Securing Building Management Systems Using Named Data Networking

Wentao Shang, Qiuhan Ding, Alessandro Marianantoni, Jeff Burke, and Lixia Zhang

Abstract

Recently, building automation and management systems, BASs and BMSs, have shifted from using proprietary protocols and specialized hardware toward widespread adoption of IP-based open standard technologies. While the IP protocol suite improves software and hardware interoperability, practical large-scale BMS deployments face challenges, including the complexity of network addressing and other configuration, reliance on middleware for even relatively simple tasks, and a lack of security. In this article, we propose a data-centric BMS design that uses named data networking, one of the proposed information-centric networking architecture designs. Our sensor data acquisition system uses a hierarchical namespace for data, encryption keys, and access control lists, implements encryption-based access control, and provides a web browser-based data visualization interface that communicates in NDN. Our design has been deployed on a UCLA campus testbed that captures, archives, and visualizes data from industry standard sensors.

Research on building automation and management systems is expanding in both academia and industry, stimulated by a growing interest in the visions of Internet of Things and smart grid. While traditional building automation and management systems (BASs/BMSs) as well as industrial control systems (ICSs) have used proprietary protocols and specialized hardware, there is a growing shift toward IP-based networking to take advantage of lower equipment and infrastructure costs, and enable interoperability with information technology (IT) in the enterprise. The adoption of Internet protocols in BMSs can bring several benefits: Ethernet switches and IP routers are usually cheaper and more readily available than proprietary BMS hardware; and the TCP/IP protocol suite is a well defined open standard, enabling communication compatibility between different BMS solutions and integration with IT systems.

However, the use of TCP/IP for BMS also faces significant issues, such as complex network configuration to meet communication and security goals; the requirement of deploying middleware to bridge application namespaces to the IP addresses, port numbers, and other designations used to route and forward data; and the lack of appropriate packet-level security mechanisms for machine-to-machine communication, as we elaborate in more detail later in this article.

In this article, we introduce a new BMS design that uses named data networking (NDN), a proposed new Internet architecture, as its foundation for network communications. As part of the larger field of information-centric networking, NDN shifts the "thin waist" of the Internet architecture from IP's host-centric model to a data-centric model, with two important consequences. First, data is named by applications, and this name is used by the network directly to fetch the data. Second, each data is associated with a cryptographic key, which is used to secure data directly.

We believe that NDN holds great promise for machine-to-machine communication and industrial control systems like BMS, providing desired functions including:

- Routing and forwarding directly based on application-defined data names rather than numeric and human-unfriendly host addresses, which greatly simplifies network configuration and troubleshooting in a typical BMS environment with hundreds of thousands of smart sensor devices that all require network connectivity
- Inherent security support at the network layer through per-packet signature and optional encryption, instead of relying on higher-layer solutions (which can be more complex and less efficient) or physical isolation of the entire system (which can easily be broken by various means)
- Built-in request-response architecture at the network layer, which is well suited for sensor data access and authenticated control, rather than using application layer protocols such as HTTPS
- In-network caching and support for distributed persistent storage, which can reduce query loads on resource-constrained devices and protect them from distributed denial of service (DDoS) attacks that are becoming more and more severe on the current Internet

Our work focuses on how to securely publish sensor data to be used by BMS applications and provide an initial framework for access control. One major contribution of this article is a novel access control solution that takes a data-centric, encryption-based, and non-interactive approach enabled by the NDN architecture. In contrast, IP-based security primitives only

Wentao Shang, Jeff Burke, and Lixia Zhang are with the University of California at Los Angeles (UCLA).

Qiuhan Ding is with Tsinghua University.

Alessandro Marianantoni was with UCLA REMAP when this work was conducted.

0890-8044/14/$25.00 © 2014 IEEE

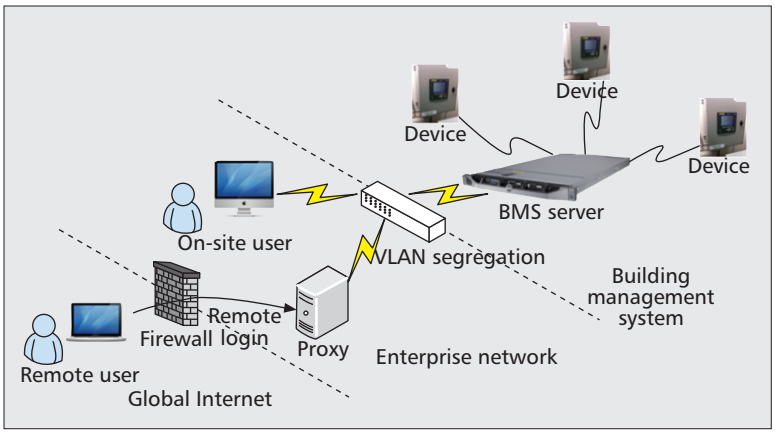


Figure 1. Partitions of a traditional BMS architecture: remote user, enterprise user, and BMS network.

protect transient communication channels, which are dynamically set up and torn down and cannot be shared across different users. Our data-centric solution presented in the article encrypts the data packet and securely distributes the encryption key to multiple users, making the solution more efficient and more scalable than the traditional IP security solutions.

