jara, M. A. Zamora and A. Skarmeta. "GLoWBAL IP: an adaptive and transparent IPv6 integration in the Internet of Things", Mobile Information Systems, IOS Press, ISSN: 1574-017x, (2012).

[10] Internet of Things Architecture, IoT-A, `www.iot-a.eu`

[11] European Research Cluster on the Internet of Things. `http://www.internet-of-things-research.eu`

[12] Internet Governance Forum, `www.intgovforum.org`

[13] Internet Assigned Number Authority, `www.iana.org`

[14] Registry for IP Europe, `www.ripe.net`

[15] KNX: The Worldwide standard for Home and Building Control, `http://www.knx.org/knx/what-is-knx`

[16] International Telecommunication Union. `www.itu.int`

[17] S. Varakliotis, P. Kirstein, G. Deiana. "The Use of Handle to Aid IoT Security". Submitted to the 2015 IEEE International Conference on Telecommunications (ICC 2015) – Internet of Things Symposium, June 2015, London, UK.

[18] S. Varakliotis, P. Kirstein, A. Jara, A. Skarmeta. "A process-based Internet of Things". 2014 IEEE World Forum on the Internet of Things (WF-IoT), pp.73–78.

[19] Digital Object Numbering Authority, `https://isocbg.files.wordpress.com/2011/09/dona-flyer2011.pdf`

[20] Object Identification. `www.oid-info.com`

[21] "Electronic Product Code". `http://en.wikipedia.org/wiki/Electronic_Product_Code`

[22] `http://www.baileynet.co.uk/Orisen/index.html`

[23] HF RFID Module SL030 3.3v – i2C. `http://skpang.co.uk/catalog/hf-rfid-module-sl030-33v-i2c-p-1065.html`