An Adaptive Encryption-as-a-Service Architecture Based on Fog Computing for Real-Time Substation Communications

Hua Zhang, *Member, IEEE*, Boqin Qin, Tengfei Tu, Ziqing Guo, Fei Gao, and Qiaoyan Wen

***Abstract*—The recent outbreak of industrial cyberattacks indicates that the current industrial network security architecture is under serious challenges. As one of the critical industrial networks, the heterogeneous and real-time substation network lacks compatibility with the conventional cryptography architecture represented by secure sockets layer/transport layer security (SSL/TLS) and public key infrastructure (PKI). To enhance the security of smart substations under the premise of low latency, in this article, we present a novel encryption-as-a-service architecture based on fog computing in this article. The architecture offloads encryption to dedicated devices and makes certificate and key management available through unified web services on the fog and cloud layers. Based on this architecture, we propose MX-SORTS, maximizing security on real-time communication of different services, an algorithm for adaptive configuration of encrypting and signing substation network traffic. By the contrast experiments with the conventional cryptography architecture, we prove that the encryption-as-a-service architecture can significantly improve the real-time and security performance of substation networks.**

***Index Terms*—Encryption-as-a-service, fog computing, smart grids, substation communications.**

I. Introduction

INCREASINGLY more shocking cyberattacks are aimed at power grids. For instance, attackers could physically destroy a power generator by exploiting the Aurora vulnerability [1]; the Sandworm attacks against the power systems led to massive black-out in Ukraine [2], [3]; hackers can gain direct access to U.S. power grid controls in the Dragonfly attacks [4], [5]. The large-scale and rapidly growing grid network security incidents indicate that the current network security architecture of power grids, especially that of the smart substation, requires a reform rather than mere security patches or guidelines.

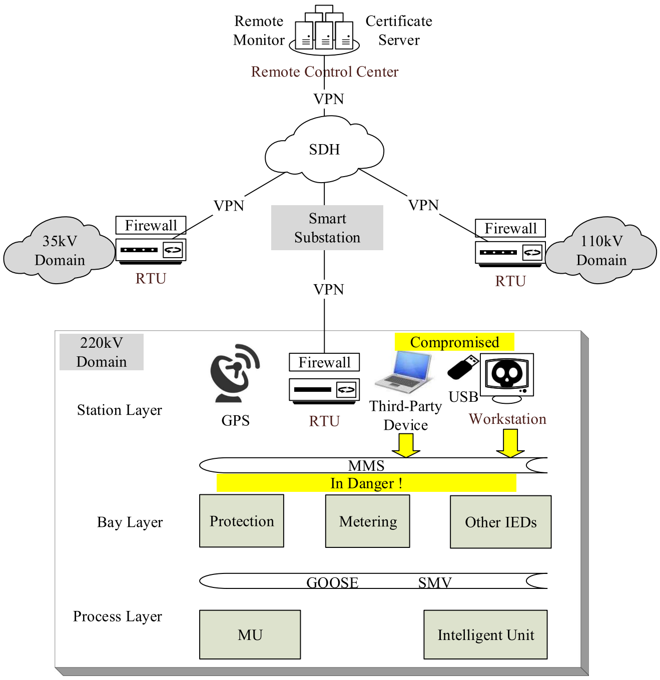


Fig. 1. Security architecture of current smart substations.

As shown in Fig. 1, a typical smart substation network is divided into several independent domains by port-based VLANs [6]. Each domain can be further divided into three layers. The process layer includes MUs (merging units) and intelligent units. Intelligent electronic devices (IEDs), for protection, metering, etc., reside in the bay layer. The station layer includes GPS, workstations, and RTUs (remote terminal units) to communicate with the remote control center through SDH (synchronous digital hierarchy). The protocols used between the three layers are defined in IEC 61850 standards [7]. The station layer and the bay layer communicate in MMS (manufacturing message specification) protocol, while the bay layer and the process layer in GOOSE (generic object oriented substation event) and SMV (sampled measured values) protocols. According to our survey, the maintenance personnel usually have

Manuscript received April 3, 2019; revised July 8, 2019; accepted October 1, 2019. Date of publication October 17, 2019; date of current version January 4, 2020. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61672110, Grant 61671082, Grant 61976024, and Grant 61972048 and in part by the Fundamental Research Funds for the Central Universities under Grant 2019XD-A01. Paper no. TII-19-1259. *(Corresponding authors: Boqin Qin; Qiaoyan Wen.)*

H. Zhang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the State Key Laboratory of Cryptology, Beijing 100878, China (e-mail: zhanghua_288@bupt.edu.cn).

B. Qin, T. Tu, Z. Guo, F. Gao, and Q. Wen are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: bobbqqin@bupt.edu.cn; tutengfei.kevin@bupt.edu.cn; guoziqing@bupt.edu.cn; gaof@bupt.edu.cn; wqy@bupt.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2948113

root privileges over the station layer devices. Unknown laptops and USBs are free to access to the station layer. Therefore, in spite of the gateway firewall and the VPN, the substation network is still exposed to various attacks, such as inadvertent portable media access [4], [8], remote updates with the falsified firmware [1], infected third-party maintenance devices [9], etc. These attacks usually occur in the network between the station and the bay layer and are closely related with the network traffic forgery, tampering, and eavesdropping, which can be mitigated by encryption and authentication of the network traffic. Meanwhile, the unauthorized operations of operators or maintainers also contribute a lot to the malfunction of power systems, so nonrepudiation and fine-grained authorization are urgent for the substation.

Cryptography is an important means of combating these attacks [10]. However, the deployment of cryptography facilities in substation networks is facing three challenges. First, few industrial devices support encryption or authentication on account of their constrained computing and storage resources [11]. Second, the stringent real-time requirements of some communication services deprecate the tedious handshake phase of secure sockets layer/transport layer security (SSL/TLS) and complex certificate management of legacy public key infrastructure (PKI) [10], when the certificate verification requires interaction with the remote control center through RTUs. Finally, the heterogeneous networks and various services with distinct security and latency requirements hinder the design of a unified cryptography strategy.

To cope with the abovementioned challenges, we propose our solution from three aspects. First, migrate the concept of encryption-as-a-service in cloud computing [12] to the substation network to offload the encryption to dedicated devices. Second, introduce fog computing [13], [14] to issue the real-time problem of cryptography management, including verification, key distribution, etc. We integrate cryptography management services into the RTU and the remote control center. Third, adapt to the heterogeneity of networks and various real-time and security requirements of diversified services with calculated adaptive cryptography configurations.

In this article, we propose a hierarchical encryption-as-a-service architecture based on fog computing and MX-SORTS algorithm for adaptive cryptography configurations in the real-time communication of smart substations. The contributions are as follows.

