Computer Networks 86 (2015) 14–26

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A blind processing framework to facilitate openness in smart grid communications

Mehmet Hadi Gunes [a,*], Murat Yuksel [a], Hayreddin Ceker [b]

[a] University of Nevada – Reno, Reno, NV 89557, USA
[b] University at Buffalo, The State University of New York, Buffalo, NY 14260, USA

ARTICLE INFO

Article history:
Received 21 April 2014
Revised 27 December 2014
Accepted 4 May 2015
Available online 18 May 2015

Keywords:
Open cyber architecture
Power grid
Privacy

ABSTRACT

Smart grid has diverse stakeholders that often require varying levels of access to grid state and measurements. At the distribution level (i.e., MAN), smart grid provides two way communication between households and utilities. At the transmission level (i.e., WAN), multiple organizations need to share the transmission lines and cooperate with participants in their region. In this paper, we propose secure communication and computation services for smart grid to transform the current "closed cyber architecture" to an "open cyber architecture". In order to ensure the privacy and integrity of communicating parties at the distribution level, we propose to utilize the smart meters as a gateway between intra-network (i.e., HAN) and inter-network (i.e., WAN) communications, and manage incoming and outgoing traffic and mediate household devices based on the instructions from the electric utility or contracted service providers. To enhance data sharing between operators at the transmission level, we propose an open cyber architecture that utilizes *blind processing* service, in which sensitive data is transmitted through the secured channel and used in computations running in an isolated environment while the outcome is rendered only to a dedicated user or process. The "open" communication between the smart substructures and "blind" computation at operation centers will increase data sharing, minimize human intervention, and mitigate cascading events. In the paper, we provide and discuss underlying mechanisms to achieve an open cyber architecture.

1. Introduction

Power grid is a crucial infrastructure for public health, safety and welfare. Proliferation of renewable energy-based electric power production, increased use of electric vehicles, and upgrading the aging electricity infrastructure for more efficient grid operations are only viable with smarter monitoring, control and consumption of the electrical energy. It is not possible to achieve the nationwide visions for a smarter grid, if the current control, monitoring, and consumption practices are not significantly changed in high voltage transmission and medium/low voltage distribution levels.

A key factor of the power infrastructure is its multi-owner property at the *transmission level* of high-voltage interconnected grids. The power transmission networks are physically inter-connected; however, the electrical and financial energy markets are governed by independent system operators (ISOs) in different *markets*. Each ISO monitors (i.e., *operations* domain) and controls (i.e., *service provider* domain) its own region and only provides power flow information on tie-lines between other transmission regions. The existing cyber-architecture in the power grid provides limited information exchange among domain owners and ISOs due to energy market constraints and trust boundaries. This "closed" cyber-architecture leaves the power grid vulnerable to

* Corresponding author. Tel.: +1 775 784 4313.
E-mail addresses: mgunes@unr.edu (M.H. Gunes), yuksem@cse.unr.edu (M. Yuksel), hayreddi@buffalo.edu (H. Ceker).

http://dx.doi.org/10.1016/j.comnet.2015.05.004
1389-1286/

cascading events and makes it difficult to detect potential problems and can lead to catastrophic failures [1–3]. As emphasized in the NARC's report [1], one of the primary weaknesses in need of attention is *"communications within the ISO and with its neighboring control areas and reliability coordinators"*. Additionally, potential coordinated attacks on these systems require the infrastructures to be more automated and self-healing [4]. As the power grid becomes more dynamic with renewable resources that provides intermittent energy, accurate monitoring and reporting is required [5–7]. The increased information sharing will thus enhance the adaptability of the power transmission grid with the proliferation of distributed renewable energy generation.

Such inter- and intra-ISO communication capabilities necessitate mechanisms to securely and efficiently exchange sensitive data for system modeling and monitoring. In order to protect both the electric utility and the user against adversaries including malicious users or external cyber attackers, we need to enhance the privacy of the user and ensure the integrity of the communication. We propose a system model that creates a symbiotic relationship between all actors within the power grid using *blind processing* [8]. In our "open cyber architecture" model, sensitive data will be transmitted through the secured channel and used in computations running in an isolated environment while the outcome will be rendered only to a dedicated user or process. Traditionally, security mechanisms are deployed to protect the transmission channel and the execution environment from third parties based on the security requirements of the data. In *blind processing*, we establish a secure channel between trusted processes which are concealed from the rest of the system, including the root processes [6].

At the *transmission level*, we propose development of an "open cyber architecture" where information sharing is the norm for ISO operations. However, such openness requires handling of various market and trust conflicts. In order to achieve open communications and promote information sharing, we develop *blind processing* service that provides authentication, privacy, and integrity assurances. Blind processing will enable the advantages of additional information exchange while respecting electrical energy market constraints and trust boundaries over the operation of the power grid infrastructure.

At the *distribution level*, we aim to revolutionize the relationship between the utility and customers via privacy protecting smart meters. The utility will be able to monitor the electricity consumption of the customer in a more detailed manner while the customers can be well informed with the cost and the amount of energy they are consuming. Secure communication can also help the utilities to inform their customers of price change during peak consumption times. Moreover, power generated at home (by solar panels, wind turbines, etc.) will better be integrated to the system.

Contributions of this paper are in two directions (i) *open cyber architecture* in Section 2 (we provide an assessment of open versus closed cyber architecture in Section 2.1 and discuss issues in the power grid communications in Section 2.2) and (ii) *blind processing prototype* in Section 3 (we provide a prototype for blind processing systems that will enable increased information sharing in an open manner by power operators in Section 3.1, and then analyze performance overhead in Section 3.2 and security issues in Section 3.3). We conclude the paper in Section 4.

2. Information sharing via open cyber-architecture

The main goal of proposed open cyber-architecture is to enhance reliability and efficiency of the large-scale multi-owner power grid infrastructure. The existing systems typically use a centralized cyber-architecture and strictly hide proprietary information from other owners. Though a "closed" approach (as in Fig. 1) hiding proprietary information makes sense in terms of business goals, the technical viability of the overall system depends on safe and sufficient sharing of basic technical information in a relatively "open" manner (as in Fig. 2). Information sharing among owners is critical to attain the needed robustness for power grid. A key proposition is to increase information sharing through more regulated means and essentially make it part of the physical system itself even to the extent that the owners may not be able to avoid sharing of some of the market related information.

The basic idea of sharing crucial information has successfully been implemented in some large-scale systems. For instance, the Internet requires its participants to provide basic connectivity information. Otherwise, the participant cannot be part of the connected network. This implicit reinforcement of information sharing is mainly driven by the "fate sharing" that naturally exists in the overall system. Participants become willing to share the information (and potentially other resources) in order to make "the whole ship float". Through trusted computing mechanisms, we aim to extend this paradigm to power grid communication infrastructure.

We abstract components of "open" communication as follows:

- **Integrated secure communication:** In order to provide means to share information, subsystems (or components) of the power grid must have secure communication capabilities integrated with the physical substrate.
- **Self-healing via automated control:** Usage of the shared information must respect the market rules and policies set forth by the domain owners. Thus, components must control the underlying systems based on domain owners' desires. Further, the system should be automated and reduce dependency on humans to resolve crisis situations. This is critical since the time required to respond to a crisis is mostly much shorter than human operation timescales.
- **Distributed planning via smart subsystems:** Since robustness of the power grid is crucial, individual components must have the planning and learning capability to be ready for unexpected events.

At transport layer, we can utilize *data aggregation* mechanisms [9,10] to minimize grid management overhead due to small-sized periodic grid measurement data as in Fig. 3. Providing GPRS/WiMAX capability for every smart meter is not cost-effective as WiFi technology is much cheaper to operate than GPRS/WiMAX. Additionally, we need to filter some of the critical proprietary information from other domains and data aggregation help enhancing data privacy.

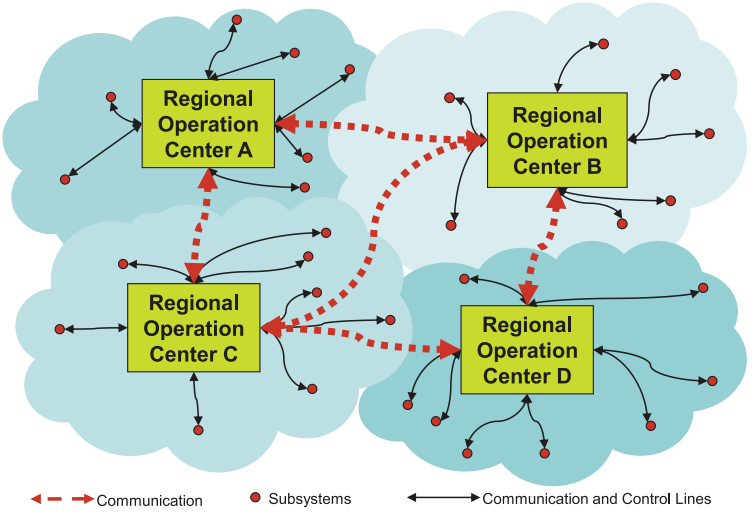

Fig. 1. Closed cyber-architecture: Information sharing takes place at large time-scales and control is limited within a domain. Decision making happens at centralized locations.

