SeReCP: A Secure and Reliable Communication Platform for the Smart Grid

Kubilay Demir and Neeraj Suri
{kubidem, suri}@cs.tu-darmstadt.de, TU Darmstadt, Germany

***Abstract*—The management of a complex cyber-physical system such as the Smart Grid (SG) requires responsive, scalable and high-bandwidth communication, which is often beyond the capabilities of the classical closed communication networks of the power grid. Consequently, the use of scalable public IP-based networks is increasingly being advocated. However, a direct consequence of the use of public networks is the exposure of the SG to varied reliability/security risks, e.g., distributed denial-of-service (DDoS). Thus the need exists for new lightweight mechanisms that can provide both cost-effective communication along with proactive DDoS attack protection. We fill this gap by proposing a novel approach termed as SeReCP, which leverages: (1) a semi-trusted P2P-based publish-subscribe (pub-sub) system providing a proactive countermeasure for DDoS attacks and secure group communications by aid of a group key management system, (2) a data diffusion mechanism that sustains the network availability in the case of both randomly sweeping and targeted DDoS attacks on pub-sub brokers, and (3) a multi-homing-based fast recovery mechanism for detecting and requesting the dropped packets, thus paving the way for meeting the stringent latency requirements of SG applications. Our evaluation on a real testbed demonstrates that SeReCP provides the required security and availability for SG applications with up to 30% failures of the pub-sub brokers. Overall, we show that SeReCP helps enable the secure use of public network based communication for safety-critical cyber-physical systems such as the SG.**

***Keywords*- Availability, Security, pub-sub, Key management**

I. Introduction

The traditional power grid is evolving into the SG to incorporate heterogeneous and geographically distributed energy sources for overall cost-effective power generation and distribution. However, the penetration of the distributed power generation sources into the power distribution network causes two-way (production & distribution) electricity flows. This entails active management of the distribution networks, which were typically designed to support only one-way power distribution. Consequently, the SG requires a scalable and efficient communication network that can facilitate the interactions for this complex cyber-physical system linking the communication, computation and control actions across the utility company areas and the SG elements.

To provide this, utilities utilize management systems such as wide area monitoring, protection and control (WAMPAC) and advanced distribution automation (ADA) among others, and communication networks, which enable acquisition of real-time sub-second measurements across the grid. WAMPAC and ADA (and other SG apps) need to collect/deliver large amounts of data with latency needs of 100ms-5sec and availability/reliability needs of 99.00%-99.999% [10] [8].

To monitor and control the power grid, the utilities currently employ proprietary and closed automation networks. However, these networks invariably encounter scalability issues to deal with the (a) increasingly large and *ad hoc* SG structure, and (b) large data traffic produced by the thousands of SG devices. As a result, the grid requires a flexible and scalable network that can provide low-latency, high-availability, secure and reliable communication. While an ideal solution would be a dedicated network, the cost-based reality results in the use of IP-based public networks such as the Internet. The caveat is of inheriting the Internet's reliability risks and security vulnerabilities that can be exploited by hackers causing security and safety risks for not only the cyber-system but also for the physical-systems, e.g., electrical grid/appliances.

In addition, compared with classical IT (Internet-Technology) systems, the SG's security and reliability requirements differ as (1) IT security typically focuses on server-side protection versus client-side while the SG needs to factor client-side (SG generators, devices) vulnerability, and (2) unlike the best-effort delivery in the Internet, the SG data traffic requires explicit assurance on timely delivery of information. In addition, most of the SG devices have constrained computational capacity and group communication requirements unlike the IT systems. Hence, many current IT security approaches cannot be easily deployed on the constrained devices of the SG.

Hence, the need is for SG communication networks to have lightweight security mechanisms for preventive/proactive defenses to DDoS attacks in the SG's distributed and composite communication-control cyber-physical environment. As pub-sub approaches inherently provide a proactive DDoS attack protection, a number of approached based on them have been proposed for the SG. GridStat [1] employs a pub-sub system and long-term security key pads to provide a secure and scalable communication between the parties. However, long-term security keys can potentially introduce severe security vulnerabilities e.g., compromised keys can be distributed to a large number of zombies to access/attack the network. SeDAX [2] also introduces a pub-sub system, which contains trusted authentication servers allowing the parties to periodically obtain topic-based group keys to assure end-to-end (E2E) confidentiality and integrity. However, SeDAX does not introduce any authentication mechanism between the publisher and pub-sub brokers, and this paves the way for DDoS attacks against pub-sub nodes. Moreover, none of the existing works [1,2,4,12,13,14,15] focus on addressing the high availability requirements of the SG devices/data traffic in case of a targeted

Research supported in part by EC H2020 CIPSEC GA 700378.

978-1-5090-5652-1/17/$31.00 2017 IEEE

or blindly sweeping DDoS attack against pub-sub brokers to sustain communication between the critical SG entities.

A. SeReCP: Concepts and Evaluation

SeReCP introduces a novel pub-sub-based proactive DDoS attack defense mechanism in addition to its being a lightweight security mechanism. In SeReCP, taking into account the requirements for SG data traffic, device resources and security, we propose a pub-sub system proactively countering DDoS attacks that cannot be handled by the constrained SG devices. However, targeted or blindly sweeping DDoS attacks against pub-sub brokers can easily render inaccessible some of the critical devices to pose safety risks. To address this issue, we employ a data diffusion approach which enables spreading the data packets across the pub-sub brokers using its token-based stateless authentication mechanism. Moreover, to account for the stringent availability and latency requirements of SG applications, we propose a multihoming-based fast "recovery" mechanism. We transmit every two consecutive data packets to two different network interfaces of a (randomly) selected pub-sub broker. If one of the network interface is under attack, the broker requests a missing packet after a relatively short waiting time using the remaining functional network interface. This allows for fast packet "recovery" compared to classical ACK-based mechanisms such as TCP's cumulative ACK. On the other hand, to protect end-to-end (E2E) confidentiality and integrity of the data, we propose a group key management system, which provides role-based access rights for both publisher and subscriber in addition to protection from replay attacks.

We evaluate our approach assessing: (1) network availability for SG applications over targeted or blindly sweeping DDoS attacks on the pub-sub brokers.For the SG, availability is not only successful data delivery, but also a delivery meeting each application's latency requirements (2) overheads in terms of resource usage and additional transmission delay produced by the proposed security mechanism. The results show that SeReCP introduces an acceptably low latency overhead of 40 ms. In addition, SeReCP provides the required availability/reliability[1] for up to 50% failure of pub-sub brokers by transmitting duplicate packets. We compare our system with the reference work of Angelos et al. [4], which also employs data diffusing mechanisms for real-time applications. [4] shows stable performance for up to 5% of pub-sub brokers being attacked. In contrast, SeReCP shows stable performance for up to 30% of pub-sub nodes being compromised without the use of duplicated packet transmission. These demonstrate SeRECP's highly promising capability to effectively build safety critical SG applications utilizing public networks.