1) We present a novel flexible cryptography architecture through the decoupling of encryption and business processes, which integrates the certificate management, monitoring and control of cryptography, and authentication and authorization into a unified web-based platform to provide encryption-as-a-service for substation communications.
2) We introduce fog computing into the cryptography process of smart substations to ensure the real-time performance and extensibility of the encryption-as-service architecture.
3) We propose the MX-SORTS algorithm, which provides an adaptive configuration of encryption and signature for various services to balance security and latency requirements in secure substation communications.

II. Encryption-as-a-Service Architecture

We propose an encryption-as-a-service architecture based on fog computing. The architecture lifts the handshake phase, including certificate validation, key exchange, etc., to a unified fog node. The cryptographic components become an independent service rather than coupled with the business processes of the substation, which guarantees the flexibility and low latency of the architecture. As shown in Fig. 2, our architecture can be divided into the cloud layer, the fog layer, and the endpoint layer.

A. Cloud Layer

The cloud layer is mainly composed of the certificate and management server in the remote control center. It remotely monitors and controls the Crypto Managers in the substations, globally manages the authorization and certificates, and provides a user-friendly management interface.

As shown in Fig. 2, the functions of the cloud layer fall into three parts: 1) the management service includes adding and removing endpoint and crypto devices, selecting the cipher suite, starting and stopping the encryption services, etc.; 2) the certificate service is in charge of the certificate generation, storage, issue, revocation, and association with the devices; and 3) the monitoring service is mainly responsible for the display of the operation status of the Crypto Providers, the report of the connection status, and the statistics of the abnormal events. In addition, RESTful APIs are provided for the third-party applications to interact with the fog layer devices.

B. Fog Layer

The fog layer consists of one or more parallel Crypto Managers embedded in the substation RTUs. As shown in Fig. 3, each Crypto Manager holds and manages cryptographic configurations and keys in the domain of the host RTU. Meanwhile, the Crypto Manager collects traffic information from the Crypto Providers in the domain, then, uses the MX-SORTS algorithm to calculate and enforce the optimal cryptography strategy for the current domain. Besides, other runtime information, such as the connection status, cryptographic errors, etc., is collected and reported to the monitor service in the cloud layer. The certificate of Crypto Manager is trusted by the devices in its domain during the network setup. Encrypted channels are built between the Crypto Manager and the endpoint layer devices, which is referred to as the crypto management channel.

The core of the Crypto Manager (or the fog node) is the MX-SORTS algorithm, which balances the security and the availability of the network traffic. It also has some ability to counter certain kinds of denial of service (DoS) attacks. The details of the algorithm are illustrated in Section III. The algorithm makes real-time decisions upon the knowledge of local network traffic. If deployed on the endpoint layer, then it will lose its global perspective and cannot make a correct decision. But if

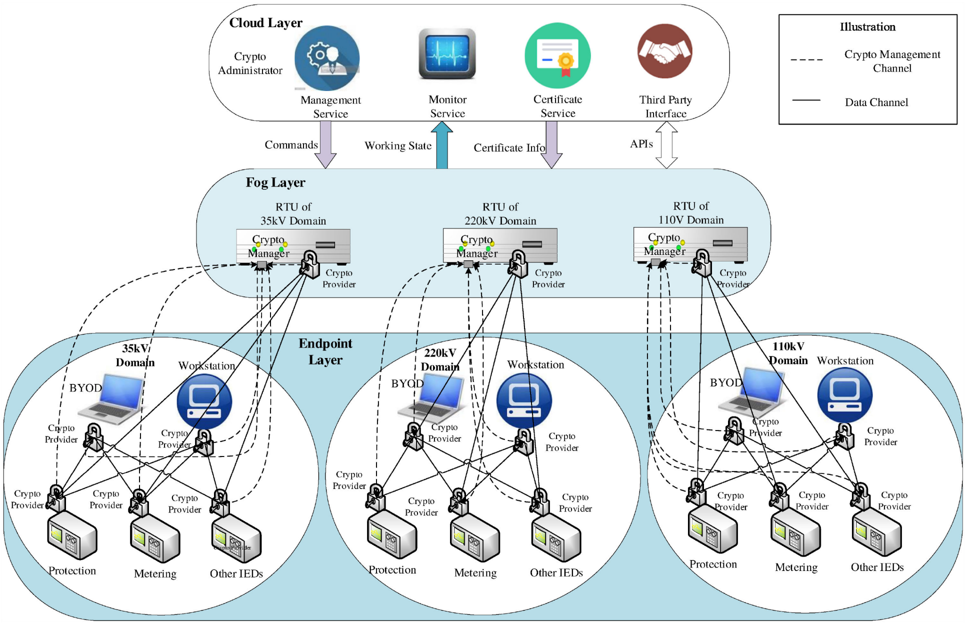


Fig. 2. Overall encryption-as-a-service architecture based on fog computing.

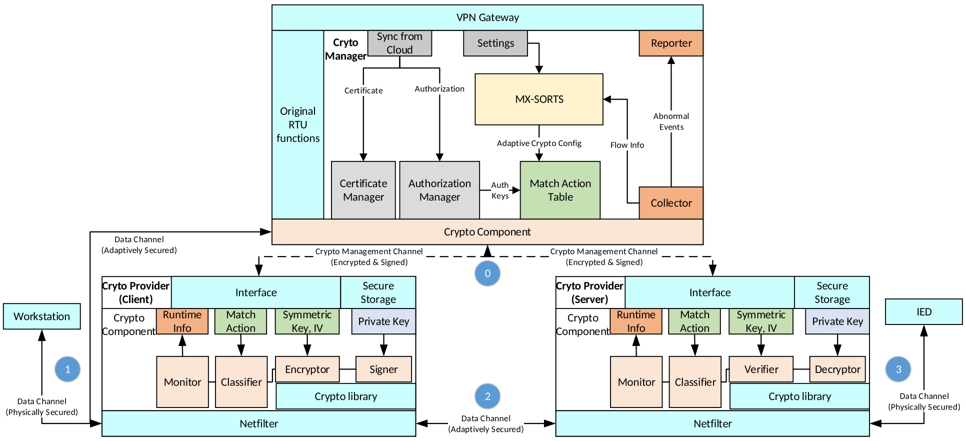


Fig. 3. Structure of the crypto edge and fog devices.

deployed on the cloud layer, it will lack the ability to make and convey decisions timely with large data transmission delay. That is why we must deploy it in fog node, where the algorithm can get enough information on the network traffic in one domain and distribute its decisions to the endpoint layer in real time.