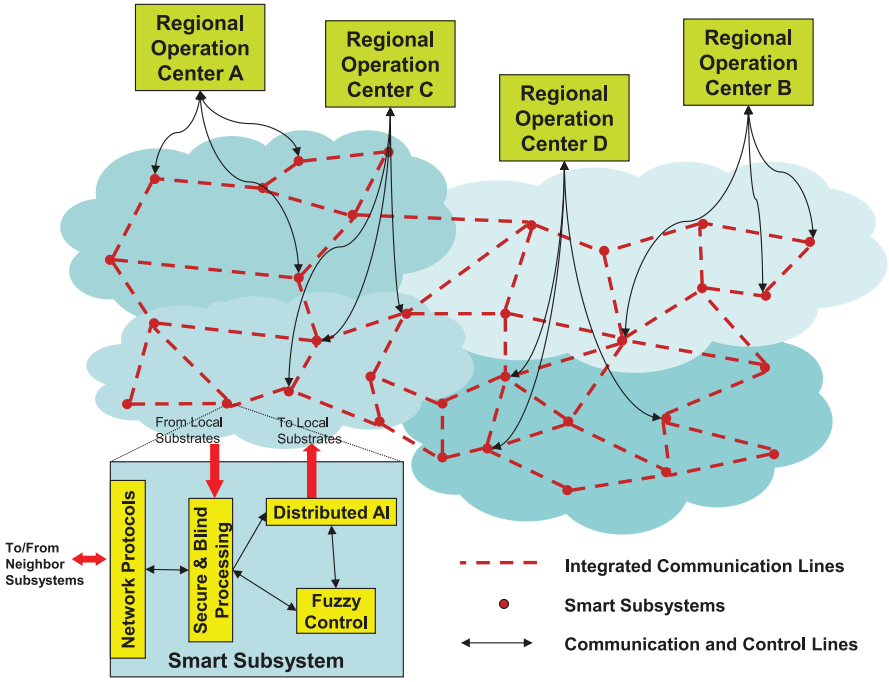

Fig. 2. Open cyber-architecture: Information sharing is integrated into the physical system and takes place at operational time-scales. Such fine-granularity and open communications enable distributed decision-making mechanisms and self-healing automated control of the physical system.

Moreover, due to the large size of power networks, disseminating all the state data sensed at substations is impractical, and routing based on importance of the data is mostly necessary. Similar issues arise with Internet data flow as networks grow analysis of information becomes a challenge [11]. Even tracking connectivity information for all connected devices is prohibitive [12,13] and management of such large networks is a challenge [14]. Researchers have studied priority/value-based forwarding [15] (with support from intermediate router queues). Utilities can employ two-staged data dissemination architecture for realizing importance-based routing: (i) *proactive flooding* of the minimum state data required (e.g., voltage and current levels of major power transmission lines) to detect risk of an important event (e.g., failure of a power transmission line), and (ii) *reactive on-demand transfer* of detailed state data following detection of a risk of a major event. Though the amount of data to transfer will be small in the proactive stage, the reactive stage must cope with huge amounts of data transfers being requested almost simultaneously. This is standard practice since an event

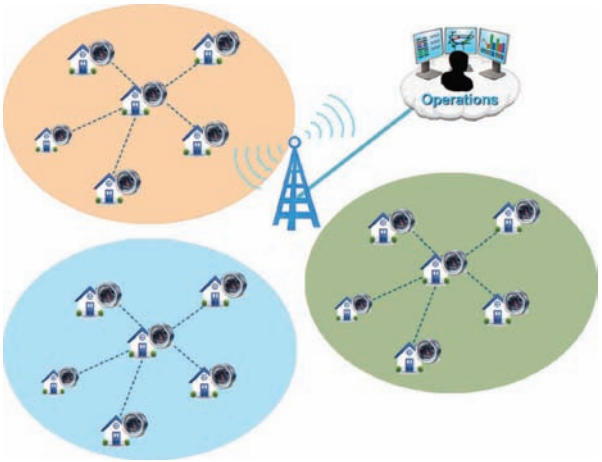

Fig. 3. Data aggregation.

will trigger many operators and smart substations to ask for detailed information about the topology around that event. This "flash crowd" [16] phenomenon exists in many networked systems (e.g., peer-to-peer) as the demand profile is quite conceivably heavy-tailed. The complexity of the problem is higher here since the set of substations/operators interested will be different for each "event". Thus, utilities can implement this reactive transfer stage with multicast.

At distribution layer, ISOs can aggregate data deliver filtered data. Data aggregation has extensively been analyzed in sensor networks to minimize the transmitted data bandwidth due to the limited sensor power [17]. Similarly, ISOs can aggregate state data when they exchange information with other ISOs at substations located on the ISO boundaries. Aggregation algorithms will minimize traffic exchange and reduce computation at substations. ISOs can filter the proprietary information and non-critical data so that large amount of intra-ISO state information can be aggregated before sending to other ISOs. Such aggregation at the ISP borders will enable ISOs to selectively hide or expose data generated from internal substations. Neighbor ISOs can further filter the incoming data at their borders to assure security as well as reduce the size of the incoming data to the levels they need to operate on.

2.1. Assessment of "open" versus "closed" cyber architecture

In order to evaluate the benefits of proposed OCA, we use IEEE 118 bus standard test system in simulation environments. The bus system, shown in Fig. 4, corresponds to a portion of the American Electric Power (in the Midwestern US) and is widely considered an important benchmark for the power industry [18]. The network is divided into four regions to demonstrate the "closed cyber-architecture". The four ISOs are responsible for operating the grid and running the energy market auctions while communicating with each other at a minimum level.

The application of the proposed open-cyber architecture to the power grid can be demonstrated on the IEEE 118-bus system controlled by four ISOs. In our simulation, the sequence of events starts with a simultaneous fault where a 220-MW power plant at bus 25 in ISO 1 area is tripped (out of service, outage) due to a mechanical failure at the same time with a transmission line between buses 27 and 28. These

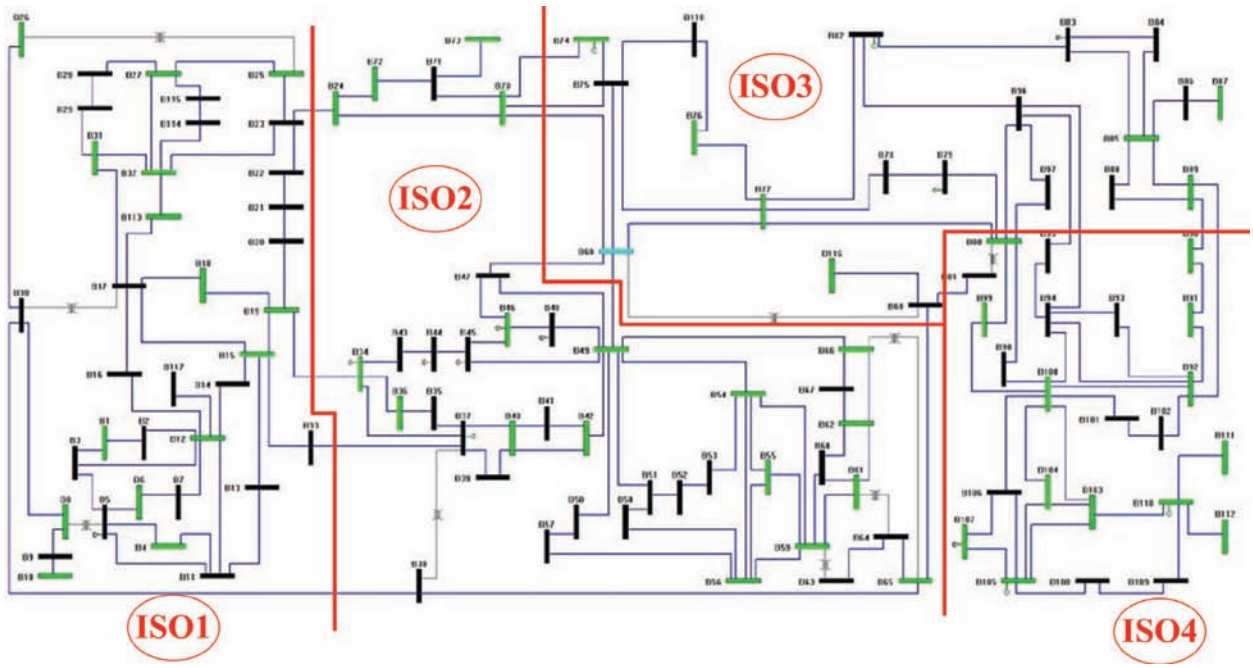

Fig. 4. IEEE 118 bus sample in Power Educational Toolbox (PET) [19] (green buses/nodes represent the generation buses, black buses represent the load buses and the blue node represents the reference bus). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