With the help of the University of California at Los Angeles (UCLA) Facilities Management, we have deployed a monitoring system testbed that collects sensing data from industry-standard components, and implemented a prototype of our access control system on this testbed. Our solution employs identity-based access control to enforce trust relationships and uses encryption to protect against unauthorized reads. To inform this work, we use the goals and challenges identified in the BMS deployment by UCLA Facilities Management. The university's current BMS system has approximately 150,000 points of monitoring across the campus, and this number is likely to grow by a few times in the next five years. There is a great demand for a secure, easy-to-use solution to meet the campus building management requirements, and our work represents a first step toward fulfilling that goal.

Background

Named Data Networking

Named data networking is a newly proposed Internet architecture that replaces IP addresses with data names at the narrow waist of the "hourglass" Internet architecture. NDN defines two packet types, *Interest* and *Data*, that carry data requests and replies, respectively. Applications issue an Interest for a specific data name or prefix, which is consumed by the returned Data packet that has the longest match with the requested name. Every Data packet contains a signature that binds the data to the name, enabling data consumers to verify the authenticity of the received segment based on an application-specific trust framework. Routing scalability is achieved by using hierarchical names, and long-term storage is provided through the use of NDN *repositories*, network-connected persistent data stores. Repositories store named content for namespaces for which they are responsible, and answer Interests in those namespaces. Data packets in NDN are signed and immutable, and can thus be served from any nodes that wish to provide them. Named data networking is described in more detail in [1–3] and in other publications listed at http://named-data.net/publications/.

[1] *In this article we use building management sysems (BMSs) and building automation systems (BASs) interchangeably*

Traditional BMS Architecture

For our purposes, BMSs[1] cover the intersection of three critical sub-areas: industrial control systems (ICSs), including supervisory control and data acquisition (SCADA), and so-called smart grid [4], *enterprise networking*, and the Internet of Things (IoT) movement [5]. Enterprise BMSs are environments that bring critical infrastructure considerations of ICS together with IoT's visions for the everyday environment. The convergence of networking in ICSs with traditional IT has been described as a sea change by the National Institute of Standards and Technology (NIST) in their ICS security review [6].

BMSs are software/hardware systems that perform control, monitoring, and management of heating, ventilation, and air conditioning (HVAC), lighting, water, physical access and other building components. Data gathered by instrumentation in BMSs is used for performance evaluation, fault detection, billing, reporting, and many other purposes; in many cases, it is also used to provide closed loop control on a local and system-wide level. Our focus in this article is on secure distribution of sensor data that is a fundamental element of these systems, and corresponding access control for end users, applications, and devices.

BMSs have typically used the industry's own protocols. Examples include LonTalk [7] between devices, Modbus [8] and BacNet [9] for devices and gateways, and OPC [10] for inter-system communication. Many such protocols were first used on point-to-point serial links or fieldbuses, but in recent years have been adapted to run over Ethernet and IP. That is, instead of running directly above the physical layer, the Modbus/BACnet data frames are encapsulated into TCP or UDP packets and distributed over an IP network. In both the traditional and IP/Ethernet cases, access control is almost always achieved through physical or logical isolation (e.g., using VLANs, firewalls, and "air gaps"). Figure 1 illustrates an architecture overview of a typical BMS environment.

Motivation

The motivation for creating a BMS architecture using NDN has emerged from the significant challenges we have observed in IP-based solutions, especially as more and more elements of buildings are monitored, actuated, and interconnected.

Addressing and network configuration: Accessing even a single sensor's data from a networked application requires knowledge of addressing on many layers, such as VLAN ID, subnet, and IP address (of a gateway or device), port number, and device ID. The overall addressing schemes are typically brittle to change and have little semantic relationship to the system's function or the nature of the data. Managing a variety of devices and applications becomes especially time-consuming and error-prone on a enterprise-scale BMS installation, which, like UCLA, could have tens to hundreds of thousands of data acquisition points deployed across different buildings, each accessed by a variety of applications.