The security focuses of the Crypto Manager are divided into two categories—the uplink to the cloud and the downlink to the endpoints. The legacy VPN is reused in the uplink to ensure the communication between the fog and the cloud, with a relaxed latency requirement of service type 3 and 6 in IEC 61850 standards [7] according to our survey, which is in between 0.5–10 s. With the prerequisite of a trusted remote control center, the uplink of the Crypto Manager is secured. The security of the downlink is guaranteed by segregation and cryptography. For segregation, VLANs ensure that one compromised Crypto Manager will not affect devices in other domains. For cryptography, the Crypto Provider, acting as the cryptographic proxy for the industrial endpoint devices, is preinstalled with the certificate of the domain Crypto Manager and the private key of the proxied endpoint device. Since the Crypto Manager mirrors the certificates of all the other in-domain endpoint devices from the remote certificate server, we can build a rather secure crypto management channel with encryption and signing on the public key infrastructure. What is more, in our experiment, the neglectable flows of crypto management cannot slow down the data channel even if they share the same physical channel. Besides, we also secure it by system enhancement and strict authorization. Therefore, we believe that eavesdropping and deception attacks can be basically prevented in the Crypto Manager on the fog layer.

As for other types of attacks such as Distributed Denial-of-Service (DDoS) attacks [15], there are two main restrictions in attacking fog nodes from compromised endpoint layer devices. First, the segregation of domain limits the attack scale of the available endpoint nodes. Second, the separation of crypto management from data channel allows the traditional firewall and intrusion detection system (IDS) to enforce stricter and more specific rules to suppress suspicious traffic in the crypto management channel, which is not as delay sensitive as the data channel. Therefore, the existing security measures can well handle the security of fog nodes.

C. Endpoint Layer

The endpoint layer is comprised of the Crypto Providers which provide encryption and signing services for the resource-constrained endpoint devices lacking encryption capacities (typically legacy IEDs). Directly connected between the negative impedance converters (NICs) of the endpoint devices and the switch ports, the Crypto Provider is designed to be a lightweight pluggable encryption-supplement proxy for the endpoint devices. As such, most of the certificate operations are delegated to the fog layer due to the constrained memory and storage of the Crypto Provider.

We try our best to make the Crypto Providers secure. According to our survey, the substation network is rigid in that each device has fixed media access control (MAC), IP addresses, and operating systems. Thus, it is possible for the Crypto Provider to memorize the MAC and IP address, and the OS-fingerprints of its proxied device, and uses the information to resist forgery attacks. The Crypto Provider is also physically bound to the proxied device and will alert when removed.

Considering the diverse locations of the Crypto Providers, we cannot provide enough security resources to them as the fog nodes. Once they are compromised by attackers, it may stop working for the device. But they will not be affected. The cryptographic keys and other related information are stored in hardware security modules [16] so as to prevent the attackers from stealing the identity and cryptography information. Thus, it is still hard for the attackers to perform deception attacks even if they successfully compromise a Crypto Provider. The benefits of separating the data channel and the crypto-management channel are that we have (largely) migrated the security of the data channel to the crypto-management channel, which means that we cannot focus on the crypto devices.

The Crypto Providers can be logically divided into client and server in terms of the signing/verification functionality during unidirectional communication processes. The client is in charge of signing packets and the server is responsible for verifying packets. In addition, both of them can encrypt and decrypt packets.

Fig. 3 illustrates the one-way process in which a client and a server collaborate to secure the traffic from the workstation on the station layer to the IED on the bay layer. The workstation tries to send a plaintext packet (typically an MMS request packet) to the target IED. The plaintext packet first passes through the client connected to the workstation, where it is classified, encrypted, and signed. The ciphertext packet is then sent to the industrial control network (the data channel) where it may be subject to illegal operations, such as eavesdropping and tampering. After that, it is classified, verified, and decrypted by the server connected to the target IED. Finally, if the packet passes verification and decryption, it will be sent to the target IED. The response packet from the IED to the workstation goes via the same routine. Every single packet can be signed if properly configured, not just the handshake packets.

TABLE I
SAMPLE FOR MATCH AND ACTION IN PACKET CLASSIFICATION AND HANDLING

Devices	Workstation1->IED1	IED1->Workstation1
SrcMAC	00:0c:29:b3:99:06	00:0d:8d:02:68:e4
DstMAC	00:0d:8d:02:68:e4	00:0c:29:b3:99:06
SrcIP	192.168.3.121	192.168.3.10
DstIP	192.168.3.10	192.168.3.121
SrcPort	-	102
DstPort	102	-
MMSType	Confirmed Request	Unconfirmed Request
ACTION	ENCRYPT-SIGN	ENCRYPT

As shown in Table I, the Crypto Provider matches a packet with a seven-tuple (SrcMAC, DstMAC, SrcIP, DstIP, SrcPort, DstPort, MMSType) to get the verdict (DROP, ACCEPT, ENCRYPT, DECRYPT, SIGN, VERIFY, ENCRYPT-SIGN, DECRYPT-VERIFY). Then, the Crypto Provider will look up the secure cache table for the corresponding cryptographic keys and other information. After that, the packet will be processed using the keys in terms of the verdict action. This is an extensible fine-grained implementation of the configurable encryption and signing.

III. ADAPTIVE CRYPTOGRAPHIC CONFIGURATIONS AND MX-SORTS ALGORITHM

The architecture provides the ability of customizing cryptographic methods for different flow types in smart substations. Now, we should find the optimal cryptographic configurations for the overall substation network. The problem is that there is a conflict between security and real-time performance in time-critical networks [17], [18]. For substation communications where availability is more important than security [19], real-time performance is a prerequisite for secure communication. Therefore, we propose MX-SORTS algorithm for adaptive cryptographic configurations so as to maximize security under the premise of real-time assurance in substation communications.

MX-SORTS are based on our empirical studies of smart substations: 1) the encryption/signing on heavy traffic service will incur great overall latency; 2) the confirmed service types are more critical than the unconfirmed ones because the former usually carry significant operations like circuit breaking [20]; and 3) the device types we are concerned with are divided into workstations and IEDs, and the former ones are more frequently exposed to third-party maintenance.

To implement MX-SORTS, we need a network security model to quantify the security gain and delay in the substation. Ray *et al.* [1] converted industrial networks N into a graph formed by subsystems/nodes (D) and links (L). They introduced the concepts of node assurance value to determine the trustworthiness of nodes and derived flow assurance value by multiplying node assurance values. We borrow some of their concepts and consider maintenance, service weight, and flow directions into our substation-specific network security model. The substation network is defined as a graph composed of devices and flows between the devices. Device risk level (DRL) and flow risk level (FRL) are defined to measure their risk levels. We also introduce security gain (SG) to measure the security enhancement for different cryptographic methods, and flow rate quota as a weight for SG and delay.

Definition 1: The risk level of device d in a substation network is defined as the product of the number of unsecured link $N(d)$ and the maintenance times per year $M(d)$

$$\text{DRL}(d) = N(d) * M(d). \tag{1}$$

Definition 2: The risk level of flow f in a substation network is defined as the product of the ratio of the DRLs of its source $S(f)$ and destination $D(f)$ devices, flow rate quota $P(f)$, and an empirical service weight $W(f)$. $W(f)$ is the critical level of the service that the flow carries. The ratio of the DRLs reflects more concerns on the flow from high-risk devices to low-risk ones

$$\text{FRL}(f) = P(f) * W(f) * \text{DRL}(S(f))/\text{DRL}(D(f))). \tag{2}$$

Definition 3: A cryptographic configuration M of a set of flow F can be denoted as $m = M(f), f \in F, m \in C$, where m is the cryptographic method applied to each flow in F, and C is the set of all available cryptograhpic methods.