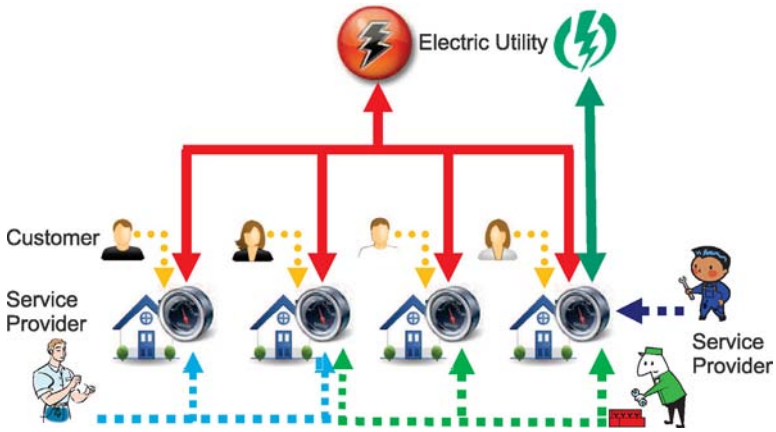

Fig. 5. Metropolitan area network.

failures cause overloading on 2 other transmission lines and 2 other transformers in ISO 1 and they eventually cause the loss of a 400-MW generation in the same region. The overloading of equipment and low frequency due to heavy load causes the event to cascade towards other ISOs resulting in 3 transmission line and 5 generator outages in ISO 2, 1 transformer, 5 generator and 2 transmission line outages (tie-line between ISO 3 and ISO 4 and between ISO 2 and ISO 3) in ISO 3 and finally 3 generator and 2 transmission line outages (tie-line between ISO 3 and ISO 4). Eventually there is not enough generation to supply the load and the whole system is out of power. The events starting in the upper left portion of the system eventually cascaded and resulted in a blackout.

In the open-cyber architecture the energy management systems of the ISOs will communicate with each other with full information exchange. In the above case, there can be two scenarios, which will prevent the blackout. Right after the first simultaneous events (transmission line and generator outages) resulting in an overload on transformer between the buses 25 and 26 in ISO 1; ISO 2 can immediately take action and reduce the overload by changing their generation schedule at bus 24 by increasing the generation by 150 MW. This action will relieve the overload on the transformer in ISO 1 and the dissemination of the event can immediately be prevented. In the second scenario, ISO 2 and ISO 3 can act together and change generation schedules in their areas to export power into ISO 1. ISO 2 can increase the generation at bus 24 by 100 MW this time while ISO 3 can increase the generation at bus 74 by 100 MW. These coordinated actions relieve the overload from the transformer between buses 25 and 26 in ISO 1 and the blackout can be prevented.

2.2. Power grid communications

In the proposed OCA, the smart substations of the power grid will be interconnected through a communication network "integrated" with the power system infrastructure. Unlike the existing communication architecture, the OCA's smart substations will be part of a self-operating (and potentially disparate from the existing networks like the Internet) communication network. In order to provide a secure communication infrastructure, it is essential to analyze all levels of the power grid. We present our approach from the top (i.e., ISO) level to the bottom (i.e., device).

2.2.1. Wide area network (WAN) communications

WAN consists of ISOs whose operation centers are interconnected as in Fig. 2. In this paper, we propose to transform offline communication between human operators with an online communication between SCADA systems. This additional communication network will greatly enhance the overall health and resiliency of the power grid.

Inter-ISO: Each utility is abstracted as a set of substations interconnected with substations of other utilities. This constitutes a mesh topology rather than a tree topology where substations exchange data and use it for potentially autonomous decisions. The major goals of this integrated network include (i) reliable delivery of critical infrastructure state information, (ii) in-network aggregation of intra-ISO measurements, and (ii) timely and efficient delivery of important event data to the relevant ISOs or substations.

2.2.2. Metropolitan area network (MAN) communications

In our model, MAN consists of four actors: the electric utility, service providers, home owners and the smart meter as in Fig. 5. As a firewall, the smart meter shields unnecessary information from outside entities and ensures identities in the communication. With this approach, the electric utility cannot have an omniscient view of the power consuming devices within a house but can only access electric consumption and delivery related issues such as overall power usage and emergency notifications. Moreover, household devices communicate with dedicated service providers through the smart meter. Similarly, the smart meter ensures identity when a homeowner accesses the system through the Internet.

Electric utility–smart meter: The electric utility obtains timely aggregate usage information from smart meters to manage the smart grid. Smart meters provide periodic reports of power usage to the electric utility. The interval and frequency of these *report* messages may be configured as needed. The electric utility can also collect daily usage reports such as minimum, average, and maximum power consumption of users. Smart meter reporting intervals can be

scheduled by the electric utility so that packet collisions and congestion are minimized. One-to-one communications between these two parties is established only after ensuring identities of both parties. To enhance user privacy, the smart meter manages household devices while trying to comply to instructions of the electric utility. For example, during on-peak hours to shave the peak loads, the electric utility will request the smart meter to reduce overall power consumption and the smart meter will determine which devices to shut down or limit based on priorities determined by the home owner.

In the event of an irregularity in power consumption or an issue in power delivery, the smart meter generate urgent *control* messages to the electric utility. These messages trigger corresponding alarms so that necessary precautions and actions are taken by the electric utility. For example, should a smart meter report the urgency of a household fire to the electric utility, the electric utility may send a broadcast or multicast signal to smart meters within the vicinity of the reported alarm/urgency. However, in a large-scale event such as power outage, every smart meter will be generating urgent error reports towards the electric utility further consuming power and causing congestion in the communication system. Hence, based on event type, electric utility can determine thresholds for number of received errors, and then generate a *control* broadcast message to suppress smart meters. Suppression messages can increase the limits for error reporting or block certain types of messages until a new *control* broadcast message is sent to reset the parameters.

Service provider–smart meter: In our model, service providers may monitor and maintain electrical household devices through the smart meter. Each service provider must first register with electric utility and then develop contracts with individual users for specific devices. Contracted devices may generate usage reports or error messages that are forwarded by the smart meter to the corresponding service provider. The smart meter becomes a proxy between contracted devices and contracted service providers. By allowing a service provider limited access to a household device information, some privacy is compromised. This compromise can be minimized by providing only sufficient information so that the service provider can perform its job. It is important to note that service providers may gain more information about specific household devices than the electric utility. Moreover, a user may configure smart meter to obtain instructions from certain service providers. For instance, they may upgrade certain software components of smart devices [20]. This is particularly useful as software bugs are occasionally identified in code of smart devices and more efficient algorithms are developed for its tasks. Automatic upgrade is crucial for cyber security.

2.2.3. Home area network (HAN) communications

HAN consists of three types of actors: home owners, the smart meter and a set of smart and legacy devices within the household as in Fig. 6. At this network, the smart meter is the authoritative entity while home owners may actively manage household devices. Smart devices register with the smart meter by exchanging identities and public keys, if available.

Smart meter–device: At the HAN level, security requirements in communications are less strict than the WAN level.

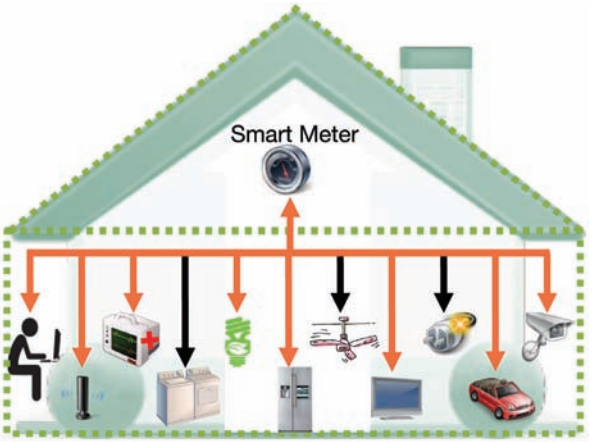

Fig. 6. Home area network.