Middleware: Middleware is often used to mitigate addressing challenges and provide a mapping from descriptors that make sense to an application to lower-level network addresses. However, it adds a configuration burden and reduction in performance that can make it unsuitable for lower-level machine-to-machine (M2M) communication, without fully resolving the mismatch between network addressing and application logic.[2] For example, a typical BACnet system configuration process usually compiles a long list of mappings

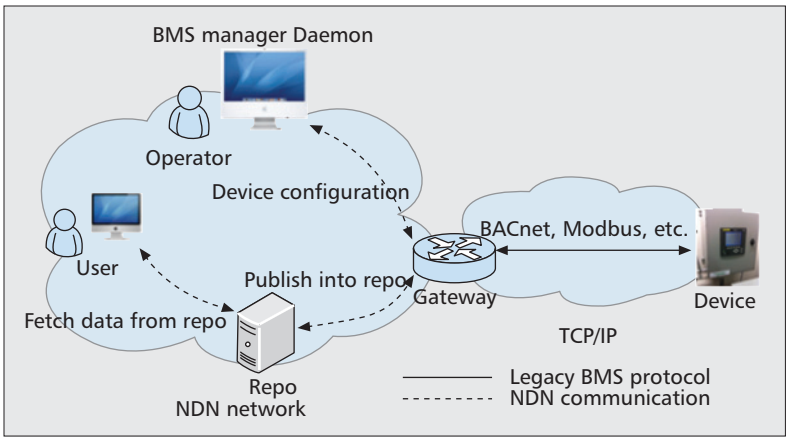


Figure 2. Major components in NDN-BMS, where there are potentially many users and gateways, and multiple devices per gateway.

between BACnet device object names and their IP addresses; inside the BACnet protocol, the system requires another indirection from human readable names to non-intuitive object IDs in order to fit into the protocol message format.

Security: Because most BMS protocols arose in industrial networks that were assumed to be physically or logically isolated, their intrinsic security is weak to non-existent.[3] However, BMSs are often part of critical infrastructure, and need robust security as they are increasingly connected to IT networks and the global Internet. Maintaining a secured IP channel among the sensor devices is costly, if not infeasible. For example, devices with low duty cycles may need to shut down and re-establish the secured connections all the time as they have to hibernate periodically to reduce power consumption.

Our goal is to create a data-centric framework that utilizes NDN to consistently address, scalably forward, and securely publish sensing data. This approach enables the connectivity demanded by modern distributed applications while limiting data access to a specific group of users (e.g., facility managers, building occupants, and authorized background applications).

In NDN-BMSs, names are consistent across applications and the lower network layer. Addressing is dealt with in one place in the architecture, and used by both applications and network delivery. This significantly reduces the complexity of network configuration compared to IP-based BMSs, where device addressing involves interoperation across multiple layers. Also, NDN-BMSs provides authentication and privacy through features intrinsic to the NDN data packets rather than relying on channel-based security. This allows us to create an encryption-based access control solution (described in the next section) that improves system scalability and mitigates DDOS attacks. We do not see similar simple and robust solutions emerging in the IP domain.

Major design objectives of NDN-BMSs include:

- A hierarchical namespace for sensor data, devices, and users that embodies intrinsic relationships in BMS applications and can be used directly for data delivery
- Bootstrapping and ongoing management of device configuration that is simpler, more scalable, and more robust than IP solutions
- Scalable user and privilege management to support enterprise-wide deployments
- Data authentication through cryptographic signatures to verify data provenance
- Data privacy through encryption-based access control without reliance on physical/logical network isolation

The first goal directly addresses the naming and addressing issue; the second and third goals involve higher-level system configuration, which is related to the challenges of addressing and middleware in BMS; the last two goals aim at the security challenges.

NDN-BMS Design

In this section, we describe the initial design of an NDN-based building management system that collects monitoring data from sensors deployed in UCLA campus buildings. Our initial deployment collects sensing data from existing industry-standard sensors and gateways that employ legacy BMS protocols. Consequently, our design includes gateway devices that read data (through an internal IP network) from sensor devices running legacy BMS protocols and publish the resulting data in an NDN network.

Figure 2 shows the architecture of NDN-BMS with three main entities: end users, sensor gateway (including the connected devices), and a privileged manager application controlled by human operators who handle out-of-band user authentication and privileged authorization. The manager application is responsible for gateway/device auto-configuration and management over the NDN network.

Gateways insert sensor data into NDN repositories (repos), which respond to user Interests for data in the sensor namespace. This indirection decouples data generation from user fetching, and provides benefits of automatic data archiving, non-interactive access control (described below), and protection from DOS/DDOS attack on gateways. In our initial implementation, a web interface (using the NDN-JS library [12]) fetches the data from the repo by issuing Interest packets with proper data prefixes.