Definition 4: The security gain of M is defined as the sum of the products of FRL and $G(m)$ on the flow, where $G(m)$ is an empirical security gain of m

$$\text{SG}(M) = \sum_{f \in F} \text{FRL}(f) * G(M(f)). \tag{3}$$

Definition 5: The overall delay of M is defined as the sum of the products of flow quota and $D(m)$ on the flow, where $D(m)$ is the delay introduced by m

$$\text{OD}(M) = \sum_{f \in F} P(f) * D(M(f)). \tag{4}$$

Our goal is to find a cryptographic configuration M that maximizes the $SG(M)$ with $\text{OD}(M) < \eta$, where η is the upper bound of the delay increment.

We propose MX-SORTS algorithm to get the optimal cryptographic configuration M from the flows F, crypto methods C, delay increment threshold η, and the window size τ, as shown in Algorithm 1.

MX-SORTS is built on the basis of the knapsack algorithm [21], which often arises in resource allocation where there are constraints. The primary 0–1 knapsack algorithm is defined as follows. Given a knapsack with a capacity of W and N items from 1 to N, each with a weight $w[i]$ and a value $v[i]$, the objective is to find which items should be loaded into the knapsack to maximize the sum of the values.

We make two adaptations to the original knapsack algorithm. First, we divide the "items" into groups to solve the service latency requirement. Second, we add a "punishment" procedure to prevent the excessive traffic of a flow, which is often the case in DoS attacks.

Algorithm 1: MX-SORTS Algorithm.

```
Input: $C, \eta, \tau$
Output: $M$
 1: function MX_SORTS$C, \eta, \tau$
 2:     Init
        $GROUP, MAX, PATH, LAST\_M, LAST\_F$
 3:     for every $\tau$ seconds do
 4:         Read $F$ with averaged flow rates
 5:         for $f \in F$ do
 6:             for $m \in C$ do
 7:                 if $D(m) \leq D(f)$ then
 8:                     Add $m$ to $GROUP[f]$
 9:         for $f \in GROUP$ do
10:             for $c \in [\eta : 0 : -1]$ do
11:                 for $m \in GROUP[f]$ do
12:                     $w = P(f) * D(m)$
13:                     $v = FRL(f) * G(m)$
14:                     if $c \geq w$ and
                    $MAX[c] < MAX[c - w] + v$ then
15:                         $MAX[c] = MAX[c - w] + v$
16:                         $PATH[f, m, c] = True$
17:         Traverse $PATH$ to get $M$
18:         Compare $M$ with $LAST\_M$ for set of
            $DEGRADED\_F$
19:         Compare $F$ with $LAST\_F$ for $f$ with the
            greatest change in flow rate.
20:         if $f$ is not the only element in
            $DEGRADED\_F$ then
21:             Set the ratio of $f$ to half
22:             Alert if $f$ is limited frequently
23:             continue
24:         else
25:             $LAST\_M = M$
26:             $LAST\_F = F$
27:             Generate and distribute rules out of $M$
```

We group the "items" because our objective M is a mapping from all the network flows to cryptographic methods. The "item" in the knapsack algorithm is a mapping from one flow to one method in MX-SORTS. There are two constraints on each mapping. First, there must be one and only one method applied to each flow. Second, the method shall satisfy the latency requirement of the service in the flow. The abovementioned constraints can be denoted as

$$\forall f \in F, \exists M(f) \in C \land D(M(f)) \leq D(f).$$

As such, we prepend a group initialization for each flow and select all the latency-qualified methods into the corresponding group (L5–L8). Then, we traverse the group (L9) to start the calculation. We also lift the back-traverse of η (L10) above the item traverse (L11) so as to guarantee that only one method is applied to each flow.

So far, the algorithm is still vulnerable to DoS attacks. Attackers can intentionally increase the rate of some compliant flow rapidly, which incurs overall delay, forcing other flows to degrade the cryptographic methods. To counter the excessive traffic of a flow, we innovatively introduce a "punishment" procedure (L17–L22) to our algorithm as follows.

Since the substation networks are usually stable [6] with occasional bursts, we introduce a time window τ to smooth the flow rate (L4) and distinguish the normal business burst from the constant DoS attacks. For each time window τ, we record the M and flow rates of the last calculation. Once the current calculation degrades the method of more than one flow, we find the flow with the maximum change in quota. Then, we "punish" the flow by limiting its speed. We implement this with a mature additive increase/multiplicative decrease (AIMD) scheme [22], which is widely used in transmission control protocol (TCP) congestion control algorithm [23]. In this scheme, we allow the linear increase of a flow rate until it leads to the degradation in cryptography of other flows. Then, we set the flow rate to half and recalculate M. If some flows are "punished" frequently, there is probably a misconfiguration or DoS attack, so an alert will be sent to the cloud.

We can find the max SG from MAX and trace the $PATH$ to get M (the optimal crypto method for each flow) within the overall delay increment threshold η. Since the threshold and the number of crypto methods are constants, the complexity of the algorithm is $O(N)$, where N is the number of the flows.

IV. Testbed Platform

A. Device List

Our physical devices in the testbed platform are as follows.

1) The IEDs are PLCs (programmable logic controllers) manufactured by Allen-Bradly, which plays the role as the IEC 61850 server.
2) The workstations and RTUs are Lenovo desktop computers with quad-core and 4 G memory, and FactoryTalk View SE Client, working as the IEC 61850 client. In order to test the security of the architecture, several network attack tools are also installed in the workstations. The RTUs are installed with a certificate database and other cryptograhic services.
3) The crypto edge devices are Raspberry Pi installed with Ubuntu 16.04 LTS and the cryptography facilities.
4) The crypto cloud device and the certificate storage are Inspur servers installed with Ubuntu 16.04 LTS. A web UI is provided to users for the management of each fog node's certificates and cryptographic configurations.

In our experiments, only two parts of the device could affect our simulation—communication modules and network performance. In our field survey, we have verified that the hardware specifications (including CPU, memory, and network interface cards) in our testbed and the field network are the same or similar types. The operating systems are both Linux (though in different distribution versions) and the network stack provides the same functionality. This means that our testbed can provide computing resources and communication capabilities similar to substation equipment. In our tests, the difference in latency between our testbed and the field network was less than 10%. Therefore, we can safely assume that the conclusions of the experimental results on our testbed can be applied to real substation networks.