Although it is important to provide defense in depth, we must find a balance between *usability* and *security*. As the mart meter is the centralized authoritative entity in the HAN, it provides certificates to smart devices if needed. When a smart device is introduced into the system, it will register with the smart meter. The smart meter may instruct individual smart devices to power off or change power cycle. Similarly, smart devices send usage reports and error messages to the smart meter. If an error message is received from a contracted device, the smart meter will forward a service request message to the corresponding service provider.

In the event that a smart meter must reduce power usage, the smart meter may shut down devices based on priorities set by the homeowner [21]. For example, a refrigerator can take precedence over a washer or dryer. Although it is important to limit as many forms of physical tampering of the smart device as possible, a home owner should have control of their household devices.

Owner–device: In our model, we allow owners to be able to remotely monitor and manage household devices and power usage. For the building automation, addressing is provided by the utility and users are authorized through the smart meter.

3. Blind processing for open communications

This section presents our blind processing system prototype to address privacy concerns in multi-owner information-sensitive systems. In the following, we present technical details pertaining to the blind processing system prototype. Then, we assess its overhead and related security issues.

3.1. Prototype

In providing blind processing service, we need to make sure that the remote system can be trusted not to reveal transmitted messages to anyone except the designated process whose execution is well-known. This requires security mechanisms that ensure integrity of a remote system and provide proof that the designated processes are not tempered with and isolated from the rest of the system. It is difficult to

address the issue of a malicious host when communicating with a remote system [22]. A host identity certificate does not guarantee that its administrators are not interfering with the execution of the code or monitoring its data. The software itself cannot be directly trusted as it might have been modified to intercept or modify messages. Similarly, the kernel itself is not trustworthy as we need an immutable root to trust. In general, it is better to place the trust on hardware as it is more difficult to compromise than a credential, a software, or a kernel [23].

Several techniques such as secure boot [24], authenticated boot [25], and independent auditing [26] have been developed to provide a trustworthy root in a system. *Trusted computing* incorporates these techniques and has a potential to provide mechanisms for the blind processing service [6,27]. Trusted functionality of a system is furnished by a trusted platform module (TPM), a tamper-resistant cryptoprocessor, where the TPM serves as the *root of trust* that an operating system and higher level applications can build upon. TPM extends conventional PC architectures to manage cryptographic keys, authenticate configuration of a platform (i.e., *attestation*), and cryptographically bind confidential data to a certain system configuration (i.e., *sealing*). The internal *cryptoprocessor* allows asymmetric key generation, encryption, decryption, signing, random number generation, and hashing while the internal *memory* provides storage for sensitive keys. An important aspect of the TMP is that it shields internal data structures and its computations cannot be subverted by the host system or the system administrator. Hence, TPM can provide assurance of conformed operation of the host system to both local and networked applications. Using a secure communication protocol, a remote system can request measurement results to inspect system state and to detect modifications in the system.

Several systems have been built using trusted computing concepts such as the Next Generation Secure Computing Base by Microsoft [28], Trusted Execution Technology by Intel [29], and secure co-processors by IBM [25,30]. In addition, researchers have developed systems that utilize the TPM for anonymous attestation [31], authentication [32], device attestation [33], digital rights management [34], digital signature [35], distributed computing [36], drive encryption [37], e-voting [38], grid security [39], identity management [40], mobile agents [41], on-line payment [42], on-line storage [43], peer-to-peer networks [44], policy enforcement [45], and virtualization [46]. Recently, there have been proposals to integrate trusted computing into the power grid [27,47,48]. These studies, except our work in [27], however, do not consider a security service that will hide information from the rest of the system including the system administrator as in the *blind processing*.

In our prototype, we utilize TPM chips to encrypt messages between processes and attest a remote system so that the messages are accessed only by the trusted process whose code is well-known. We have developed a prototype using a Dell Latitude E6400 laptop with a Broadcom TPM-1.2 chip using modified Linux 2.6.32.32 kernel with TrustedGRUB on Xen 4.1 hypervisor to demonstrate proof of concepts for blind processing through TrouSerS API [49]. Secure root processes provide interaction mechanisms with TPM hardware and prevent external processes from accessing protected memory. We use security kernels to set up an isolated execution environment for the process whose memory and storage will be protected from the rest of the system. We ensure an appropriate trust chain is built with a remote system starting with the TPM at its core. Before communication, we ensure that a remote peer has *correct hardware* (i.e., known devices, CPU, and TPM), *trusted computing base* (i.e., secure-kernel providing process isolation), *correct credentials* (i.e., keys and certificates), and *trustworthy state* (i.e., unaltered processes whose behavior is well-known).

We assume that systems at WAN and MAN will have TMP-like chips but not necessarily all devices at HAN. TPM chips will encrypt messages between processes and attest a remote system so that the messages are only accessed by the trusted process whose code is well-known. Security kernels also set up an isolated execution environment for the process whose memory and storage is protected from the rest of the system.

3.1.1. Privacy assurance

Fig. 7 presents a conceptual model of blind communication between two domains where we consider a multi-owner networked system to be composed of competitors.

There are different types of sub-structures in the model:

- *Type-1 virtual machines (VM-A1, VM-B1)*: These types of VMs are used for setting a common session and group key with collaborators. The group key is distributed to the sub-structures in the same domain and provides a communication channel with the other domains.
- *Type-2 virtual machines (VM-A2, VM-B2)*: Type-2 VMs are intermediary gateway between outside world and internal domain. They collect data from the subsystems inside the domain and process it to create reports. In certain cases, the filtered data to be sent to other domains is redirected to Type-1 VMs.
- *Smart subsytems* are devices that can either send the collected data to Type-2 VM to be aggregated and relayed or directly communicate with the other domains using the group key distributed by Type-1 VMs.
- *Legacy systems* are outdated devices that are being used as part of the system and still generate data for the overall health of the system. In this model, they can communicate only with Type-2 VMs. In case the data generated by legacy devices needs to be sent to the other domains, the data transfer is through Type-2 VMs to Type-1 VMs.

For blind processing, we use TPM chips to encrypt messages transmitted to both competitor systems at transmission and distribution levels. This prevents eavesdropping at the host system in addition to the communication channel. In particular, human operators/administrators will not be able to access plain-text of messages from other domains as their decryption keys will be concealed in the TPM.

Moreover, smart meters have firewall functionality so that it mediates all incoming and outgoing messages from a household. Once the smart meter successfully attests to the identity of a remote party, it can then establish a secure communication channel using stored keys to encrypt/decrypt transmitted messages. It is important to limit the amount of information that can be gathered from household to a “need to know” basis. The smart meter shields all device-specific information from the electric utility and report/negotiate

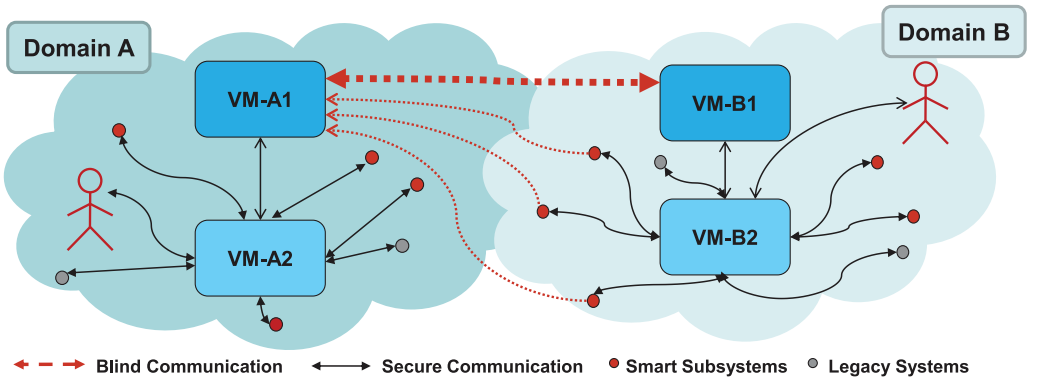

Fig. 7. Intra- vs inter-domain communication.

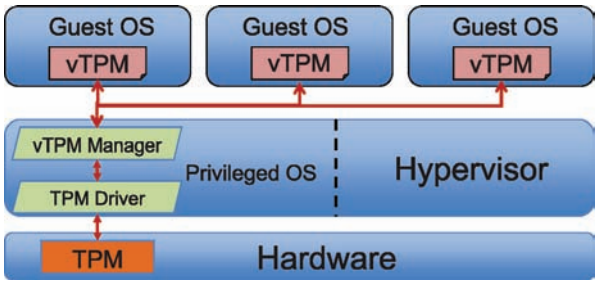

Fig. 8. Secure system model.