Namespace Design

In an NDN system data names can be hierarchical and human-readable. We believe that by using human-readable names which are carefully mapped to application-specific structure in the data, the creation of complex distributed applications can be simplified. Because the network layer of NDN directly uses these names for data routing and forwarding, the namespace structure must be carefully designed so that data names are both meaningful to applications and compliant with the requirement of network routing scalability.

Figure 3 shows the NDN-BMS namespace used by the UCLA campus testbed. The root node represents the common prefix for the UCLA BMS namespace (/ndn/ucla.edu/bms). Under the root prefix, there are two sub-namespaces: */building* for data publishing and */user* for identity management. The chain of trust is constructed using an identity-based trust management policy. The BMS operator owns the BMS root key, which is trusted by all the entities on the BMS network. The root key signs the building namespace key, which further signs the identities (i.e., public keys) of the gateways deployed in each building. The gateways then use their keys to sign the data packets published under their namespaces. The root key also signs the user namespace key, which is used to sign the BMS user identities.

[2] *Nor does it typically keep track of configuration data changes in layers 2 and 3!*

[3] *See our brief discussion in [11] or the more extensive overview of ICS security by NIST [6].*

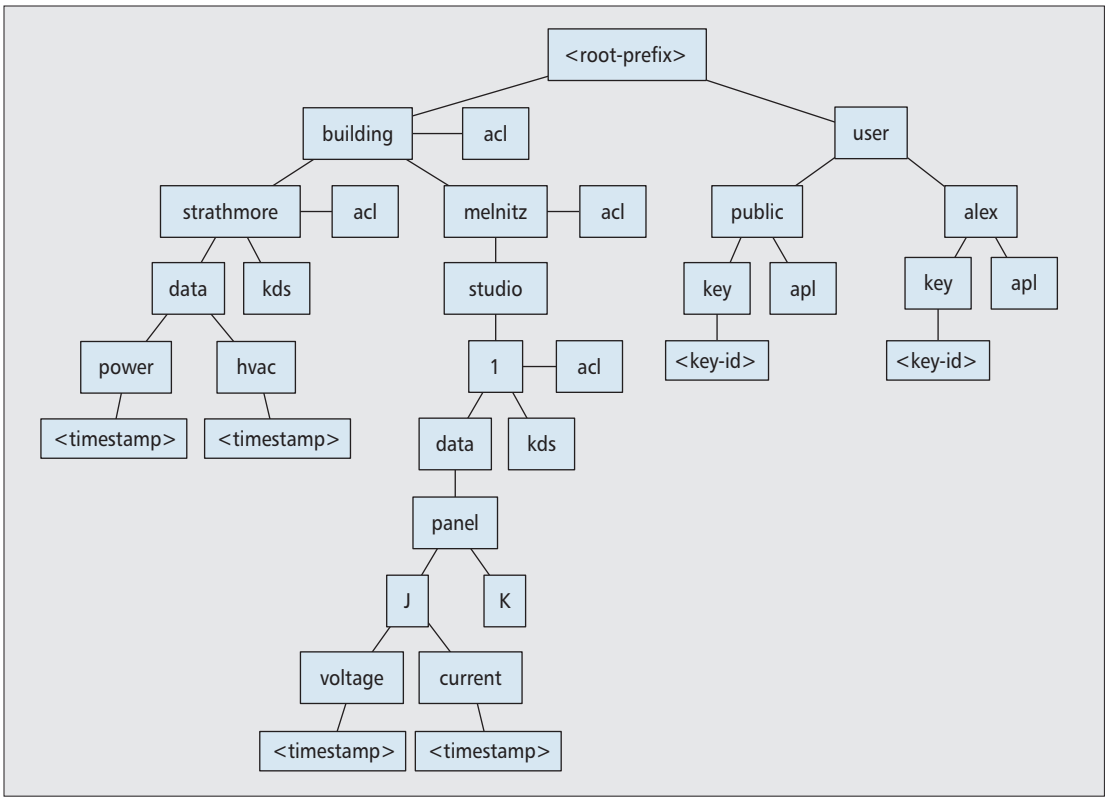


Figure 3. Example NDN-BMS namespace design for the prototype deployment at UCLA.

Data Structure Following Building Organization — In BMS protocols such as BACnet, it is common to name devices (and the data they generate) to reflect their physical location. We follow this convention for data naming in our NDN-BMS and construct the namespace based on the hierarchy in the building structure [13]. For example, the building (identified by its real-world name or label) may be (optionally) partitioned into *floors* (identified by the floor number) first, and then into *rooms* (identified by the room number). Devices may be grouped further by physical attachment (e.g., different panels) and/or functionality (power, voltage, temperature, etc.). Critically, the namespace schema for a given BMS can, to a large extent, remain at the discretion of those who install and manage the system.