TABLE II
DRLs of Devices

Device Types	N	M	DLR
Workstation	3	2	6
IED	2	1	2

TABLE III
FRLs of Flows

Flow Types	S	D	W	P
CREQ	WS	IED	0.75	0.1
CRSP	IED	WS	0.75	0.1
UREQ	IED	WS	0.25	0.8

TABLE IV
Security Gains and Delays of Available Cryptographic Methods

Method	SG	Latency(ms)	Delay Incurred(ms)
None	0	59.7	0
Encrypt	0.5	71.3	11.6
Encrypt&Sign	1	90.6	30.9

B. Testbed Architecture and Optimal Strategy Computing

As shown in Fig. 4, the devices in the substations are divided into three domains in our testbed, with two workstations, three IEDs, and one crypto fog device (RTU) in each domain. The workstations and the IEDs communicate with each other by IEC 61850 MMS protocol on the industrial Ethernet. The experiment mainly involves two types of services—the confirmed service and the unconfirmed service. The confirmed service includes the GetNamedList, GetVariableAccessAttribute, Read, and Write services, while the unconfirmed service includes the InformationReport services. All of the services are defined in the IEC 61850 standards [24].

Table II shows information on the devices in our testbed. The workstation communicates with three IEDs without encryption, $N = 3$. The workstation is regularly maintained twice a year according to our survey, $M = 2$. The IED communicates with two workstations without encryption, $N = 2$. The IED is regularly maintained once a year according to our survey, $M = 1$.

Table III shows information for flows in our testbed. CREQ (confirmed request) goes from workstations to IEDs; CRSP (confirmed response) and UREQ (unconfirmed request) go from IEDs to workstations. The security weight W of confirmed service is empirically set to three times that of the unconfirmed one. The flow rate quota P is measured before encryption.

Table IV shows information on the available cryptographic methods in our testbed. "None" means no encryption or signing applied. To quantify the security gains of different cryptographic methods, we refer to the general security goals—confidentiality, integrity, and availability [18]. Besides, in view of the unauthorized maintenance, we also take authentication into account [25]. We stipulate that each security goal accounts for 0.25 in SG. As such, the SG of only encryption (no integrity or authentication) is 0.5, while the SG of encryption&signing is 1.

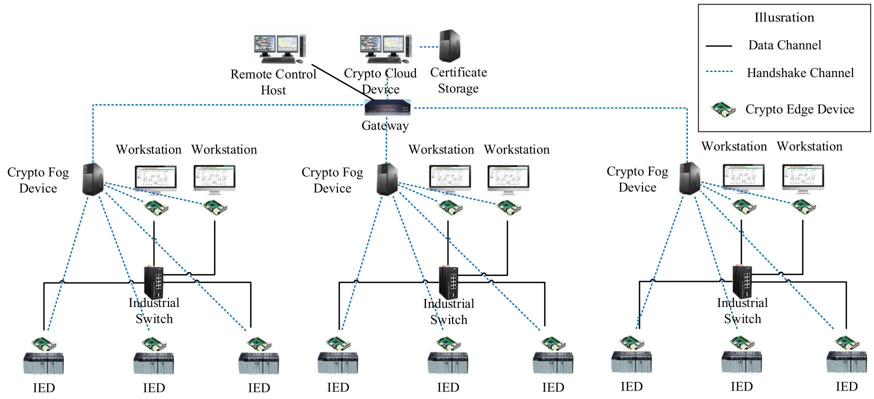


Fig. 4. Testbed platform architecture.

TABLE V
ADAPTIVE CONFIGURATIONS WITH CHANGING QUOTAS OF FLOWS

ID	Quotas	Optimal Methods	SG(10^{-2})	OD(ms)
1	0.05,0.05,0.9	S,S,E	16.25	13.53
2	0.1,0.1,0.8	S,E,E	27.08	13.53
3	0.3,0.3,0.4	S,E,N	71.25	12.75

Now, we input flows F, methods M, and a threshold $\eta = 15$ ms (25% delay incurred on 59.7 ms) to the MX-SORTS algorithm. The optimal cryptography strategy is to apply encryption on all the flows and to apply extra signing on the confirmed request flow. The result demonstrates that MX-SORTS algorithm shows more emphasis on critical traffic and applies more security measures on it. Now, the cryptographic configurations are generated and distributed to the corresponding crypto edge devices to control the cryptography processes for each type of flow.

To prove the effectiveness of the MAX-SORTS algorithm and adaptive cryptographic configurations, we change the quota of different flows under the same total flow rate. Part of the results is shown in Table V. In Table V, the sequence of the flow types is CREQ, CRSP, and UREQ. The sequence applies to both quotas and methods. S, E, N represent encryption&signing, only encryption, and none, respectively. We can see that while the quota of CREQ increases, the method degrades from S to E for CREP, and then, the method degrades from E to N for UREQ. This is because CREQ usually carries control commands, which is more important than the response in CRESP and the regular report in UREQ. Thus, the MX-SORTS adaptively tilts limited time resources to more important flows. The runtime flow quotas are collected from each Crypto Provider. Actually, we will average the flow quotas within 30 s (time window τ) before changing the cryptographic configuration in case of jitters. The adaptive MX-SORTS configuration balances between the security and the latency in dynamic flow rates. Therefore, it guarantees availability first and tries best to maximize security.

C. Security of the Traditional and Our Encryption-as-a-Service Architecture

In order to compare the security and reliability performance of the traditional and our architecture, we design several attacking scenarios including data injection attacks, replay attacks, TCP reset attacks, DoS attacks, and unauthorized operation. Data injection, replay, and DoS attacks are derived from a survey of the smart grid attacks [18], and the other two are based on our empirical study. We adopt OpenVPN [26] to implement the traditional architecture because it relies on SSL/TLS and X.509 certificate, which are the security suggestions from IEC 62351 standards [20] for smart grid security. We also manually configure the certificate and keys for each crypto node using OpenVPN.

TABLE VI
COMPARISON OF THE SSL/TLS-BASED AND THE ENCRYPTION-AS-A-SERVICE ARCHITECTURE IN DATA INJECTION, REPLAY AND TCP RESET ATTACK SCENARIOS

Architecture	Data Injection	Replay	TCP Reset
None	50/50	50/50	50/50
SSL/TLS-based	0/50	0/50	50/50
Encryption-as-a-service	0/50	0/50	0/50

1) Data Injection Attacks: To implement the data injection attacks [18], we self-implement an MMS injector with a python library dpkt [27]. On sniffing the packets matching the write service, we will modify the value and resend it to the victim. The write packets are sent to the victim 20 times and the experiment is repeated 50 times.

The results in Table VI show that all the data injection attacks successfully caused the wrong output in nonencryption in the 50 experiments, but none of them succeeded in our architecture and the traditional one. The results indicate that our architecture is at least as effective as the traditional one in dealing with data injection.

2) Replay Attacks: We implement the replay attack [28] with TCPReplay [29]. The tool is installed in one workstation to simulate the compromised device that repeats sending the circuit-breaking commands of the other workstation 20 times. The attack is carried out 50 times for the circuit break commands both in the no-encryption, OpenVPN, and encryption-as-a-service architecture.