B. Contributions

(1) We define the security requirements and threats for the SG. Based on this, we propose a novel pub-sub approach, which provides secure/reliable communication in case of DDoS attacks and for link/node failures.

(2) Considering the high availability requirements of the SG traffic, we propose a multihoming-based fast "recovery" mechanism in addition to the data diffusion approach, which provides minimum drop/ack/re-transmission over attacks on the intermediate pub-sub brokers.

(3) Given the constraints of SG devices and also for their group communication requirements, we introduce a novel group key management mechanism, which provides replay and repudiation attack protection in addition to confidentiality and integrity assurance.

(4) The evaluation of SeReCP is performed on a real test-bed NorNet [5], providing multihomed nodes distributed all over Norway. The evaluation validates the effectiveness of SeReCP in terms of availability under the attack and for its low overhead.

The remainder of the paper is organized as follows. Section 2 introduces the SG security requirements. Section 3 details the background, security and attack models, followed by Section 4 which develops the SeReCP approach. Section 5 conducts the security analysis for SeReCP followed by its evaluation in Section 6. We present the related work in Section 7.

II. SG Network and Security Requirements

Traditionally, power grid communication systems have been physically isolated from public networks. This has been changing due to the cost effectiveness of utilizing public networks and the technical features offered by them for bandwidth, latency, stability and availability. While decreasing the cost of operation, employing public networks naturally makes the power grids vulnerable to cyber attacks. We survey some differences of SG communication security requirements from classical IT systems (e.g., Internet, Web), and introduce the SeReCP approach to address the corresponding requirements.

A. SG security requirements

In SG communication networks, the security objective is to defend the data from unauthorized acts with the prioritized concerns (driven by safety implications) being: 1) data availability, 2) data integrity, and 3) data confidentiality.

For availability requirements, SG applications require timely and reliable access to information. Lossy or delayed information can result in an inaccurate system state estimation with consequent incorrect control decisions resulting in damage to the grid. For integrity, the unauthorized modification of information can result in wrong decisions on power management. For confidentiality, to protect personal and proprietary information, the need is to prevent unauthorized information access and disclosure. For system reliability, confidentiality might not be critical, yet for systems involving interactions with customers, such as demand response and advanced metering infrastructure (AMI) applications, it is important.

A unicast delivery of a time-critical command by a constrained SG device to each of the entities inevitably results in large delay and potential for damages to power equipment. The more efficient approach is multicast to deliver a time-critical message to all related entities belonging to the same group. Hence, authentication/confidentiality schemes for SG security must be able to efficiently support multicast communication (*Requirement 1*).

[1] Availability and reliability of SG communication network are interchangeable [8].

B. Differences from typical IT security

IT-based cyber security solutions, e.g, firewalls, intrusion detection systems (IDS), and Virtual Private Networks (VPN), are known to be effective in securing the IT infrastructure. However, the resource constraints (computational, memory and bandwidth) of SG devices often preclude the direct applicability of such IT solutions.

In a typical IT system, the application servers are often more secure than the edge/client nodes. In SG networks, the edges requires the same level of security as the control center servers, as the edge devices (such as relays, circuit breakers,...) can cause harm to human life, damage equipment or power lines. Furthermore, SG communication nodes offer limited functionality given their resource constraints. Hence, directly employing sophisticated IT-based DDoS defense/authentication mechanisms has limited applicability to the SG resulting in the need for lightweight and proactive DDoS protection mechanism to be employed (*Requirement 2*). We advocate broker-based pub-sub systems to provide for proactive DDoS mechanisms, as well as multicast communication.

In the case of failures in IT networks, a simple solution might be just rebooting a node or an application. However, in many SG control applications, this is not admissible from a control stability viewpoint. Moreover, the DDoS attacks leading to violation of the timing requirements or loss of control messages data can result in imbalance of the grid. Therefore, SG communication networks are required to avoid single-point-failures regarding physical network infrastructure, routing protocol and security mechanisms (*Requirement 3*). To cope with this, we introduce a data diffusion approach enabling delivery of the scattered data packets over multipath, thus ensuring minimum packet drop in the case of pub-sub broker failures. In addition, we propose multihoming based fast ”recovery” mechanism in order to resend the dropped packets. To address authentication, the use of high-overhead public key based authentication is of limited usability in the resource-constrained SG devices (*Requirement 4*). Therefore, we propose a token-base mechanism providing a stateless light-weight authentication between brokers and publishers in addition to an efficient group key management system for E2E security.

III. Goals, Models and Assumptions

A. Security Goals

Our security goal is to ensure delivery of the publishers' data to the corresponding subscribers within the deadline[2], stipulated in the application requirements. To achieve this, DDoS attacks must be proactively prevented or at least mitigated to meet the high availability requirements of the critical devices.

Moreover, any lossy or outdated data needs to be detected by the brokers and subscribers. The origin of data should be identifiable within the group communication paradigm. Also data requiring confidentiality can be decrypted by the corresponding subscriber but neither by the broker nor by intruders.

B. Reference Pub-Sub Model

We consider that the utility employs a combined network (i.e., public and private), taking into account the applications' availability requirements and cost-effectiveness. To deal with the complexity of this heterogeneous network, we take advantage of the SeReCP middleware, which provides a scalable QoS-aware pub-sub system. This P2P-based pub-sub system selects the ”strongest” nodes (in terms of computation capacity, multihoming feature and trust-level), as brokers. These brokers are clustered depending on their autonomous system (AS) and geographical proximity to obtain the network state information in a scalable probing overhead (Fig. 1). SeReCP is devised for a messaging paradigm where, upon reception of a publication from a publisher, the broker transmits the data, considering the QoS requirement of the application and the network state information, to the brokers responsible for delivery the data to the corresponding subscribers.

The main role of SeReCP in combating DDoS attacks is to distinguish between authorized and unauthorized traffic and then to either enable the traffic to access to the destination or to drop/filter it, respectively. Thus, the system provides the functionality of a firewall scattered over the wide area network to prevent any congested link to the target(s). SeReCP leverages some existing approaches to provide this. Examples being QoS-aware robust overlay network [3], pub-sub platform for smart grid [1], Overlay-based DDoS attack defense mechanism [4]. However, considering the SG security threats associated with the network model, we introduce a new advanced attack model and a mechanism, SeReCP, which counters these attacks in addition to covering scalability and QoS issues.

C. Perturbation/Attack Model

An adversary, whose aim is to render critical devices (publishers) inaccessible, can mount a DDoS attack against either subscribers through broker(s) or directly broker(s) that maintain the communication between those publishers and the corresponding subscribers. The attacks can also be mounted for a short time, which force the peers to reset their communication as well as authentication. This introduces an unacceptable loss of availability for the critical applications.

On the other hand, for applications requiring high availability and low latency, the accidental failure of broker(s) providing connection between a publisher and its subscribers might pose safety-risks as, until new connection established over new broker(s), some of the critical node might be inaccessible.