Similarly, operators can choose the scope or granularity of responsibility for a given gateway. In the example of Fig. 3, we have one gateway that covers the entire Strathmore building, while another gateway is only responsible for a single studio in Melnitz Hall. Finer granularity reduces the load of a single gateway device but can increase management complexity. Note that it is straightforward to configure a gateway to publish in a certain part of the hierarchy — it just registers the appropriate prefix with its upstream hub, a process that is beyond the scope of this article but a straightforward part of the architecture, and then publishes data under that prefix.

Data Naming — The NDN name of a sensor data packets (e.g., /ndn/ucla.edu/bms/building/melnitz/studio/1/data/panel/J/voltage/<timestamp> in Fig. 3) indicates the physical location of the sensor (panel J inside studio 1, Melnitz Hall), the type of data (voltage), and the time when the data is acquired, expressed as a timestamp in the last component of the name. If the packet contains aggregated data (i.e., a series of sensor readings generated within a given time window), the timestamp reflects the generation time of the first data point in the series. The timestamp-based versioning allows BMS users to fetch sensor data within any time range once the data is archived in the repo.

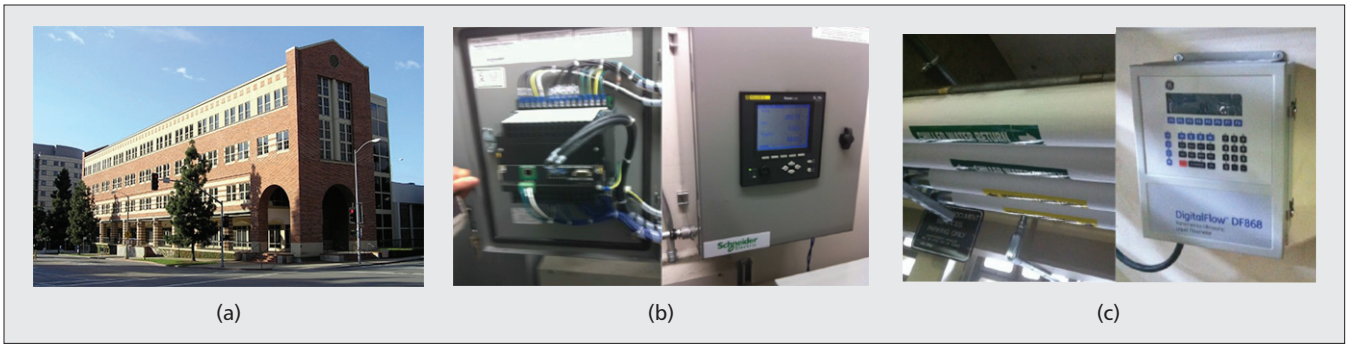
User Identity — The NDN-BMS uses public keys to uniquely identify users. Keys are published as NDN data and thus can take advantage of all NDN features for distribution. User keys are themselves signed by a trusted authority. An example user key name, /ndn/ucla.edu/bms/user/alex/key/<key-id>, is shown in Fig. 3. The last component of the name corresponds to the public key's SHA256 digest, which differentiates multiple public keys belonging to the same user.[4]

User Privilege Management

The NDN-BMS uses an identity-based access control scheme. Each gateway is configured with an access control list (ACL) specifying the identities of the users who should have access to its data. Each user also has an access privilege list (APL) indicating the data namespaces he/she can access, which is published into the NDN repo as well. When a user obtains access to some data namespaces in the BMS, the operator will publish the user's identity certificate into the repo and update relevant gateways' ACLs to grant access to the user. While this certificate publishing process can be automatized, the user's real-world identity must be checked out-of-band.

When a user's privileges for a namespace are changed, the manager daemon enumerates the namespaces in that user's APL and updates the ACLs for the affected gateways. The gateway ACL and user APL provide mappings along two different directions: from data namespaces to user identities and

[4] *We considered allowing identity names not within the BMS namespace, for example, enabling a user with a key under /ndn/ucla.edu/cs to access the data under /ndn/ucla.edu/bms/building/strathmore. However, requiring all BMS users to have identities inside the BMS namespace greatly simplifies privilege management and trust policy.*



(a) (b) (c)

Figure 4. UCLA Strathmore building monitoring system deployment: a) monitored building; b) electrical demand monitor; c) chilled water flow meter.

vice versa. This enables the BMS manager daemon to apply user privilege updates to relevant gateways without going traversing the entire BMS namespace. Note that ACLs and APLs can also be published as NDN data, enabling applications to use the same mechanisms to fetch sensor data, keys, and access control information.

We considered using capability-based access control, in which the gateway ACL specifies the list of capabilities required to access the sensor data, and users obtain the appropriate capability certificates for access. This approach simplifies ACL management in that the gateway ACL is not affected by changes in user privileges (e.g., users joining or leaving). However, it essentially converts the ACL management problem into a user certificate management problem, without reducing the complexity. In contrast, identity-based ACL is straightforward to implement and easy to debug.