The results in Table VI show that all the replay attacks successfully caused circuit breaking in no-encryption scenario in the 50 experiments, but none of them succeeded in SSL/TLS and

TABLE VII
COMPARISON OF THE SSL/TLS-BASED AND THE FOG-BASED ARCHITECTURE WITHOUT "PUNISHMENT" PROCEDURE IN DOS ATTACK SCENARIOS WITH COMPLIANT PACKETS

RESP(ms) PR(p/s) M	200	400	600	800	1000
None	59.68	423.43	1452.10	3907.98	TIMEOUT
SSL/TLS-based	90.38	757.93	3520.24	TIMEOUT	TIMEOUT
Punishment-disabled	89.64	441.89	1585.29	3663.88	TIMEOUT
Punishment-enabled	91.57	429.06	795.54	453.70	850.38

our architecture. The results indicate that both the SSL/TLS and our architecture can cope with the primary level of replay attacks. But their prevention mechanisms are different. In SSL/TLS, a new session key is generated and used in each session, while in our architecture, the regularly updated symmetric key with timestamps assures the timing order. Considering the burst requests that may occur in the industrial control networks, it is both quick and secure to use a predistributed and timely changed symmetric key compared to session key.

3) TCP Reset Attacks: To implement the TCP reset attacks [30], we make one workstation as an attacker, which sends a TCP RESET packet with IED's source IP address to the other workstation, causing the connection to close. The forgery TCP reset packets are sent to the victim 20 times and the experiment is repeated 50 times.

The results in Table VI show that all the TCP reset attacks successfully caused disconnection in both nonencryption and SSL/TLS architectures in the 50 experiments, but none of them succeeded in our architecture. The results indicate that SSL/TLS is vulnerable to the TCP reset attacks while our architecture can effectively avert this kind of attacks. The reason is simple—the encryption-as-a-service architecture promises a global perspective to the relatively fixed industrial control network. The MAC addresses are considered in the identity verification which invalidates the forgery. So, this is one of the security advantages of our architecture over the traditional SSL/TLS.

4) DoS Attacks: To verify the effectiveness of the "punishment" procedure, we self-implement a tool with dpkt to simulate a compromised device trying to block another device's service by flooding the device with large volumes of well-constructed compliant CREQ packets. We reset the timeout value from 1 to 4 s. The results are shown in Table VII. The packet rates (PR), here, refer to the rates of the attacker's random-generated packets sent to the victim. As for randomly generated DoS packets, they will be directly dropped by the Crypto Providers' matching rules.

The results in Table VII show that the "punishment" procedure works well in restraining the DoS attacks with compliant packets. We can find that the response time goes up and down within 1000 ms. This is because the "punishment" can "remember" the previous flow information to make decisions in sending the speed limitation command to the corresponding Crypto Providers once the rate increment of a flow affects the method choice for other flows.

5) Unauthorized Operation: We randomly generate privileges for workstations to access different IEDs. For our fog-based architecture, once we have updated the privilege in the cloud, the fog will generate new match rules and update the Crypto Edge devices swiftly. The average time is 0.13 s in our 50 experiments. But to achieve the same effect, it costs at least 5 min to configure all the crypto nodes, even on this relatively simple network. Besides, the cloud will receive an abnormal event on unauthorized access, showing the nonrepudiation operation source. Thus, our architecture can better control the third-party maintainers than the traditional one.

The results of the above-dicussed comparative experiments prove that our encryption-as-a-service architecture can better enhance the security of existing networks under real-time constraints than traditional architectures, and can effectively issue the prementioned challenges in substation security.

V. Related Works

IEC 61850 [7] divides the current architecture of smart substations into three layers—the process layer, the bay layer, and the station layer [24]. Communication interfaces are defined between layers to provide the data exchange functions for control, protection, sampling, etc. IEC 61850 defines the abstract data and service models mapped to MMS, GOOSE, and SMV. Different services provide different functions with distinct security and latency requirements. IEC 61850 defines performance classes ranging from P1 to P12 (3–10 000 ms) applied to different message types. Our article only focuses on Type 2 (medium speed messages for authomatics) with P4 (100 ms), Type 3 (low-speed messages for operators) with P5 or P6 (500 ms or 1000 ms), and Type 6 (command messages and file transfer with access control) with P10–P12 (500–10000 ms). The services mentioned in our article are of Type 3 and 6 with the latency requirement above 500 ms to allow for extra delay from cryptography. Due to the real-time nature and complexity of power equipment, the confidentiality and authentication are not prioritized during the design of IEC 61850.

IEC TC57 developed IEC 62351 standards [17], [20], [31] to enhance integrity, authenticity, and confidentiality of the IEC 61850 protocols. IEC 62351 does not recommend encryption for GOOSE and SMV due to the limited response time. As for MMS, IEC 62351 suggests the adoption of SSL/TLS for the transport layer and signing based on X.509 certificates for the application layer. The signature only covers the initial authentication and is vulnerable to at least three different attacks [32]. Besides, the role-based access control using X.509 certificates is unnecessarily complex to implement [32]. Moreover, IEC 62351 does not provide detailed security supplement solutions for the endpoint devices without cryptographic functionality.

During our survey across the country, we found that many substations did not provide any cryptographic methods inside the substation networks, only focusing on the preliminary network with VPNs, firewalls, and IDSs. Most of the industrial devices, especially legacy IEDs, did not provide SSL/TLS option in communication. Even if we trust the remote center, once the air gap is broken by on-spot access, the substation network could be exposed to nearly all kinds of attacks such as data injection attacks, time synchronized attacks, DoS attacks, dynamic system attacks, etc. [18]

There are many studies on securing smart grids by cryptography. The preshared key is a quick, prevalent but less secure cryptographic method widely used in smart grids, especially in the AMI (advanced metering infrastructure) system, which was adopted by Merabti *et al.* [33], Ye *et al.* [34], and Liu *et al.* [35]. Meanwhile, He *et al.* [19] and Tsai *et al.* [10] combined the public key cryptosystem and the symmetric key cryptosystem into the security of smart grids.

Merabti *et al.* [33] proposed the key management scheme using shared keys for smart grid communications. A preshared key at the remote end validated the sending entity, whereas a session key was used to validate data exchanges after a successful authentication. Ye *et al.* [34] proposed a comprehensive security protocol for network communication of AMI in smart grids. Shared keys may simplify the encryption process, but the distribution of preshared keys is complex. Besides, the smart metering network is quite different from the smart substation network in that the latter requires high-real-time command delivery like trip signals. Liu *et al.* [35] proposed a dynamic secret-based encryption scheme for smart grid wireless communication, where the retransmission sequence was used to update the dynamic encryption keys. The retransmission frequency is very low due to the high availability of substation wired networks. Moreover, the misbehaviors in the data transferring process might interfere with the encryption process. Thus, the dynamic secret-based encryption scheme is not suitable in our case.