For critical applications, replay and repudiation attack can pose high risk e.g., receiving an outdated measurements can result in a wrong decision for field devices. An adversary can obtain an elevated privilege by compromising some secrets to resend/delay some of the data.

We consider a strong threat model where (1) An adversary can have access rights to some underlay routers to eavesdrop, capture, drop, resend, and alter to accomplish an replay or DDoS attack against brokers. Exploiting the obtained elevated privilege, a large amount of zombies can launch an attack brokers to deny the service. (2) The secrets of some publishers, subscribers and brokers can be compromised by the intruders to attack brokers and (if attacker gains a right to pass authentication of the brokers) subscribers.

[2] Maximum acceptable latency in the message delivery

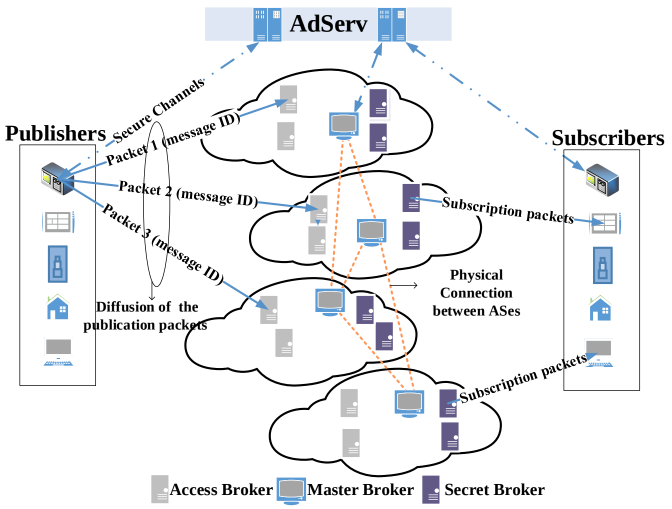


Fig. 1: After obtaining the ticket from AdServ by using the secure channel, the publisher diffuses the data packets over N access brokers. Access brokers check the authenticator and hand them over secret broker(s) to check the validity of the ticket and distribute to the subscribers.

D. Assumptions

We assume that all publishers/subscribers/brokers cannot be fully compromised, and only the secrets held by the minority of them can be compromised. In addition, we do not consider an intruder to compromise the publishers' secrets for rights increase of the publication rate to launch a DDoS/DoS attack against the subscribers. We assume such attacks to be handled by existing reactive measures e.g., filtering policy. We consider that publishers know only the access brokers IP addresses but not each others.

We assume that all nodes have valid certificates issued by a Certification Authority and that each node's certification is delivered to Authorization Servers in a secure way. For devices that do not possess enough resources for asymmetric-key cryptography, we employ physical unclonable functions (PUF), combining symmetric-key and ID-based key cryptosystems [6]. This assumption is reasonable [1] considering the SG's relatively static communication node structure.

IV. Developing the SeReCP Mechanisms

SeReCP's main goal is to provide (a) a proactive DDoS attack protection by differentiating authorized/unauthorized traffic at the resource-rich broker nodes than being handled at the constrained subscriber nodes, and (b) a secure and reliable communication. SeReCP is designed for a messaging paradigm providing E2E timely delivery guaranty using a lightweight pub-sub paradigm extending on [1] instead of data storage/querying [2] or complex event processing.

The main components of SeReCP are as follows (Fig. 1):

Publishers: The devices that produce the data, which is required for the subscribers. This data can be a measurement for some applications or a command for some actuators. This data is signed and (if necessary) encrypted by the publisher.

Subscriber: The entity that needs the data to decide or actuate depending on the application context.

Administration servers (AdServ): We consider these servers as trusted and robust for the DDoS attacks. They have three roles in the system: (1) Bootstrap node (2) Authorization server (3) Pub-sub system administrator. AdServ maintains all certifications of the nodes. When the nodes apply to AdServ to access the network, they obtain the security keys and other information regarding their respective role in the network by using a secure channel (public key or PUF).

Brokers: There are three type of brokers: (1) Access Brokers (AB), receiving the publications from publishers and validating their authenticators. (2) Secret brokers, transmitting the publications to the corresponding subscribers after verifying their tickets' authenticity (3) Master brokers, responsible for the probing to the other clusters/master brokers and maintaining the network state.

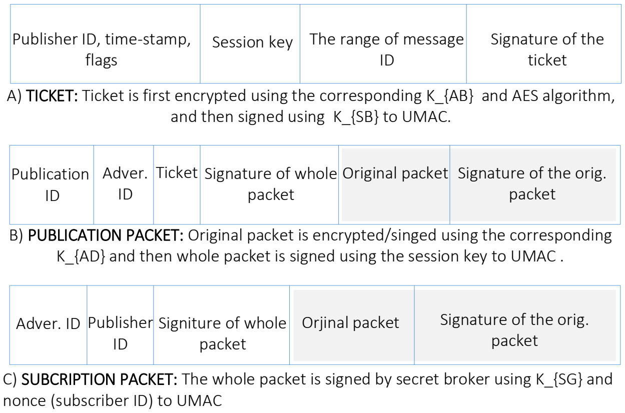
After clustering all nodes according to their respective Autonomous System (AS) and geographical proximity (Fig. 1), AdServ dynamically chooses the "strongest" nodes as brokers for each cluster (versus employing dedicated brokers) in order to provide scalability. AdServ assigns the roles as publisher/subscriber/broker to each node. Every node in the network takes part as a publisher or/and subscriber or/and broker at the same time. AdServ informs the corresponding brokers about the publishers' advertisements[3] and their corresponding subscribers. The access broker IP addresses/IDs are also delivered to publishers in the initialization process by AdServ (the changes in the broker list are also delivered by AdServ, and we consider this to happen only infrequently).

SeReCP provides security in two steps. (1) From publishers to the brokers (2) E2E (publishers to subscribers). We focus on the authentication between publishers and access brokers in order to avoid the DDoS attacks.

A. Authentication/Communication protocol between publishers and access brokers

In the construction of our authentication protocol between the publisher and brokers, we leverage Angelos, et al. [4]'s approach due to its high resilience against the DDoS attack. However, [4] is based on a strong and potentially unrealistic assumption of fully trustworthy brokers, and also assumes that the shared key encrypting the ticket cannot be obtained by an adversary in any way. An adversary who compromises the shared key can then mount a severe attack on the target (e.g., subscribers). Our approach removes these significant constraints. In case of using the connection maintaining application or network level state, the connection can be forced to reset by even short-time DDoS attack against access brokers. The loss of availability can cause some disturbance for the critical SG applications. To cope with this, SeReCP employs a token-based authentication (similar to a Kerberos), which alleviates the necessity of application state at the brokers. Taking advantage of this "stateless" authentication, publishers diffuse the data packets over N access brokers. In case of d

[3] Advertisements are type of publications and each of them has a group of subscribers. Each publisher can publish one or multiple advertisements. We assume that these assignments are managed by AdServ.