The NDN-BMS employs a hierarchical ACL structure that follows the data (and thus the building) organization to describe users with different levels of privilege. The ACL associated with each prefix records the identities of the users who are authorized to access data under that prefix, as shown in Fig. 3. For example, the users listed in the ACL of the prefix /ndn/ucla.edu/bms/building/melnitz/studio/1 will have access to the data points published by the gateway in studio 1 of Melnitz Hall, while the users shown in the ACL of the prefix /ndn/ucla.edu/bms/building will have access to the entire BMS network on the UCLA campus. Theoretically, each data point could have its own ACL, but we believe that in practice the granularity of access control does not need to be smaller than the per-gateway level.

Gateway Configuration

When the gateway is installed and connected to the network for the first time, it must contact the BMS manager daemon to retrieve its configuration file before publishing sensing data. Specifically, it needs to obtain its own identity name under which its public key is published, the application and installation specific prefixes under which it should publish data, its access control list, the BMS manager's public key that will act as the trust anchor, and a local routable name to set up bidirectional communication with the manager.[5] Once configured, the gateway generates its public/private key pair locally and sends the public key to the manager, which then creates the certificate for the gateway's identity and publishes the certificate into repo.

The communication between gateway and BMS manager daemon is through NDN Interest/Data exchange. Initially, the daemon listens on a well-known prefix and waits for Interest packets from the gateways. (This prefix must be reachable within the BMS network.) A symmetric key is shared between the gateway and the manager to bootstrap the trust relationship. We assume that when the gateway device is shipped from the manufacturer it is configured with a device serial number, which can be used to derive this symmetric key upon installation by a human operator. During the auto-configuration process, the NDN packets between gateway and manager daemon are either encrypted or signed (using hash-based message authentication code, HMAC) by this symmetric key to provide protection against man-in-the-middle attacks.

Configuration update is accomplished through two-way Interest-Data exchanges. When a gateway's ACL is updated (e.g., a new user being added), the manager first publishes the new ACL into repo and then sends an Interest to the gateway's local routable name to notify the change. The gateway returns a data packet to acknowledge the notification, and sends another Interest to the manager to ask for the name of the latest ACL (through which it can retrieve the actual ACL data from repo).

Data Publishing and Access Control

Sensor data published by the gateway is packaged in JavaScript Objenct Notification (JSON) format. To enforce access control, it is encrypted, and only a legitimate user can decrypt it. Since the same data may be accessed by many users, the system employs one level of indirection to avoid publishing an encrypted copy per user: the sensor data is encrypted using a shared symmetric key (sym-key), while the distribution of the sym-key still follows the asymmetric encryption scheme.

The sym-key is generated by the gateway and changed regularly at a reasonable frequency (much lower than the sensor data rate). Every time a new sym-key is created, the gateway starts a key distribution service (KDS) process, which goes through its ACL, fetches and verifies each user's public key, and then publishes an asymmetrically encrypted copy of the sym-key for each legitimate user under the /kds sub-namespace shown in Fig. 3. The name of the sym-key is constructed as /<gateway-prefix>/kds/<user-public-key-id>/<timestamp>, where the timestamp describes the key version. Every data packet will encode the timestamp of the decryption key in the payload so that users can reconstruct the sym-key name immediately upon data reception.

With one significant difference, the encryption-based access control described above is similar to the TLS/SSL protocol widely used in today's Internet. Both use public/private key pairs to authenticate communication peers and then generate

[5] *The last one is resolved locally through NDN prefix auto-configuration (similar to DHCP in IP networks).*

[6] *Note that our access control scheme is also different from the one proposed for lighting control systems in [11], where the signed Interest is used to implement real-time device control. Here we are targeting the data acquisition applications where low-latency and direct communication is not necessary.*

(a)

(b)

Figure 5. UCLA Melnitz Hall monitoring system deployment: a) monitored space; b) electrical demand monitor.

a shared secret to encrypt data exchanged between the two parties. However, the fundamental difference between TLS and the scheme described here is that in our data-centric approach the encryption key is associated with the *data* itself, rather than the communication channel. In this way, the maintenance of a secure communication perimeter is not required for data security.[6]

Current Deployment at UCLA

As a first step toward validating NDN-based BMS research in practical deployments, we have implemented an NDN-BMS prototype to control the distribution of sensing data collected from two industry-standard systems in different facilities on the UCLA campus. We transitioned an existing chilled water flow and electrical demand monitoring system from using the industry standard Modbus/TCP protocol to publishing data using the NDN-BMS on the NDN project's testbed (Fig. 4). With assistance from Siemens and UCLA Facilities, we also designed and installed a dedicated electrical demand monitoring system for a studio in the UCLA School of Theater, Film and Television, shown in Fig. 5. Internally, this monitoring system uses the BACnet/IP protocol to distribute sensor data, which is then published onto the NDN testbed using the NDN-BMS.