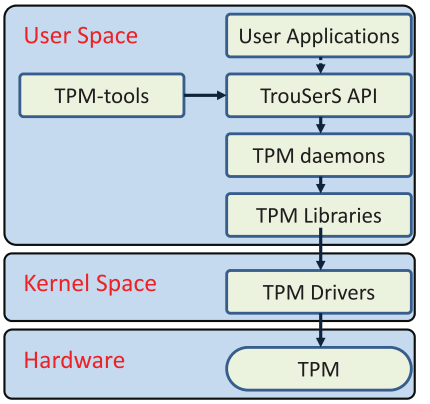

Fig. 9. Dependencies.

overall power consumption. Similarly, the smart meter provides device-specific information only to contracted service providers responsible for that particular device.

In order to provide process isolation for blind execution, we developed prototype with a security kernel. Security kernels implement specific security policies, define verifiable protection behavior of the system, and comply to the security model in controlling underlying hardware resources [50]. The Xen 4.1 hypervisor (available at http://www.xen.org/) in our configuration provides an abstract interface to the underlying hardware resources while enforcing access control rules to multiple guest operating systems as shown in Fig. 8. In our system, we implemented Task-Role Based Access Control [51]. The privileged domain operating system extends the interfaces of the underlying services and ensures isolation of applications. To ensure a chain of trust as shown in Fig. 9, we customized the Intel TXT BIOS, the TrustedGRUB and the Linux kernel v2.6.32.32.

Any stored data is encrypted using storage keys shielded in the TPM similar to mechanisms proposed in [52]. Sensitive data is sealed to a certain system state and bound to processes involved in the blind execution. The system state is measured by the Core Root of Trust for Measurement (CRTM) from system boot and includes measurements of the BIOS, the master boot record, the security kernel, O/S processes, and isolated processes involved in the blind processing.

An important issue in blind processing is how to develop trustworthy software to process data and how to establish mechanisms to verify the integrity of corresponding processes. We need mechanisms to identify processes involved in blind processing and verify their integrity. Note that detecting all software vulnerabilities is a challenging task and beyond the scope of this project.

Another important challenge for blind processing is how to provide investigative access without negatively affecting the blind processing service. Investigative access is important to ensure proper operation of the entire system and prevent malicious behavior. A system is not able to know the messages it is receiving from another domain. Hence, a malicious system may inject faulty data into the communication to affect the operation of a competitor. To prevent this, operators may use anonymized auditing that provides mechanisms to verify the integrity of data without accessing the actual data [53]. Additionally, a non-profit or a government agency may perform random inspections to assure lawful operation.

3.1.2. Remote system authentication

When communicating with a remote process, a system needs to establish its identity to prevent unauthorized access. Key distribution and verification is a central issue in any networked system [54]. In our case, the communication system is an identity based network, i.e., all devices and users at any of the levels have unique identities. These identities are used to ensure messages are sent to and received from a legitimate trusted entity similar to the public key infrastructure (PKI) [55]. Note that, we do not need human operators to

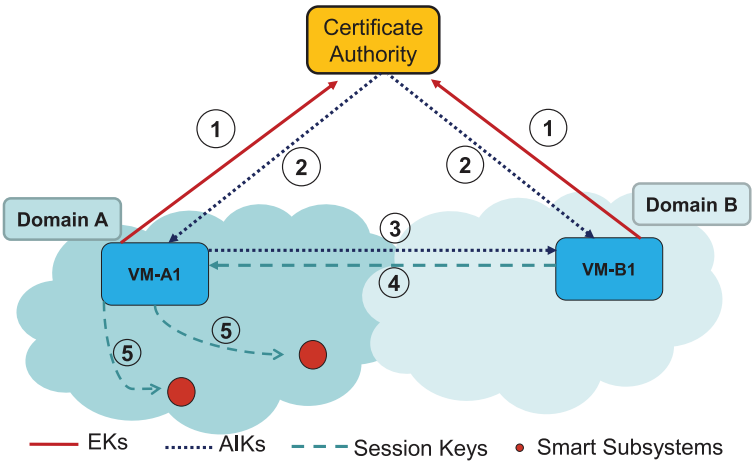

Fig. 10. CA hierarchy.

certify the public keys (as indicated in [56]) when TPMs are used. As the private keys of TPMs are embedded during the manufacturing process, the manufacturer of TPM can easily provide certificates for the public keys of devices without human intervention.

Each TPM has a unique identity, i.e. Endorsement Key (EK), currently 2048-bit RSA public–private key pairs for our prototype, sealed in the TPM by the manufacturer but AIKs are used as aliases during communication with a remote party to minimize exposure of EKs. The private keys are sealed within the chip, cannot migrate to other chips, and is not viewable outside the chip even by system administrators. AIKs are generated by the TPM as needed and signed by a Certification Authority (CA) based on the system's credentials. The credentials include an *endorsement credential* by the TPM manufacturer that testifies the TPM is genuine, a *platform credential* by the platform manufacturer that testifies the TPM has been correctly incorporated into the system, and a *conformance credential* by a testing laboratory that testifies the TPM and its platform conform to the TCG specifications. AIK testifies the system is a trusted platform with a genuine TPM and is used to establish identity and authenticity to a remote party. New session keys, which are certified by AIK, can be generated for data communication thereafter.

At WAN level (i.e., inter-ISO), each domain needs its own Certification Authority (CA) independent of other domains since Endorsement Keys (EKs) need to be private to each domain. EKs of a TPM are permanent and cannot be revoked in case it is deciphered. With a separate CA for each domain, we can authenticate processes from other domains. Authentication can be achieved by ensuring identities in a hierarchical manner, as shown in Fig. 10. First, A1 uses its EK and credentials to obtain a signature from CA1 for an AIK it generated. CA1 will then sign the generated IKE for inter-domain and intra-domain communication. Then, C1 will verify the identity of A1 through CA2, which knows the signatures of CA1. Once C1 authenticates with A1 in a similar way, A1 and C1 can exchange messages using the newly generated temporary session keys.

Similarly at the MAN and HAN levels, the smart meter, electric utility, service providers, and some of the smart devices will have certificates. The electric utility will be the authoritative certification agent in providing certificates for MAN entities. The certificate of electric utility will be stored in smart meters during installation and the certificates for smart meters and service providers will be signed by the electric utility. After a contract agreement between a smart meter and a service provider is established, both entities will exchange certificates to ensure identity and legitimacy of public keys. Similarly, the smart meter will be the authoritative entity in handling certificates in the HAN. If needed, certificates for smart devices will be signed by the smart meter and used in communication with service providers.

In order to reduce processing overhead in encryption/decryption, communicating systems typically use symmetric session keys, which are agreed upon using public key cryptography. As public key cryptosystems are considerably slower than symmetric key cryptosystems, session keys will be devised to exchange bulk of the messages [57]. Session keys can also be utilized for longer durations as actors within the WAN are not very dynamic [58].

3.1.3. Communication protocol

In Fig. 10, the communication between CA and the virtual machines is followed through public key distribution by a certificate authority.

(1) $(Cr_{A1}, EK_{A1}, AIK_{A1})$: VM-A1 uses its EK and credentials to obtain a signature from CA for an AIK it generated.
(2) $Enc^{CA}_{Priv}(Info, Cr_{A1}, AIK_{A1})$: CA then signs the generated AIK for inter-domain and intra-domain communication.
(3) $Enc^{B1}_{Pub}(Enc^{AIK_{A1}}_{Priv}(n, K_g))$: Virtual machines exchange information to set a common session key. In this case, VM-A1 sends request to VM-B1 to set a common session key K_g for group communication.

(4) $Enc^{A1}_{Pub}(Enc^{AIK_{B1}}_{Priv}(n+1, K_g))$: VM-B1 accepts and responds back with incremented nonce to prevent man-in-the-middle attack.
(5) $Enc^{A1}_{Pub}(Enc^{AIK_{Subsystem}}_{Priv}(K_g))$: VM-A1 notifies smart subsystems to use the group key K_g to report essential information directly to VM-B1.

3.1.4. Integrity assurance

All data communication systems at WAN and MAN levels use integrity assurance mechanisms as they might belong to different organizations. Additionally, as the smart meter acts as a gateway between the HAN and MAN and serve as a firewall for the HAN, it is important for the smart meter to be equipped with components that prevent hardware/software tampering. Establishing trust relationship with the smart meter provides assurances to both external and internal entities. A tamper-resistant system, for instance, protects the electric utility and service providers from attacks generated by malicious user smart meters.