Publisher ID, time-stamp, flags	Session key	The range of message ID	Signature of the ticket

A) **TICKET:** Ticket is first encrypted using the corresponding K_{AB} and AES algorithm, and then signed using K_{SB} to UMAC.

Publication ID	Adver. ID	Ticket	Signature of whole packet	Original packet	Signature of the orig. packet

B) **PUBLICATION PACKET:** Original packet is encrypted/singed using the corresponding K_{AD} and then whole packet is signed using the session key to UMAC .

Adver. ID	Publisher ID	Signiture of whole packet	Orjinal packet	Signature of the orig. packet

C) **SUBCRIPTION PACKET:** The whole packet is signed by secret broker using K_{SG} and nonce (subscriber ID) to UMAC

Fig. 2: Ticket and packet structures

percentage of N access brokers deny service due to the DDoS attacks (e.g., N = 100, d = 10), the dropped data can be checked/corrected in the subscribers by using forward error correction (FEC) or transmitting redundant replicated packets from the publisher (a acknowledge mechanism can also be used for applications requiring relatively lower availability).

Key and ticket establishment: When a publisher applies to AdServ using the secure channel in order to join the network, AdServ delivers three type of secrets to the publisher: (1) secret key and initial sequence number for their each advertisement (the subscriber groups also obtain the corresponding secret key and sequence number), (2) a session key, a 128 bit symmetric key, and (3) S (= number of access broker/number of the clusters) tickets valid for the corresponding access brokers. These secrets can be regularly updated depending on the criticality of the applications running on publishers[4]. Table 1 summarizes the keys used in SeReCP.

Fig. 2(A) illustrates the ticket consisting of a session key, a publisher ID, a range of publication ID numbers, a time-stamp, flags indicating ticket features, and signature of the ticket. While the tickets are signed using a shared key K_S to UMAC [11], distributed to all secret brokers by AdServ, they are encrypted using the corresponding access broker's key K_A.

A packet sent to an access broker contains five parts, as shown in Fig. 2(B): (1) the publication ID, a monotonically increasing number encrypted using the session key (2) advertisement ID (3) the corresponding ticket (4) signature of the whole packet, derived using the session key (5) original packet, encrypted and signed using a key, K_O, derived using secret key and sequence number of the advertisement as inputs to a pseudo random function (PRF), more details in the section 4-B.

Communication Protocol between Publisher and Access Brokers: When a publisher obtains the three secrets, it is ready to publish its data over access brokers. To transmit each packet, an access broker is selected in a pseudo-random manner. The selection is performed for each packet by using the last 4 digits of a random number, derived using the session key and the publication ID as inputs to pseudo random generator (PRG), as an index to the list of access brokers. For each subsequent packet, the publication ID regularly increases, thus diffusing the packets over access brokers in a pseudo-random manner.

Once the access broker receives the packet from the publisher, it obtains the session key by decrypting the ticket of receiving packet using its respective K_A, and validates the packets signature using the session key to UMAC. This provides a packet validation with low computation and also protects from computational DDoS attacks. After the validation, the packet's publication ID (as decrypted using the session key) is checked to be within the acceptable range defined in the ticket and is larger than the last seen publication ID. The publisher ID and the last seen publication ID are only things stored by access brokers. In addition, the access broker validates whether the packet is routed to the correct access broker. To do this, it matches the respective access broker ID and the last four digit of the random number (derived by using the session key and publication ID along with PRG). These checks provide a strong replay and repudiation attack protection with minimum memory occupation in access brokers. Finally, the access broker hands the packet, containing the ticket in a decrypted form, over the corresponding secret brokers[5].

The ticket verification is delegated to the secret brokers to perform the authentication in separate nodes. The validity of the ticket is checked by fulfilling a UMAC validation using a shared key K_S in secret broker(s). Moreover, publication ID in the packet and the last seen publication IDs[6] are compared whether there is a abnormal difference, thus dropping the packets fabricated with a random publication ID by a intruder compromising some of the access broker keys, K_A and the shared key, K_S.

Re-keying procedure: Ticket usage is restricted by the range of publication ID numbers and the time-stamp, i.e., 500 packets and 1-2 hours, thus avoiding reuse of the ticket by multiple zombies. Once a secret broker notices the ticket in the receiving packet is about to expire in terms of either the time or the range of publication ID numbers, it produces a new ticket by enlarging the range of publication ID number and signing using K_S. Then, the secret broker sends new ticket to S access broker (randomly selecting one from each cluster). The access brokers issue new tickets to publishers after encrypting it using its respective K_A. However, in the re-keying process, the session keys of publishers are not changed. This can only be fulfilled by AdServ and we consider this done using the secure channel depending on the application criticality e.g., hourly, daily.

Multihoming based fast "recovery": SG applications have stringent latency requirements as delayed/lost messages could result in improper control operations. Taking these requirements into account, we propose multihoming based fast "recovery" mechanism, enhancing the approach of [4] by detecting/re-transmitting the dropped packets in a timely manner. To achieve this, we redesign the data diffusion mechanism by enabling publishers to forward every two consecutive

[4] The tickets can be more frequently updated by secret brokers but with the same session key. In the AdServ's update, the session key can be updated.

[5] AdServ constructs a hash table, mapping advertisement IDs with their corresponding secret broker ID, and then issues this list to all access brokers

[6] Secret brokers maintain publisher ID and the last seen S publication IDs for themselves as well.

TABLE I: The keys used by SeReCP

Keys	**Usage**
Session Key	Signing/verifying the publication packets in publishers and access brokers, respectively
K_O	Signing/verifying and encrypting/decrypting the original packet in publisher/subscribers, respectively
K_A	Encryption/decryption of the ticket in the corresponding access brokers
K_S	Verification of the ticket in the secret brokers
$SecretKey_i$	Input to PRF to derivate K_O
K_G	Signing/verifying the subscription packet in secret broker/subscriber, respectively

data packets to two different network interfaces of a (pseudo) randomly selected pub-sub broker. The access broker sets a timer on receiving one of the messages to check whether the other message arrives within the stipulated time period. If not, it requests the dropped message using the remaining functional network interface. We consider that the knowledge about which IP addresses belongs to the same access brokers is maintained by only publishers but not by public. Our experiments show that this mechanism provides high mitigation for delivery of the dropped packets without violating the defined maximum latency of the applications with high availability and latency requirement in case of even 30% of the access broker coming under the DDoS attack. Furthermore, this mechanism does not introduce any additional overhead when there is no attack unlike replicated data delivery or FEC, and provides much lower latency compared to classical acknowledge mechanisms e.g., TCP's cumulative ACK mechanism.

How does SeReCP overcome the shortcomings of [4]: To address the limitation of [4] of full trust in brokers, we develop a novel solution as: (1) SeReCP employs S tickets encrypted by S different K_A rather than a ticket encrypted by a shared key. (2) The authentication in SeReCP is fulfilled in two separate brokers having different keys to check the ticket's diverse parts.