Our prototype implementation employs a Python-based data publishing service and a browser-based data visualization interface using our Javascript NDN library, NDN-JS [12] to fetch and process the data. The data publishing service communicates from a trusted IP address using legacy BMS protocols to sensor gateways in both buildings, which are protected behind firewalls. The service packages the data in JSON format and publishes it into an NDN repository process running on the same host. The server also hosts the website that provides the data retrieval and visualization interface for public users described above. This user interface employs NDN.JS to issue Interest packets to the NDN network, employing an Exclusion filter in the Interest [1] to specify the relevant timestamp range for the requested data sequence. Fetched data is decrypted and plotted as, for example, a time series on the web page using a JavaScript visualization library, Envision.js.

To decrypt the sensor data packets, the Web application first fetches the symmetric key that is encrypted using its public key.[7] The symmetric key is decrypted using a PKCS#1 V1.5 algorithm and further used to decrypt the sensor data that has been encrypted using AES-CBC cipher.

Currently, the signing policies and BMS root public key are preconfigured in the JavaScript code so that the web interface can easily verify the data packets following the chain of trust.[8]

Figure 6 shows a snapshot of the visualization interface, which displays the electric current from one sensor located in the Strathmore building. In our deployment, the configuration process operates entirely in the NDN name space. It does not require cross-layer configuration as needed in many current IP-based BMS, in which one must configure VLANs, IP configuration, and BMS protocol-specific addressing even for basic data access, and protocol-level security is likely not even available in many installations.

We did not carry out performance measurements on the prototype system, as it is a simple pure Python implementation not optimized for performance metrics in the first place. However, we can still analyze the system efficiency at a higher level from the following perspectives:

- The NDN-BMS asynchronous data retrieval model, in which application requests are supported by a distributed set of repositories, is likely be more responsive and flexible than synchronous data fetching in traditional BMSs, as the sensor data are prefetched from the source and published as NDN data packets by one or more repositories. The overhead of collecting, encrypting, signing, and packaging sensor data will occur as the data are generated, in background jobs.
- The load on BMS data servers in our approach is likely to be lower than in typical IP-based systems, as the data distribution model of NDN and encryption-based access control eliminates the need for clients to interact directly with the data source; network caches and repositories can also respond to the same Interests. This will also make our system resilient to DDoS attacks, which are common in traditional client-server applications.
- The basic security approach of our system is expected to scale better than relying on channel-based security, because the sensor data are encrypted only once and used to serve multiple users, which will naturally enjoy all the benefits from data caching in the NDN architecture. In contrast,

[7] *The Web interface currently uses a built-in key pair, which has been authorized by being signed and published under the bms namespace. In future versions, per-user keys will be employed.*

[8] *JavaScript has some security limitations that may prove challenging for production applications, but it has been a valuable prototyping environment; all the cryptographic libraries (RSA-SHA256 signature, PKCS#1 asymmetric cipher, and AES-CBC symmetric cipher) are implemented in pure JavaScript, so the code can run in any modern browser.*

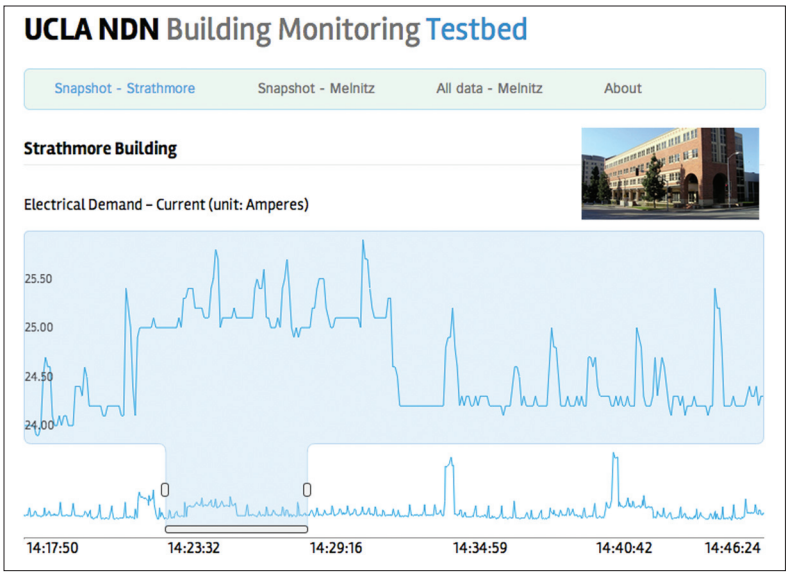


Figure 6. Browser-based data visualization interface, which uses NDN-JS to fetch and decrypt the sensor data.

traditional IP security solutions will have to encrypt the same data for each user over a transient secured connection, in addition to the overhead of setting up the secured connection every time the user contacts the server.