He *et al.* [19] proposed an enhanced public key infrastructure to secure smart grid wireless communication networks. Tsai *et al.* [10] proposed a secure identity-based signature and encryption scheme for anonymous key distribution in smart grid. However, the above-mentioned methods require the firmware or software updates of the legacy industrial devices or even replacement of the legacy industrial devices, which requires abundant resources.

As for the security architecture enhancement for large cyber-physical systems (CPS), one promising way is to utilize network function virtualization. Wu *et al.* [36], Chekired *et al.* [37], and Li *et al.* [38] used software-defined networking (SDN) to solve the problems in safety-critical CPS. Unfortunately, it is inconvenient and uneconomical to upgrade the legacy network devices to support SDN in the substation.

It can be inferred from their works that to achieve real-time communication encryption, it is essential to realize fine-grained control of the encryption process, and change the original complex key exchange processes. Meanwhile, to adapt to large-scale deployment of complex industrial control networks, the overall management of certification and authority is required. Besides, we should offload the cryptographic computing to dedicated cryptography devices to minimize the change to current security-deficient substation networks. Therefore, we introduce encryption-as-a-service and fog computing to solve the problems.

Encryption-as-a-service [12], [39] is a concept widely used in cloud computing. It is described as a model that allows cloud service customers to take advantage of the security that encryption offers without having to install and use encryption on their own. This concept matters in the design of substation security architecture since a large number of legacy resource-constrained substation devices lack the cryptography functionality. Therefore, we implement crypto devices to authenticate and encrypt messages for industrial devices.

The latency requirement for many substation services is stringent. Certificate and key interaction waste the most time before session establishment. We gather the functions into RTU-based fog nodes for real-time performance. Fog computing [13], [14], [40] offers a place for collecting, computing, and storing on-spot data before transmitting them to the cloud, which bridges the smart substation and the cloud. It is geographically distributed and overhauls cloud computing via additional capabilities including reduced latency, increased privacy, and locality for smart grids. In view of the abovementioned (AMI) advantages, fog computing is now a promising method in partitioning computing resources [41] and privacy preserving [42] in IoT networks. Their works inspire us to design our architecture based on fog computing.

VI. Conclusion

In this article, we proposed the MX-SORTS algorithm to adaptively configure the selection of cryptographic methods on different services, so that we can balance between the delay of the encryption and real-time requirements of substation networks. We introduced the concept of encryption-as-a-service into smart substations and migrate time-consuming key management to the fog node. By integrating the certificate server to RTUs, we can achieve real-time performance in secured communications in smart substations. In the future, we will deploy our architecture in more industrial control networks, such as sewage treatment factories, which have distinct real-time requirements and communication protocols.

Acknowledgment

The authors would like to thank the reviewers for the comments on scheme security and the structure of this article, these comments were helpful for us to further improve the security of the scheme and the writing.

References

[1] M. Zeller, “Myth or reality—Does the aurora vulnerability pose a risk to my generator?” in *Proc. IEEE, 64th Annu. Conf. Protective Relay Eng.*, 2011, pp. 130–136.

[2] J. E. Sullivan and D. Kamensky, “How cyber-attacks in Ukraine show the vulnerability of the US power grid,” *Elect. J.*, vol. 30, no. 3, pp. 30–35, 2017.

[3] R. Khan, P. Maynard, K. McLaughlin, D. Laverty, and S. Sezer, “Threat analysis of BlackEnergy malware for synchrophasor based real-time control and monitoring in smart grid,” in *Proc. 4th Int. Symp. ICS SCADA Cyber Secur. Res.*, 2016, pp. 53–63.

[4] R. Tan *et al.*,“Optimal false data injection attack against automatic generation control in power grids,” in *Proc. 7th Int. Conf. Cyber-Phys. Syst.*, 2016, pp. 1–10.

[5] B. Genge, I. Kiss, and P. Haller, “A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures,” *Int. J. Crit. Infrastruct. Protection*, vol. 10, pp. 3–17, 2015.

[6] Y. Zhang, Z. Cai, X. Li, and R. He, “Analytical modeling of traffic flow in the substation communication network,” *IEEE Trans. Power Del.*, vol. 30, no. 5, pp. 2119–2127, Oct. 2015.

[7] R. Mackiewicz, “Overview of IEC 61850 and benefits,” in *Proc. IEEE PES Power Syst. Conf. Expo.*, 2006, pp. 623–630.

[8] I. Agrafiotis, J. R. Nurse, O. Buckley, P. Legg, S. Creese, and M. Goldsmith, “Identifying attack patterns for insider threat detection,” *Comput. Fraud Secur.*, vol. 2015, no. 7, pp. 9–17, 2015.

[9] S. Ponomarev and T. Atkison, “Industrial control system network intrusion detection by telemetry analysis,” *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 2, pp. 252–260, Mar./Apr. 2016.

[10] J.-L. Tsai and N.-W. Lo, “Secure anonymous key distribution scheme for smart grid,” *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 906–914, Mar. 2016.

[11] A. Ray, J. Åkerberg, M. Björkman, and M. Gidlund, “Assessing security, capacity and reachability of a heterogeneous industrial network during planning phase,” *EAI Endorsed Trans. Secur. Saf.*, vol. 3, no. 7, pp. 1–11, 2016.

[12] A. El Bouchti, S. Bahsani, and T. Nahhal, “Encryption as a service for data healthcare cloud security,” in *Proc. IEEE 5th Int. Conf. Future Gener. Commun. Technol.*, 2016, pp. 48–54.

[13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of things,” in *Proc. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[14] M. Aazam and E.-N. Huh, “Fog computing and smart gateway based communication for cloud of things,” in *Proc. Int. Conf. Future Internet Things Cloud*, 2014, pp. 464–470.

[15] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, “Network attacks: Taxonomy, tools and systems,” *J. Netw. Comput. Appl.*, vol. 40, pp. 307–324, 2014.

[16] Wikipedia Contributors, “Hardware security module,” 2019. Accessed: Jul. 1, 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hardware_security_module&old id=900229769/

[17] P. McDaniel and S. McLaughlin, “Security and privacy challenges in the smart grid,” *IEEE Secur. Privacy*, vol. 7, no. 3, pp. 75–77, May/Jun. 2009.

[18] A. Sanjab, W. Saad, I. Guvenc, A. Sarwat, and S. Biswas, “Smart grid security: Threats, challenges, and solutions,” Jun. 2016, pp. 1–8, *arXiv preprint arXiv:1606.06992*.

[19] D. He, S. Chan, Y. Zhang, M. Guizani, C. Chen, and J. Bu, “An enhanced public key infrastructure to secure smart grid wireless communication networks,” *IEEE Netw.*, vol. 28, no. 1, pp. 10–16, Jan./Feb. 2014.

[20] F. Cleveland, “IEC TC57 security standards for the power systems information infrastructure—Beyond simple encryption,” in *Proc. Transmiss. Distrib. Conf. Exhib.*, 2005, vol. 2006, pp. 1079–1087.