In SeReCP, an adversary whose aim is to launch a DDoS attack against the subscribers needs to compromise either all of S tickets maintained by publishers (this attack takes until ticket expire i.e., 500 packets and its results is relatively limited) or both all of K_A, and K_S kept by the corresponding access brokers and secret brokers, respectively. In case of an attacker compromises only some of S K_A and K_S, the secret brokers can suspect the packets to be fabricated by using compromised keys since their publication IDs are quite different from others. Thus, SeReCP renders a DDoS attack against the subscriber to be much harder to launch.

We employ a multihoming based fast "recovery" mechanism to meet high availability requirements without requiring replicated data delivery as in [4]. Our experiments demonstrates that transmitting a replicated packet for only applications with high availability requirement among the others, SeReCP meets the applications' availability requirements in case of even 50% access broker under attack.

To provide secure communication between the brokers we consider a symmetric key pad similar to [1]. However, new pads can be regularly issued by AdServ using the secure channel unlike [1]. This produces limited overhead, since we employ this method only between brokers but not across all nodes as in [1].

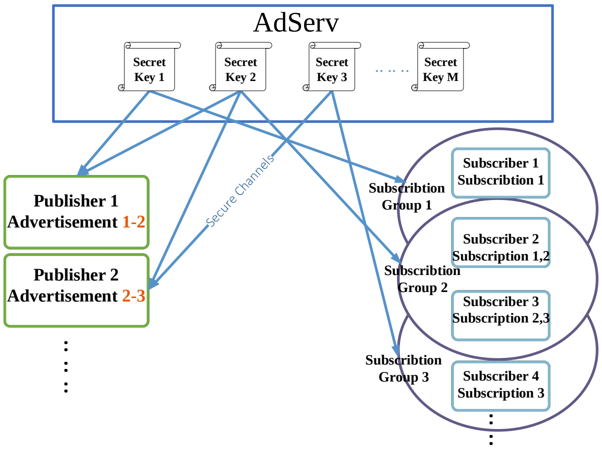

AdServ
Secret Key 1
Secret Key 2
Secret Key 3
Secret Key M
Secure Channels
Publisher 1 Advertisement 1-2
Publisher 2 Advertisement 2-3
Subscribtion Group 1
Subscriber 1 Subscribtion 1
Subscribtion Group 2
Subscriber 2 Subscription 1,2
Subscriber 3 Subscription 2,3
Subscribtion Group 3
Subscriber 4 Subscription 3

Fig. 3: Secret Key Distribution

B. E2E security and group key management

Group key management system: SeReCP provides efficient DDoS protection by employing the above methods. However, to protect the E2E integrity and confidentiality between publisher and their subscribers, SeReCP requires a key management system. Hence, we introduce a group key management system (done by AdServ), which first identifies the advertisements of each publisher e.g., advertisements 1 and 2 for publisher 1 according to the application running on the device and the utility policy, and then issues the corresponding secret keys for the publishers, e.g., secret keys 1 and 2 for publisher 1. Additionally, AdServ clusters subscribers according to the advertisements they subscribe, e.g., subscribers 1 and 2 subscribe for advertisement 1 (a subscriber can also subscribe to multiple advertisements), and then issues the secret keys to the corresponding subscription group members e.g., secret key 1 for subscription group 1 (subscribers 1 and 2). This process is illustrated in Fig. 3.

E2E security: Malicious intruders (during the transmission over the underlay network) can alter the data to launch a replay or repudiation attack against the subscribers. To cope with this, the publishers encrypt and sign each packet with K_O:

$$K_O = f(SecretKey_i, SequenceNumber)$$

where f indicates PRF, $SecretKey_i$ is a symmetric key issued by AdServ for advertisement$_i$, and $SequenceNumber$ is the current sequence number of the packet of the advertisement$_i$. The subscribers use the pair of publisher ID and advertisement ID as index to maintain the last seen sequence number in a list. Upon reception of a packet, using the publisher and advertisement IDs as index in the list, the last seen sequence number is obtained and then $SequenceNumber$ is calculated by adding 1 to it. The subscriber makes use of $SequenceNumber$ and $SecretKey_i$ to PRF in the process of derivation of the K_O.

The subscribers can authenticate the original packet performing a UMAC validation using K_O. In case of the packet is forwarded by an intruder to a subscriber not in the correct group, the packet cannot be decrypted by the subscriber without K_0. In addition, since K_O is derived using the current $SequenceNumber$, launching a replay attack is impossible for intruders if they have no the $SecretKey_i$ and the $SequenceNumber$.

On the other hand, upon receiving a packet from an access broker, the secret broker extracts the ticket and publication ID, but add the publisher ID into the subscription packet. To protect from repudiation attack, secret brokers sign the subscription packet using K_G and a nonce (subscribers' respective ID) to UMAC as illustrated in Fig 2(C). K_G is issued by AdServ to secret brokers and the corresponding subscription group (it does not need to happen do frequently e.g., daily). Thus, even if intruders compromise a subscriber's secrets ($SecretKey_i$, $SequenceNumber$, and K_G), they cannot fabricate a packet for the other subscribers by spoofing the IP addresses (as if it is coming from the publisher over the secret brokers), since the intruder must know subscriber IDs (a random 32 bit value) of the target subscribers to embody the UMAC signature.

To alter a packet, an intruder would need to compromise (a) all secrets of the secret brokers, (b) members of that subscription group, and (c) obtain detailed knowledge of the network/nodes structure and data flow relationships. This is often unrealistic.

V. Security Analysis

We now present the security analysis for SeReCP.

A. DDoS Attack

Rendering a publisher inaccessible is possible by launching a DDoS against either access brokers maintaining the communication between publisher and subscriber or the corresponding subscribers by gaining elevated privilege on the intermediate brokers[7]. To cope with direct attacks on the brokers, SeReCP employs packet diffusing in addition to redundant packet transmission or FEC, thus providing a significant mitigation in case of d% brokers under the DDoS attack. Moreover, taking into account the stringent latency and availability requirements, we propose a multihoming based fast ”recovery” mechanism, proving rapid re-transmission of dropped packets. In the second case of attacks, SeReCP employs S different tickets and two step authentication to make a DDoS attack against the subscribers much harder, considering this type of attacks launched by attackers who are able to compromise some secrets of the nodes. Thus, to mount a attack on the subscriber, an adversary must compromise both K_A and K_S belonging to access brokers and secret brokers, respectively. This is a difficult task for an attacker.