Preventing a tamper attempt requires a secure package with minimal and carefully engineered access control. Moreover, the TPM-like chip mounted on a host system may be designed to quickly erase its secrets in response to tamper detection such as penetration attempts, temperature extremes, voltage variation, and radiation [25]. Defense aspects include: (1) *tamper detection* to have the device able to sense when tamper is occurring; (2) *tamper evidence* to ensure that tamper causes some observable consequence; (3) *tamper resistance* to make it hard to tamper with the device; and (4) *tamper response* to have the device take some appropriate countermeasure [59].

We utilize secure root processes of the TPM to develop authenticators that ensure integrity of processes using the CRTM as in [60]. Moreover, as CRTM performs integrity measurement at load-time, run-time vulnerabilities of critical systems can be detected using run-time attestation [61] and verifiable code execution [62]. To ensure the integrity of a system, a remote challenger will request measurements of the communicating process before sending any data. We assume the challenger has already established the identity of the remote system by verifying its AIK and ensured the presence of a trustworthy execution environment before the integrity check. The challenged system obtains relevant PCR values signed with the private AIK and gather corresponding stored measurement log (SML) entries. SML is a log of integrity changing events in the system. It is updated along with the PCR to track changes in the system state as the PCR digest has no meaning by itself. The log is stored outside the TPM as its size may grow arbitrarily. PCRs provide evidence of tampering with the log since the stored hash value only can be generated from a specific sequence of events. The signed PCR and SML values are sent to the challenger along with credentials for the AIK. The challenger then inspects the supplied credentials, analyze the SML to conform the system state by examining the sequence of events, and verify system integrity by comparing PCR values with stored fingerprints.

Integrity measurement of a complete interactive system is a challenging task, as thousands of measurements and knowledge of their fingerprints may be required for various software [63]. In our case, we are interested in the integrity of a known set of processes loaded in a deterministic order. Using a security kernel, a system ensures integrity of the TPM, the BIOS, the security kernel and a well-known set of processes providing blind processing.

Table 1
Blind processing timing overhead.

Benchmark	With (s)	Without (s)
System boot	82.145	47.395
Communication App	1.351	1.112
Computation App	8.114	7.624

3.2. Evaluations of performance impact

In evaluating the performance impact of blind processing on our system, we measured the boot time, startup time for two applications (a communication oriented one and a computation oriented one), and TPM functions.

First, we measured the timing overhead of blind processing mechanisms on the system in Table 1. Table presents boot timing of different components with blind processing mechanisms, i.e., TPM and TSS, in place or not. The numbers present average of 50 trials. The application boot times were performed in idle environments where no process excessively consumes the CPU, memory or network bandwidth. Results show negligible difference in application boot times. However in the case of system boot time, there is a considerable slowdown. This is due to the TPM measurements and iterative state verification at boot time. The TPM unseals data using the storage root key and stores hashes into the Platform Configuration Register (PCR) as it ensures the chain of trust. As this overhead is on boot time, it does not affect the operational performance of the system.

In addition to measuring the startup performance of system, we analyzed the sealing, encryption, and hashing performance of the system. Since these functions are frequently used in our prototype, we measure the performance overhead they will bring. Using `tpm_seal` to seal a file with the storage root key, we performed 50 tests of each function where we sealed files with different sizes, varying from 100KB up to 10GB as presented in Fig. 11. The overhead is negligible for reasonable sizes of data that will be transferred between the system components. For example, it takes less than 1 s for 10M for all functions to operate. Even for 100MB, it takes only 1.60 s for sealing operation, 0.45 s for hashing operation and 5.80 for 3DES. We added the simulation results for 1GB and 10GB to see the performance with big data. However, grid data typically is not that large so typical data processing overhead is expected to be less than several seconds.

3.3. Potential vulnerabilities and attacks

This section presents an overview of potential vulnerabilities we try to safeguard against. In particular, each subsection presents basic information regarding an important attack, followed by the vulnerability or possible point of entry within our architecture, and finally how our architecture addresses these security risks.

For application attacks (e.g., web browser, mobile code, and web service), wireless technology attacks (e.g., Wi-Fi and

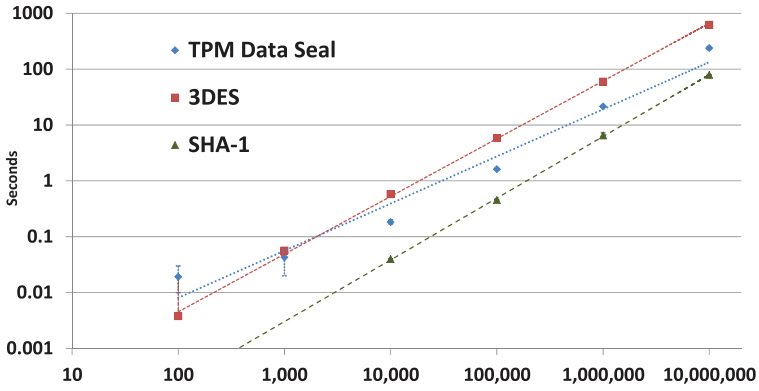

Fig. 11. Data seal timing (logarithmic scale).

Cellular), social engineering attacks, and physical attacks we need secondary defense mechanisms to prevent the utility from these attacks as discussed in [64]. For instance, one of the biggest weaknesses in infrastructures is social engineering. Unfortunately, there is no sound computational mechanism to safeguard against these types of attacks. Instead, a good approach in addressing this risk is educating all stakeholder in the system regarding the potential issues and best mechanisms.

Man-in-the-middle attacks are a form of eavesdropping where an attacker intervenes between two communicating parties, e.g. Alice and Bob. The attacker creates independent connections at each communication end, i.e. Alice and Bob, making each end believe they are talking directly to each other. However, as the attacker is intercepting the messages, the attacker may sniff or manipulate information transmitted over the channel. The attacker may additionally inject new messages into the stream intended for either end point.

In our system, an attacker may create a machine that complies to the trusted architecture in order to attempt a man-in-the-middle attack. However, a successful man-in-the-middle attack requires the attacker's machine to be verified through the trusted authority. If this is not possible, an attacker may also attempt to represent a trusted authority in order to remotely attest to a machine within the trusted infrastructure. However, *given the constraint of 2048-bit private identity and endorsement keys being created during the device manufacturing and viewable only inside the TPMs*, it would be impossible to successfully launch this attack. An attacker may attempt to brute force the 2048-bit key, but it would be infeasible to penetrate in a reasonable amount of time.

A malicious entity may also try to break the chain of trust by having a process intervene between one of the trust layers. For example, an attacker may try to subvert the guest domain, privileged domain, or the hypervisor. However, *PCRs within the TPM would change and therefore invalidate the system state*. In such violations, the system may simply flag for breach, halt the system, or restore the system to a known and verifiable secure state. Another option would be to combine the two methods where the system would retain its integrity while notifying administrators/users of a vulnerability that requires a fix.

Session injection attacks are similar to man-in-the-middle attacks such that a third party intervenes the communication channel. In our case, injection refers to data injection where an attacker exploits a vulnerability that causes processing of invalid data. This is a major area of concern as an attacker may discover a vulnerability in the system and exploit it to cause an unintended behavior. It is difficult to predict these types of attacks since systems and software are developed by humans, which are prone to errors.

In order to prevent these types of attacks, our system isolates memory resources by using virtual machines. According to the Open Web Application Security Project, application-level attacks are most popular means of entry into a system [65]. Application level attacks target databases and web applications that often are publicly accessed by external users. Our system employs access control where the security requirements of applications dictate which domain (privileged or guest) it may run on. Authorization and access control is further handled at the user level within the applications. Typical security mechanisms such as ASLR, pointer obfuscation, and non-executable memory may also be deployed within the customized kernel.

Similarly, *session hijacking* refers to the exploitation of a valid session where an attack gains some identifiable information, e.g. session key, to gain unauthorized access. In our system, we focus on protection of machine's private Attestation Identity Key (AIK), which is received after being authorized by a trusted authority, since with the private AIK an attacker can remotely attest to other machines in the network. This implies the attacker was able to circumvent the 2048-bit RSA-generated AIK or successfully masquerade as such, which is not possible *due to public key crypto system* that protects private key and securely distributes the public key. Furthermore, machines in our architecture are able to ask other machines for their Endorsement Key (EK), which contains details regarding platform credentials from the Platform Configuration Register (PCR), endorsement credentials, and conformance credentials, which may be used at either end to re-verify the attestation.