Conclusion

This article describes a novel building management system we designed and prototyped on top of the NDN architecture. We developed a working solution for user management, device configuration, data publishing, and access control. The Python-based prototype is deployed on the UCLA campus in conjunction with traditional BMS solutions. This work is an initial attempt to create a secure data-centric BMS architecture and provides valuable insights into general NDN application design patterns, such as using hierarchical namespace across application data and user identities to simplify user authentication and data access control.

Acknowledgments

This work is supported by the National Science Foundation (CNS-1040868 and CNS-1248049). We also appreciate the collaboration of UCLA Facilities Management.

References

[1] L. Zhang *et al.*, "Named Data Networking (Ndn) Project," tech. rep. NDN-0001, Oct. 2010.
[2] Van Jacobson *et al.*, Networking Named Content," *Proc. CoNEXT*, 2009.
[3] D. Smetters and V. Jacobson, "Securing Network Content," *PARC*, 2009.
[4] X. Fang *et al.*, "Smart Grid — -the New and Improved Power Grid: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 14, no. 4, 2011.
[5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, 2010, pp. 2787–2805.
[6] K. Stouffer, J. Falco, and K. Scarone, "Guide to Industrial Control Systems (ICS) Security," tech. rep. 800-82, NIST, June 2011.
[7] ISO/IEC 14908-1:2012, 2012.
[8] Modbus, 'Modbus Application Protocol Specification v1.1b3," tech. rep., 2012.
[9] ASHRAE, "Standard 135-2010 — Bacnet — A Data Communication Protocol for Building Automation and Control Networks (ANSI approved), tech. rep., Modbus, 2010.
[10] OPC Foundation, "What Is the OPC Foundation?," tech. rep., 2012.
[11] J. Burke *et al.*, "Securing Instrumented Environments over Content-Centric Networking: The Case of Lighting Control," *Proc. IEEE INFOCOM 2013 NOMEN Wksp.*, Apr. 2013.
[12] W. Shang *et al.*, "NDN.JS: A Javascript Client Library for Named Data Networking," *Proc. IEEE INFOCOM 2013 NOMEN Wksp.*, Apr. 2013.
[13] J. Ortiz and D. Culler, "A System for Managing Physical Data in Buildings," tech. rep. EECS-2010-128, EECS Dept., UC Berkeley, 2010.

Biographies

WENTAO SHANG (wentao@cs.ucla.edu) received his B.E and M.E degrees from the Electronic Engineering Department of Tsinghua University in 2009 and 2012. He is currently a Ph.D. student in the Computer Science Department of the University of California at Los Angeles (UCLA). His research focuses on Internet architecture and protocols. He is working on the NDN project, building applications to explore the newly proposed data-centric network architecture.

QIUHAN DING (dingqiuhan@gmail.com) is currently a senior undergraduate in the Department of Electronic Engineering at Tsinghua University. She participated in the NDN project in summer 2013 at UCLA. She is currently doing research in the Lab of New Generation Network Technology & Application at Tsinghua University.

ALESSANDRO MARIANANTONI (alex@alexrieti.com) was a senior researcher at the UCLA REMAP for the Cyber Physical aspects of the NSF Future Internet Architecture initiative. He works on artistic, cultural, and entertainment projects integrating new media and technological innovation. As an artist and producer he has presented interactive installations in the United States, Europe, and Asia. He earned a degree in Computer Science from the University of LAquila, with a thesis on perceptual interfaces at the University of Southern California's Integrated Media System Center.

JEFF BURKE (jburke@remap.ucla.edu) is assistant dean for Technology and Innovation at the UCLA School of Theater, Film and Television where he co-founded REMAP, a joint center of TFT and the Henry Samueli School of Engineering and Applied Science, which uses a mixture of research, artistic production, and community engagement to investigate the interrelationships among culture, community, and technology. He is Co-Principal Investigator and application team lead for the NDN project.

LIXIA ZHANG [F] (lixia@cs.ucla.edu) is a professor in the Computer Science Department of UCLA. In the past she served on the Internet Architecture Board, the editorial board of *IEEE/ACM Transactions on Networking*, Vice Chair of ACM SIGCOMM, and co-chair of the Routing Research Group under IRTF. She is a fellow of ACM, the recipient of the 2009 IEEE Internet Award, and holds the UCLA Jon Postel Chair in Computer Science.