B. Replay Attack

Replay attacks pose risks for both brokers and subscribers. If the system is vulnerable to replay attack, an attacker can re-send the same packet in order to mislead brokers and subscriber. However, SeReCP uses two dedicated countermeasures to protect the system from replay attacks. (1) Access brokers keep the last seen publication ID along with the publisher ID. Upon reception of a packet, they check if the publication ID in the receiving packet is larger than the last seen one. (2) Publishers sign the original packet K_O derived by using $SecretKey_i$ and $SequenceNumber$ to PRF. Upon receipt of a packet, the subscribers derives K_O by using $SecretKey_i$ and $SequenceNumber$ to PRF. Then, the signature is checked using K_O whether the packet is altered/fresh, or not.

C. Repudiation Attack

Repudiation attacks are a severe concern over group communication since an adversary who compromised the secrets of a member of a subscription group can fabricate packets by spoofing the publisher's IP address. SeReCP employs a K_G for each subscription group. AdServ issues these keys to the secret brokers and the corresponding subscribers. They also sign the packets using the subscriber ID (as nonce) and K_G to UMAC. This provides a capability for subscriber to check whether the packet originates from the secret brokers and implicitly the publisher.

D. Drop and delay Attack

An attacker can drop some of the packet to lead to failure of the authentication mechanism. Particularly, the dependence on the monotonically increasing message ID in the system might cause the failures in case of such a attack occurs. To simply mitigate this issue, the subscribers keeps a reasonable number of missing sequence numbers.

VI. Evaluation

The main goal of SeReCP is to provide high communication availability between publishers and subscribers during even the high-volume DDoS attacks in order to avoid inaccessibility to some critical devices, which can pose severe safety risks on power grids. In this section, we present an evaluation of how well SeReCP meets these goals in addition to its additional overhead in terms of latency and traffic.

We consider that the wide area network utilized by the utility to manage the SG, covers a territory or a country. To obtain realistic results we employed NorNet Testbed [5] that contains multihomed[8] nodes spread over entire Norway. We deployed daemons at 30 nodes with 2-3 network connections.

To measure the round-trip time latency between publishers and subscribers (when the brokers interpose), we deploy publisher and subscriber daemons at two different nodes, and broker daemon at the all other nodes. In each experiment we change the tasks of all nodes till the results stabilize. By doing so, we obtain the results across deployments. Correspondingly, for availability measurements, we deploy the publisher and subscriber daemons on the same node in order to enable the implementation of diverse attack scenarios.

Each SeReCP's publication packet (Fig 2(B)) contain 48 bytes of additional data (36 bytes ticket, 4 bytes publication ID, 4 bytes Advertisement ID, and the 4 bytes signature) over the original packet; the subscription packets include only 12 bytes additional data (Advertisement ID, publisher ID, and the signature are 4 bytes). Taking into account SG applications high availability requirements, this additional traffic constitutes a reasonable overhead.

[7] The publishers and subscribers IP addresses are not public and they permit only some predefined IP addresses to communicate. Hence we assume a direct DDoS attack on the publishers and subscribers is not possible.

[8] A multihomed node has connections to multiple Internet service providers (ISP) via different network interfaces.

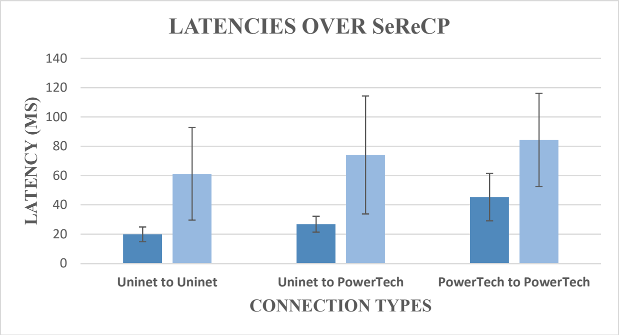


Fig. 4: Latency results between publisher and subscriber as ISP connections. SeReCP introduces only a 40ms additional latency

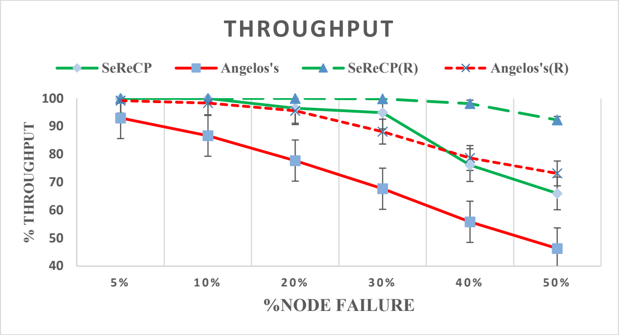


Fig. 5: The normalized throughput results of SeReCP and Angelos[4]. In the case of transmitting replicated (R) packets, both SeReCP(R) and Angelos'(R) achieve higher network resilience.

TABLE II: Performance evaluation parameters

Application	Availability	Latency	Priority
Wide Area Situational Awareness (WASA)	100%	200 ms	high
Real Time Pricing	99.33%	1150 ms	middle
Customer Information	98.50%	2000 ms	low

Fig. 4 shows the round-trip time latency between publishers and subscribers by comparing direct communication using UDP with the communication over the brokers using SeReCP. As we investigate if our approach can be implemented for SG applications using the public networks, we present the actual latency results for both direct communication and SeReCP. In addition, although we obtain the results for each deployment case, since we see that the results mainly differ according to the ISPs (e.g., Uninet, PowerTech) between two ends, we illustrate the three representative combinations. The latency results show that SeReCP adds around 40ms latency in comparison to the direct connection. The ISPs' underlay infrastructures have significant effect on the latency. Although SeReCP introduces an additional 40ms latency, the obtained latency values, between 60-80 ms, are reasonable for most of the SG applications which range at 200-2000 ms as in Table 2 [8].

We first compare our approach with Angelos et al. [4] regarding throughput, in the normalized form, in the presence of DDoS attack. Throughput refers to the rate of successful message delivery. Fig. 5 denotes the successful delivery rates of SeReCP and [4] without/with duplicate packets in the case of different rates of failures in access brokers. Without duplicate packets, we see that while the connection performs well up to 30% failure of pub-sub brokers using SeReCP (recall that real time applications can perform well up to 10% packet drop by using UDP or TCP [7]), the performance drops after 5% failures for [4]. Utilizing our multihoming-based fast "recovery" mechanism, SeReCP delivers the dropped packets in time (i.e., the re-transmission time of the acknowledge mechanism of subscribers). On the other hand, in the case of sending duplicate packets, whereas SeReCP can maintain the connection without stalling up to 50% failure, the connection can perform well up to 20% failure for [4]. These results demonstrate that sending redundant replicated packets significantly enhances the throughput in the presence of DDoS attack. However, even by sending duplicate packets, [4] cannot introduce effectiveness as for simple SeReCP.