Cold boot, physical and side channel attacks refer to an attacker gaining physical access to a machine. An attacker may retrieve crucial keys after using a cold boot to restart the machine from a completely *off* state [66]. These types of

attacks rely on data remanence within memory, which may still be readable up to a few minutes after the machine has been powered off. Fortunately, these types of attacks are very intricate and require astounding esoteric knowledge to perform. In addition to being technically difficult to perform, the success rate is also not fully guaranteed. However, these types of attacks have been demonstrated to potentially be effective against full disk encryption schemes, even when TPM chips or secure coprocessors are employed.

Mitigating these types of attacks at software level is very difficult since additional protection software may itself be unreliable or ineffective. To address this security risk at the software level, systems may re-encrypt encryption keys upon disk unmount or use two-factor authentication where a pre-boot PIN or removable USB would be required alongside the TPM to boot. In order to mitigate these types of attacks, it is important for the hardware, i.e. TPM devices or secure co-processors, to be tamper-resistant. For example, any physical tampering of the device could result in the device short-circuiting. Moreover, a system may utilize advance power management, such that when a system powers off or goes into sleep mode, all sensitive information is intentionally wiped from the memory. TCG specifies compliancy for trusted systems whereby the BIOS must overwrite memory during POST if the system has not shut down cleanly. Finally, a trusted third party, e.g., a government agency or independent auditors, may perform verification checks at random intervals so that an owner does not temper the TPM chips to gain competitive advantage.

4. Conclusion and future work

The contribution of blind processing is to provide a holistic approach that integrates trust mechanisms into communication and computation. Currently, people trust the other party will act in good faith in handling their sensitive data. Blind processing mechanisms not only secures the data but also provides assurance for the data privacy even with potentially malicious system administrators. In this paper, we present general framework for blind processing and built a prototype.

As a future work, our prototype can be expanded into other multi-owner systems and cloud environments. Moreover, as we utilized off the shelf components, an optimized design can be investigated. Similarly, even though we did not observe considerable performance degradation in our prototype, we might need to optimize the system for large scale deployment and processing.

Acknowledgments

We would like to thank Dr. Cansin Yaman Evrenesoglu for his collaboration as without his contribution this paper would not be possible. This material is based upon work supported in part by the National Science Foundation, United States under grant number EPS-IIA-1301726.

References

[1] N.A.E.R. Council, Technical Analysis of the August 14, 2003, Blackout: What Happened, Why and What Did We Learn?, Technical Report, NERC, Princeton, NJ, 2004.
[2] U.-C.P.S.O.T. Force, Final Report on the August 14, 2003, Blackout in the United States and Canada: Causes and Recommendations, Technical Report, U.S. Department of Energy, Washington, D.C., 2004.
[3] N.Y.I.S. Operator, Blackout August 14, 2003 Final Report, Technical Report, NYISO, Albany, NY, 2005.
[4] J. Miller, Research on the characteristics of a modern grid: operates resiliently against attack and natural disaster, Energy Pulse 4 (3) (2009).
[5] H. Farhangi, The path of the smart grid, IEEE Power Energy Mag. 8 (1) (2010) 18–28, doi:10.1109/MPE.2009.934876.
[6] M.H. Gunes, C.Y. Evrenosoglu, Blind processing: securing data against system administrators, in: FIP/IEEE International Workshop on Management of Smart Grids, 2010.
[7] M. Yuksel, K. Bekris, C.Y. Evrenosoglu, M.H. Gunes, S. Fadali, M. Etezadi-Amoli, F. Harris, Open cyber-architecture for electrical energy markets, in: Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks, in: (LCN'10), IEEE Computer Society, Washington, DC, USA, 2010, pp. 1024–1031, doi:10.1109/LCN.2010.5735675.
[8] J. Naruchitparames, M.H. Gunes, Enhancing data privacy and integrity in the cloud, in: International Workshop on Security and Performance in Cloud Computing, 2011.
[9] V.C. Gungor, F.C. Lambert, A survey on communication networks for electric system automation, Comput. Netw. 50 (2006) 877–897, doi:10.1016/j.comnet.2006.01.005. http://dl.acm.org/citation.cfm?id=1143155.1648631.
[10] T. Sauter, M. Lobashov, End-to-end communication architecture for smart grids, IEEE Trans. Ind. Electron. 58 (4) (2011) 1218–1228, doi:10.1109/TIE.2010.2070771.
[11] B. Li, J. Springer, G. Bebis, M.H. Gunes, A survey of network flow applications, J. Netw. Comput. Appl. 36 (2) (2013) 567–581.
[12] H. Kardes, M.H. Gunes, T. Oz, Cheleby: a subnet-level internet topology mapping system, in: COMSNETS, 2012, pp. 1–10.
[13] D.S. Shelley, M.H. Gunes, Gerbilsphere: inner sphere network visualization, Comput. Netw. 56 (3) (2012) 1016–1028, doi:10.1016/j.comnet.2011.10.023. http://www.sciencedirect.com/science/article/pii/S1389128611003987.
[14] E. Arslan, M. Yuksel, M.H. Gunes, Network management game, in: 2011 18th IEEE Workshop on Local Metropolitan Area Networks (LANMAN), 2011, pp. 1–6.
[15] M.E. Sisselman, W. Whitt, Value-based routing and preference-based routing in customer contact centers, Prod. Oper. Manage. 16 (3) (2007) 277–291.
[16] T. Stading, P. Maniatis, M. Baker, Peer-to-peer caching schemes to address flash crowds, in: Proceedings of the Workshop on Peer-to-Peer Systems (IPTPS), 2002.
[17] R. Rajagopalan, P.K. Varshney, Data aggregation techniques in sensor networks: a survey, IEEE Commun. Surv. Tutorials 8 (2006) 48–63.
[18] R. Christie, Power Systems Test Case Archive, 1999, http://www.ee.washington.edu/research/pstca.
[19] F.M.A. Abur, Y. Lu, Educational toolbox for power system analysis, IEEE Comput. Appl. Power 13 (4) (2000) 31–35.
[20] J. Benoit, S. Gagnon, L. Tetreault, Securing distribution automation, Cooper Power Systems: Technical Report, March 2010, http://www1.cooperpowereas.com/PDF/Securing_Distribution_Automation.pdf.
[21] P. Communications, Smart homes, 2010, http://www.powerlinecommunications.net/smarthomes.htm.
[22] E. Gallery, C.J. Mitchell, Trusted computing: security and applications, Cryptologia 33 (3) (2009) 217–245.
[23] R. Anderson, M. Bond, J. Clulow, S. Skorobogatov, Cryptographic processors-a survey, Proc. IEEE 94 (2) (2006) 357–369.
[24] W.A. Arbaugh, D.J. Farber, J.M. Smith, A secure and reliable bootstrap architecture, in: IEEE Symposium on Security and Privacy, IEEE Computer Society, 1997, p. 65.
[25] J.G. Dyer, M. Lindemann, R. Perez, R. Sailer, L. van Doorn, S.W. Smith, S. Weingart, Building the IBM 4758 secure coprocessor, Comput. 34 (10) (2001) 57–66. http://doi.ieeecomputersociety.org/10.1109/2.955100.
[26] S.W. Smith, Outbound authentication for programmable secure coprocessors, Int. J. Inf. Secur. 3 (1) (2004) 28–41.
[27] J. Naruchitparames, M. Gunes, C. Evrenosoglu, Secure communications in the smart grid, in: Consumer Communications and Networking Conference (CCNC), 2011, pp. 1171–1175.
[28] M. Peinado, Y. Chen, P. Engl, J. Manferdelli, NGSCB: a trusted open system, in: 2011 IEEE 9th Australasian Conference on Information Security and Privacy, Springer, 2004, pp. 86–97.
[29] Trusted execution technology architectural overview, http://www.intel.com/technology/security.
[30] T.W. Arnold, L.P. Van Doom, The IBM PCIXCC: a new cryptographic coprocessor for the IBM eserver, IBM J. Res. Dev. 48 (3–4) (2004) 475–487.