[21] Wikipedia Contributors, “Knapsack problem,” 2019. Accessed: Jul. 4, 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Knapsack_problem&oldid=90374 8529

[22] Wikipedia Contributors, “Additive increase/multiplicative decrease,” 2018. Accessed: Jul. 7, 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Additive_increase/multiplica tive_decrease&oldid=873175398

[23] Wikipedia Contributors, “TCP congestion control,” 2019. Accessed: Jul. 5, 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=TCP_congestion_control&oldid =904800758

[24] D. Della Giustina *et al.*, “Smart grid automation based on IEC 61850: An experimental characterization,” *IEEE Trans. Instrum. Meas.*, vol. 64, no. 8, pp. 2055–2063, Aug. 2015.

[25] X. Lu, W. Wang, and J. Ma, “Authentication and integrity in the smart grid: An empirical study in substation automation systems,” *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 6, 2012, Art. no. 175262.

[26] “VPN software solutions and services for business.” [Online]. Available: https://openvpn.net/

[27] “dpkt.” [Online]. Available: https://dpkt.readthedocs.io/en/latest/

[28] J. D. P. Rao, M. S. Rai, and B. Narain, “A study of network attacks and features of secure protocols,” *Res. J. Eng. Technol.*, vol. 8, no. 1, pp. 4–8, 2017.

[29] A. Turner and M. Bing, “Tcpreplay: Pcap editing and replay tools for* nix,” 2005. [Online]. Available: http://tcpreplay.sourceforge.net

[30] S. Floyd, “Inappropriate TCP resets considered harmful,” Internet Eng. Task Force, Fremont, CA, USA, RFC 3360, Aug. 2002.

[31] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, “Smart-grid security issues,” *IEEE Secur. Privacy*, vol. 8, no. 1, pp. 81–85, Jan./Feb. 2010.

[32] R. Schlegel, S. Obermeier, and J. Schneider, “Assessing the security of IEC 62351,” in *Proc. 3rd Int. Symp. ICS SCADA Cyber Secur. Res.*, 2015, pp. 11–19.

[33] M. Merabti, B. Alohali, and K. Kifayat, “A new key management scheme based on smart grid requirements,” in *Proc. 9th Int. Conf. Comput. Eng. Appl.*, Dubai, United Arab Emirates, 2015, pp. 22–24.

[34] F. Ye, Y. Qian, and R. Q. Hu, “A security protocol for advanced metering infrastructure in smart grid,” in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 649–654.

[35] T. Liu *et al.*, “A dynamic secret-based encryption scheme for smart grid wireless communication,” *IEEE Trans. Smart Grid*, vol. 5, no. 3, pp. 1175–1182, May 2014.

[36] J. Wu, S. Luo, S. Wang, and H. Wang, “NLES: A novel lifetime extension scheme for safety-critical cyber-physical systems using SDN and NFV,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2463–2475, Apr. 2019.

[37] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, “Decentralized cloud-SDN architecture in smart grid: A dynamic pricing model,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 3, pp. 1220–1231, Mar. 2018.

[38] G. Li, J. Wu, L. Guo, J. Li, and H. Wang, “SDN based dynamic and autonomous bandwidth allocation as ACSI services of IEC61850 communications in smart grid,” in *Proc. IEEE Smart Energy Grid Eng.*, 2016, pp. 342–346.

[39] H. Rahmani, E. Sundararajan, Z. M. Ali, and A. M. Zin, “Encryption as a service (EAAS) as a solution for cryptography in cloud,” *Procedia Technol.*, vol. 11, pp. 1202–1210, 2013.

[40] F. Y. Okay and S. Ozdemir, “A fog computing based smart grid model,” in *Proc. Int. Symp. Netw., Comput. Commun.*, 2016, pp. 1–6.

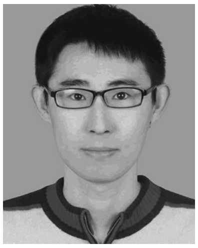
[41] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, “Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4702–4711, Oct. 2018.

[42] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, and J. Hu, “APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT,” *J. Netw. Comput. Appl.*, vol. 125, pp. 82–92, 2019.

Hua Zhang received the B.S. degree in communication engineering and the M.S. degree in cryptography from Xidian University, Xi’an, China, in 2002 and 2005, respectively, and the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2008.

She is currently an Associate Professor of Cyber Security with BUPT. In 2002, she participated in the research of cryptography. In 2005, she joined the Network Security Research Center (NSRC). Her current research interests focus on smart grids security, network security, cryptographic application, and privacy preserve.

Prof. Zhang is a member of the Chinese Association for Cryptologic Research (CACR).



Boqin Qin was born in Shandong, China, on May 24, 1992. He received the B.E. degree in network engineering from the Computer Science School, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. He is currently working toward the Ph.D. degree with the State Key Laboratory of Networking and Switching Technology (SKLNST), BUPT, China, where he has been since 2014.

His current research interests focus on smart grids security, network security, and program analysis.

Tengfei Tu received the B.S. degree in software engineering from Qingdao University, Qingdao, China, in 2013, and the Ph.D. degree in computer science and technology from the State Key Laboratory of Networking and Switching Technology (SKLNST), Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2019.

He is currently a Postdoctoral Researcher with the Institute of Network Technology, BUPT, China. His current research interests focus on web security, mobile security, and cloud computing security.

Ziqing Guo was born in Lanzhou, China, on May 1, 1991. He received the B.S. degree in mathematics and the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2013 and 2019, respectively.

During the Ph.D. studies, he was with the Network Security Research Center (NSRC), State Key Laboratory of Networking and Switching Technology (SKLNST), BUPT. His current research interests include applied cryptography, information security, information retrieval, and data mining.

Fei Gao was born in Shijiazhuang, China, on January 23, 1980. He received the B.E. degree in communication engineering and Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2002 and 2007, respectively.

He is a Professor of Cyber Security with the State Key Laboratory of Networking and Switching Technology (SKL-NST), BUPT. In 2002, he joined the Network Security Research Center (NSRC), State Key Laboratory of Networking and Switching Technology (SKLNST), Beijing, China, where he participated in the research on quantum cryptography and quantum information. His current research interests include quantum position verification and the applications of quantum key distribution.

Prof. Gao is a member of the Chinese Association for Cryptologic Research (CACR).

Qiaoyan Wen was born in Xi’an, China, on July 27, 1959. She received the B.S. and M.S. degrees in mathematics from Shaanxi Normal University, Xi’an, China, in 1981 and 1984, respectively, and the Ph.D. degree in cryptography from Xidian University, Xi’an, China, in 1997.

She is a Professor of cyber security with the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, and the Founder and Leader of the Network Security Research Center (NSRC), State Key Laboratory of Networking and Switching Technology (SKLNST), Beijing, China. Her current research interests include cryptography, information security, Internet security, and applied mathematics.

Prof. Wen is a Senior Member of the Chinese Association for Cryptologic Research (CACR).