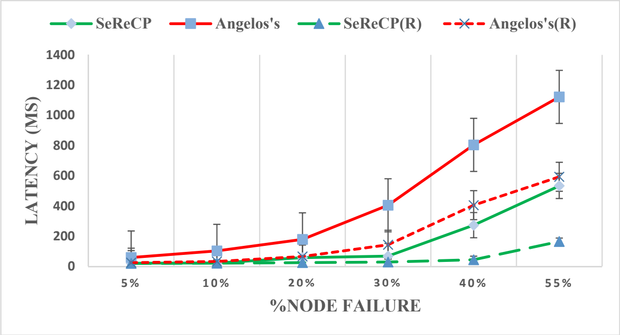


Fig. 6: Latency measurements during the attacks. The transmitting replicated packet helps obtain lower latencies in both approaches

To assess the effectiveness of the approach, an important factor is latency over an DDoS attack. Fig. 6 shows the corresponding latency results for the experiments. By sending duplicate packets SeReCP provides reasonable latencies for the real-time applications[9] for up to 50% failures. [4] with duplicate packets, and similarly simple SeReCP, introduce similar curves and latencies up to 30% failures for the real-time applications.

We next evaluate the communication system to provide the required availability for SG applications. Availability refers to the rate of delivered packets that do not violate the application latency requirements. Therefore, a packet should arrive to the destination not only before the re-transmission time of the acknowledge mechanism, but also within the acceptable latency for SG applications. In light of the above results, we employ the duplicate packet method depending on the availability requirements of the applications. To do so, we categorize SG applications into three priorities with respect to availability and latency requirements (Table 2). We select three real SG applications [8] which represent general SG applications. The duplicate packet method is not used for Customer Information application. On the other hand, while

[9] Real-time apps typically have 150-200ms latency [9].

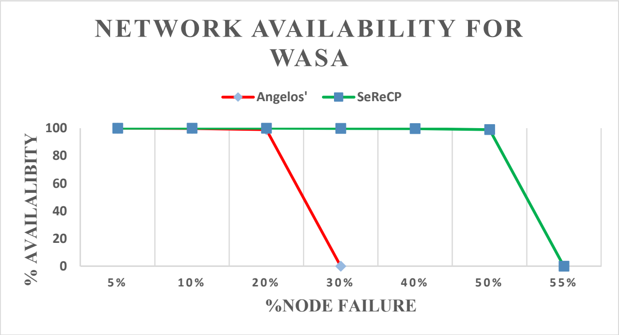


Fig. 7: Network availability for WASA (high priority/critical)

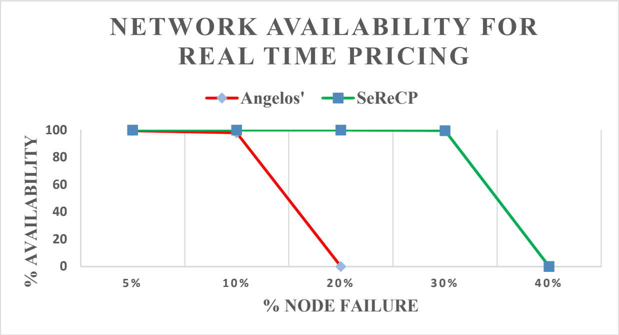


Fig. 8: Network availability for RT Pricing (middle priority)

all packets are duplicated for WASA application, only 50% of packets (randomly chosen) are transmitted as duplicated for the Real Time Pricing application.

Using these three duplication methods (100%, 50%, and 0% duplication) for the corresponding applications, we obtain results regarding the network availability for each application in diverse failure rates of access brokers by employing SeReCP and [4]. Fig. 7 denotes that by employing 100% duplication, while [4] provides the required availability for WASA application for only up to 20% failure, SeReCP introduces the same availability up to 50% failure for WASA despite its strict latency requirement i.e., 200 ms. Fig. 8 depicts that although Real Time Pricing relatively has lower latency requirement (1150 ms), by duplicating only 50% of packets, the required availability can be provided up to around 30% and 20% failure by SeReCP and [4], respectively. Finally, without duplication of packets, whereas SeReCP can still provide the required availability up to 30% failure due to the much lover latency requirement i.e., 2000 ms, [4] represents a sharp decline after 5% failure, as illustrated in Fig.9. Overall, by employing the duplication of packets depending on the application requirements rather than duplication of all packets as in [4], SeReCP can introduce high resilience against DDoS attack and provide the required availability for each application at least up to 30% failure of access brokers. In addition, if the first priority applications are safety-critical applications, then SeReCP sustains the required availability for up to 50% failures for these applications despite the second and the third priority cases shutting down after 30% failures.

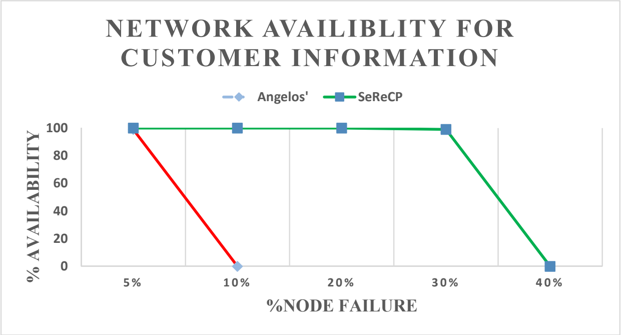


Fig. 9: Network availability for Customer Information (Low priority)

Discussion: In our evaluation, we assess our approach for its ability to provide the required availability for real-time SG applications during the DDoS attack on the pub-sub brokers. Furthermore, we compare our approach with [4], which is also proposed to protect the E2E connection of real-time application from DDoS attacks. Firstly, we evaluate our approach in terms of its additional latency and overhead. The results denote that SeReCP's additional overhead and latency is reasonable for most SG applications. It is worth highlighting that, although the latency results, obtained the actual Internet infrastructure, are reasonable for most SG application, the ISP connections of the end hosts significantly affect the latency. Secondly, we evaluate the normalized throughput of SeReCP and [4] for real time applications. Without duplicate packets, whereas SeReCP can protect the E2E communication up to 30% failure, the connection stalls after 5% failure in [4]. This shows that SeReCP preserves the E2E communication even without duplicate packets up to 30% failures. With duplicate packets, SeReCP can maintain the same performance up to 50% failures. In the third evaluation, by employing different duplicate packet rates depending on SG applications' "priority", while SeReCP can provide the required availability for middle and low "priority" SG applications up to 30% failures, it represents the same performance up to 50% failure for high "priority" ones. If we consider pub-sub brokers size comparable to a typical Akamai setting (ca. 2500 nodes) [4], an attacker coordinating around 1,300,000 zombies can bring down 30% of access brokers. However, this volume of attacks are a small percentage of the DDoS attacks experienced in the past. Even in this situation, SeReCP can provide the required availability for high priority (safety critical) applications despite the failure of the low and middle priority ones. This substantiates SeReCP to be a promising approach to make the public network usable for SG applications in a secure and reliable manner.

VII. Related works

To put our contributions in context, the related works span two distinct subjects fields: (i) Secure and reliable communication platforms for Smart Grid, and (ii) Overlay-based proactive DDoS defense mechanisms.