[31] B. Smyth, L. Chen, Direct anonymous attestation (DAA): ensuring privacy with corrupt administrators, in: 4th European Workshop on Security and Privacy in Ad hoc and Sensor Networks, Springer-Verlag, 2007, pp. 218–231.
[32] J. Marchesini, S.W. Smith, O. Wild, J. Stabiner, A. Barsamian, Open-source applications of TCPA hardware, in: Applied Computer Security Applications Conference, 2004, pp. 294–303.
[33] J.M. Mccune, A. Perrig, M.K. Reiter, Bump in the ether: A framework for securing sensitive user input, in: USENIX Annual Technical Conference, 2006, pp. 185–198.
[34] B.A. Lamacchia, Key challenges in DRM: An industry perspective, in: ACM Workshop on Digital Rights Management, 2002.
[35] A. Spalka, A.B. Cremers, H. Langweg, Protecting the creation of digital signatures with trusted computing platform technology against attacks by trojan horse programs, in: IFIP SEC, Kluwer Academic, 2001, pp. 403–420.
[36] T. Jaeger, P. Mcdaniel, L.S. Clair, Shame on trust in distributed systems, in: 1st Workshop on Hot Topics in Security, 2006.
[37] N. Ferguson, AES-CBC+Elephant diffuser : A Disk Encryption Algorithm for Windows Vista, Technical Report, Microsoft, 2006.
[38] P.P. Tsang, V.K. Wei, Short linkable ring signatures for e-voting, e-cash and attestation, in: Information Security Practice and Experience, Springer, 2005, pp. 48–60.
[39] W. Mao, F. Yan, C. Chen, Daonity: grid security with behaviour conformity from trusted computing, in: 1st ACM Workshop on Scalable Trusted Computing, ACM, 2006, pp. 43–46. http://doi.acm.org/10.1145/1179474.1179486.
[40] N. Kuntze, A.U. Schmidt, Trusted ticket systems and applications, in: New Approaches for Security, Privacy and Trust in Complex Systems, 232, Springer, 2007, pp. 49–60.
[41] S. Pearson, How trusted computers can enhance for privacy preserving mobile applications, in: 1st International IEEE WoWMoM Workshop on Trust, Security and Privacy for Ubiquitous Computing, 2005, pp. 609–613. http://dx.doi.org/10.1109/WOWMOM.2005.52.
[42] K. Borders, A. Prakash, Securing network input via a trusted input proxy, in: 2nd USENIX workshop on Hot topics in security, USENIX Association, 2007, pp. 1–5.
[43] M. van Dijk, J. Rhodes, L.F.G. Sarmenta, S. Devadas, Offline untrusted storage with immediate detection of forking and replay attacks, in: ACM Workshop on Scalable Trusted Computing, ACM, 2007, pp. 41–48. http://doi.acm.org/10.1145/1314354.1314364.
[44] S. Balfe, A.D. Lakhani, K.G. Paterson, Trusted computing: providing security for peer-to-peer networks, in: Fifth IEEE International Conference on Peer-to-Peer Computing, IEEE Computer Society, 2005, pp. 117–124. http://dx.doi.org/10.1109/P2P.2005.40.
[45] U. Shankar, T. Jaeger, T. Jaeger, R. Sailer, R. Sailer, Prima: policy-reduced integrity measurement architecture, in: 11th Symposium on Access Control Models and Technologies, ACM Press, 2006.
[46] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, D. Boneh, Terra: a virtual machine-based platform for trusted computing, in: 19th Symposium on Operating System Principles, ACM Press, 2003, pp. 193–206.
[47] M. Jawurek, M. Johns, F. Kerschbaum, Plug-in privacy for smart metering billing, CoRR 6794 (2011) 192–210. abs/1012.2248.
[48] GridNet, Ensuring a secure smart grid, White paper, November 2010, http://www.cisco.com/web/strategy/docs/energy/SmartGridSecurity_wp.pdf.
[49] J. Naruchitparames, Enhancing the privacy of data communications within information-sensitive systems, Master's thesis. Department of Computer Science and Engineering, University of Nevada, Reno, 2011.
[50] P. Derrin, K. Elphinstone, G. Klein, D. Cock, M.M.T. Chakravarty, Running the manual: an approach to high-assurance microkernel development, in: Haskell'06: Proceedings of the 2006 ACM SIGPLAN workshop on Haskell, ACM, New York, NY, USA, 2006, pp. 60–71. http://doi.acm.org/10.1145/1159842.1159850.
[51] H.A. Narayanan, M.H. Gunes, Ensuring access control in cloud provisioned healthcare systems, in: IEEE International Workshop on Consumer eHealth Platforms, Services and Applications, 2011.
[52] U. Kühn, C. Stüble, User-friendly and secure tpm-based hard disk key management, in: Proceedings of First International Conference Future of Trust in Computing, Vieweg+Teubner, Berlin, Germany, 2009, pp. 171–177.
[53] J. Zhang, N. Borisov, W. Yurcik, Outsourcing security analysis with anonymized logs, in: SecureComm and Workshops, 2006, 2006, pp. 1–9.
[54] C. Kaufman, R. Perlman, M. Speciner, Network Security: Private Communication in a Public World, 2nd ed., Prentice Hall, 2002.
[55] A. Metke, R. Ekl, Security technology for smart grid networks, IEEE Trans. Smart Grid 1 (1) (2010) 99–107, doi:10.1109/TSG.2010.2046347.
[56] H. Khurana, M. Hadley, N. Lu, D.A. Frincke, Smart-grid security issues, IEEE Secur. Privacy 8 (1) (2010) 81–85, doi:10.1109/msp.2010.49.
[57] W. Diffie, The first 10 years of public-key cryptography, 1988.
[58] I.T.L. at the National Institute of Standards, Technology, smart grid cyber security strategy and requirements.
[59] S.H. Weingart, Physical security devices for computer subsystems: a survey of attacks and defences, in: CHES'00: Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems, Springer-Verlag, London, UK, 2000, pp. 302–317.
[60] M. Alam, X. Zhang, M. Nauman, T. Ali, J.-P. Seifert, Model-based behavioral attestation, in: SACMAT'08: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, ACM, New York, NY, USA, 2008, pp. 175–184. http://doi.acm.org/10.1145/1377836.1377864.
[61] E. Shi, A. Perrig, Bind: a fine-grained attestation service for secure distributed systems, in: IEEE Symposium on Security and Privacy, 2005, pp. 154–168.
[62] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, P. Khosla, Pioneer: verifying integrity and guaranteeing execution of code on legacy platforms, in: ACM Symposium on Operating Systems Principles, 2005, pp. 1–15.
[63] J.M. McCune, A. Perrig, A. Seshadri, L. van Doorn, Turtles all the way down: research challenges in user-based attestation, in: HOTSEC'07: Proceedings of the 2nd USENIX Workshop on Hot Topics in Security, USENIX Association, Berkeley, CA, USA, 2007, pp. 1–5.
[64] T. Flick, J. Morehouse, Securing the Smart Grid: Next Generation Power Grid Security, Syngress, 2010.
[65] Open web application security project (OWASP) top 10, 2010.
[66] J.A. Halderman, S.D. Schoen, N. Heninger, W. Clarkson, W. Paul, J.A. Calandrino, A.J. Feldman, J. Appelbaum, E.W. Felten, Lest we remember: cold-boot attacks on encryption keys, Commun. ACM 52 (5) (2009) 91–98.

Mehmet Hadi Gunes is an Associate Professor at University of Nevada, Reno. He received BS degrees in Computer Science & Engineering and Electronics Engineering from Isik University of Turkey in 2002; MS in Computer Science & Engineering from Southern Methodist University in 2004; and PhD in Computer Science from University of Texas at Dallas in 2008. His research interests include Communications: protocols, health systems, smart grid communications; Complex networks: biological networks, social networks, graph data mining, network visualization; Internet measurements: Internet topology, Internet modeling; Network security: anonymizers, private communication, secure cloud. His research has been funded by National Institute of Justice.

Murat Yuksel is an Associate Professor at the CSE Department of The University of Nevada – Reno (UNR), Reno, NV. Prior to joining UNR, he was with the ECSE Department of Rensselaer Polytechnic Institute (RPI), Troy, NY as a Postdoctoral Research Associate and a member of Adjunct Faculty until 2006. He received a BS degree from Computer Engineering Department of Ege University, Izmir, Turkey in 1996. He received MS and PhD degrees from Computer Science Department of RPI in 1999 and 2002 respectively. His research interests are in the area of networked, wireless, and computer systems with a focus on big-data networking, optical wireless, network management and optimization, cloud-assisted routing, network architectures and economics, and peer-to-peer. He has been on the editorial board of Computer Networks, Elsevier. He published more than 100 papers at peer-reviewed journals and conferences and is a co-recipient of the IEEE LANMAN 2008 Best Paper Award. He is a senior member of IEEE, life member of ACM, and was a member of Sigma Xi and ASEE.

Hayreddin Ceker is a graduate student and research assistant at the Computer Science and Engineering Department of University at Buffalo, The State University of New York. He obtained his MS thesis on "Secure Communication in the Smart Grid" from University of Nevada, Reno in 2013.