Secure and reliable communication for SG: GridStat [1] proposed a pub-sub network of message routers controlled by a hierarchical management plane to meet the NASPInet's QoS and security requirements. SeDAX [2] proposes a data-centric communication method on a secure overlay network, which contains trusted authentication servers allowing the parties to periodically obtain topic-based group keys to assure E2E confidentiality and integrity. SmartC2Net [12] aims to develop resilient solutions that facilitate the SG operations on top of heterogeneous off-the-shelf communication infrastructures. C-DAX [13] employs a pub-sub paradigm to decouple communication parties in space, time, and synchronization. C-DAX enables topic access control, end-to-end integrity and end-to-end confidentiality of the data, and authentication of nodes. Despite their lack of countermeasure for high-volume DDoS attacks, they offer promising features to incorporate with SeReCP to provide secure and reliable communication.

Overlay-based proactive DDoS defense mechanisms: Overlay networks can offer an Internet-wide network of nodes to create a first-level firewall against DDoS attacks. In this scenario the requests first need to pass through the nodes of Overlay Network before getting to the target server. Secure Overlay Services (SOS) [14] architecture consists of a three-layer hierarchy of overlay nodes to control the access to the protected target server. The goal is to ensure that any client can find a path to the target server under DDoS attacks; keeping the probability of compromising all available paths between clients and the target server small. Although SOS can protect against blind DDoS attacks, it is however ineffective against sophisticated and targeted DDoS attacks, because it is based on the assumption that the adversary attacks only a fixed subset of overlay. To overcome the shortcomings of SOS, [4] introduces a multipath overlay technique. In [4], a client randomly spread the data packets across all overlay nodes in order to sustain the communication under the attacks. Alternately, SIEVE [15] offers a lightweight distributed filtering protocol. SIEVE intends to expand the filtering and receiving capacity of the protected target. In this architecture the server needs to provide some kind of secret to the client that can help it to pass through the filter. Since SIEVE isolates the protected server in IP level by deploying it in a private network in order to protect the server from direct flooding attacks, it is not deployable in a network that contains large amount critical nodes/servers spread over a large-scale geographical area. On the other hand, to obtain further information about the security of pub-sub systems, [16] introduces a comprehensive analysis of the relevant state-of-the-art.

VIII. Conclusion

The proposed SeReCP approach provides a proactive DDoS attack defense by using a pub-sub infrastructure in addition to providing secure E2E data delivery in a lightweight manner, considering SG applications requirements. To maintain the availability in case of targeted or sweeping attack on a access broker maintaining the communication between a given publisher and its subscribers, we employ a packet diffusion mechanism, spreading the packets over access brokers in a pseudo-random manner thanks to its token-based authentication mechanism. Moreover, we propose a multihoming-based fast "recovery" mechanism, enhancing the packet diffusion mechanism by detecting and requesting the dropped packets in still access brokers rather than in the subscribers, thus enabling the system to meet the stringent latency requirements of SG applications. Finally, to preserve E2E confidentiality and integrity of the data, we propose a group key management system, which provides role-based access rights for both publisher and subscriber in addition to guard from replay attacks.

To assess the effectiveness of our approach against DDoS attacks, an actual SG test-bed was used. The experiments shows that SeReCP introduces a small 40ms overhead acceptable for most SG applications. Furthermore, we conducted an DDoS attack by randomly bringing down access brokers, and compared the availability across SeReCP and state of the art [4]. The results showed that by assigning the rate of duplicate packets depending on the applications availability and latency requirements, SeReCP provides the required availability up to 30% failure and 50% failure of pub-sub brokers for the application with relatively lower requirements and for the application with stringent requirements, respectively. This showed that SeReCP, with its lightweight mechanism, can resist attacks much larger than we have seen to date. Overall, these results validate SeReCP to provide the required security for SG applications in case of SG's used of public networks.

References

[1] Bakken D et al (2011) "Smart generation and transmission with coherent, real-time data." Proc IEEE 99(6), pp. 928-951

[2] Kim Y et al (2012) "SeDAX: A scalable, resilient, and secure platform for SG comm." IEEE J. Selected areas in Comm., 30(6): pp. 1119-1136

[3] Demir, K., Germanus, D., and Suri, N. (2015). "Robust QoS-aware communication in the smart distribution grid." Peer-to-Peer Networking and Applications, pp. 1-15.

[4] Stavrou, A. and Keromytis, A. (2005) "Countering DoS attacks with stateless multipath overlays." Proc. ACM CCS, pp. 249-259

[5] Dreibholz, T. (2015) "The NorNet Testbed A Large-Scale Experiment Platform for Real-World Experiments with Multi-Homed Systems". https://www.simula.no/research/projects/nornet

[6] Seferian, V., Kanj, R., Chehab, A. and Kayssi, A. (2014) "PUF and ID-based key distribution security framework for advanced metering infrastructures," Proc. Smart Grid Comm., pp. 933-938.

[7] Nahum, E., Rosu, M., Seshan, S., and Almeida, J., (2001). "The effects of wide-area conditions on WWW server performance", Proc. ACM SIGMETRICS, pp. 257-267.

[8] US DoE (July 12, 2010) "Implementing the National Broadband Plan by Studying the Communications Requirements of Electric Utilities To Inform Federal Smart Grid Policy" TR, http://energy.gov/sites/prod/files/gcprod/documents/UtilitiesTelecom Comments CommsReqs.pdf

[9] Ren, S., Guo, L., and Zhang, X. (2006). "ASAP: an AS-aware peer-relay protocol for high quality VoIP' Proc. ICDCS, pp. 70-70.

[10] Wang, Y., Yemula, and P., Bose, A. (2015). "Decentralized communication and control systems for power system operation" IEEE Trans on Smart Grid, 6(2), pp. 885-893.

[11] Black, J., Halevi, S., Krawczyk, H., and Krovetz, T., Rogaway, P. (1999) "UMAC: Fast and secure message authentication" In Annual Intl. Cryptology Conf, pp. 216-233

[12] Ciontea C, et al. (2015) "Smart grid control and communication: the SmartC2net Real-Time HIL approach" In: PowerTech15 IEEE Eindhoven, pp. 1-6

[13] Heimgaertner, F., Hoefling, M., Vieira, B., Poll, E., and Menth, M. (2015). A security architecture for the pub/sub C-DAX middleware. IEEE Intl Conf on Comm Workshop (ICCW), pp. 2616-2621

[14] Keromytis, A., Misra, V. and Rubenstein, D. (2002) "SOS: Secure overlay services" In ACM SIGCOMM Computer Commun Review, volume 32, pp. 61-72.

[15] Fu, Z. and Papatriantafilou, M. (2012) "Off the wall: Lightweight distributed filtering to mitigate distributed denial of service attacks", Proc. IEEE SRDS, pp. 207-212.

[16] Esposito, C., and Ciampi, M., (2015) "On Security in Publish/Subscribe Services: A Survey," in IEEE Communications Surveys & Tutorials, vol. 17, no. 2, pp. 966-997.