This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2016.2613545, IEEE Transactions on Smart Grid

Resilient End-to-End Message Protection for Cyber-Physical System Communications

Y.-J. Kim, *Member, IEEE*, V. Kolesnikov, *Member, IEEE*, and M. Thottan, *Member, IEEE*

***Abstract* —Cyber-physical system communications for safely and effectively operating a mission-critical infrastructure must be securely protected to prevent the infrastructure from becoming vulnerable. The protection scheme used must be resilient and light-weighted for cyber-physical system field devices having constrained computing and communicating resources, and also scalable for control servers associating with a large number of the field devices. In addition, cyber-physical system applications such as smart metering require end-to-end privacy protection. However, as shown in this paper, none of conventional security schemes comprehensively meets the above requirements; group security schemes scale well for a massive number of devices but are weak in terms of privacy protection and resilience; point-to-point security schemes such as IPsec inherently have resilience but are limited to address scalability and thinness requirements.**

Motivated by the limitations of conventional security schemes, we design new group security scheme, REMP (Resilient End-to-end Message Protection), exploiting the following notions: *long-term keys per-node* that are given by REMP authentication server, *encryption keys per message sent* that are probabilistically derived from a long-term key, and *end-to-end authenticators per message sent* that consist of a message sender's identity and a message authentication code. Compared with conventional group security schemes, we improve end-to-end security strength in terms of confidentiality, integrity, message source authentication, and key exposure resilience, while preserving scalability and extensibility.

***Index Terms* — Cyber-Physical System Security, Group Key Management, Scalability, Resilience, Message Authentication**

I. Introduction

Using Cyber-Physical Systems (CPS) is expected to significantly improve safety, reliability, and efficiency in operating nationwide or statewide critical infrastructures such as power grids or transportation networks. As shown in Fig. 1, a CPS for the critical infrastructures (hereafter called large-scale CPS) can be modeled as a machine-to-machine communication system that combines a central control facility for providing intelligence, sensors as physical inputs, and actuators for implementing control operations [1]. In a large-scale CPS, a massive number of field sensors (*i.e.*, Internet of Things scale) continuously publish measured data; their associated control facility collects the measured data to perform real-time (*i.e.*, responding to time deadlines) data analysis and if necessary sends control commands to actuators where physical actions will be executed. An example of large-scale CPSs is the smart grid where the electricity utility can cost-effectively maintain the balance between power load and supply through the use of smart meters/sensors and timely and accurate reporting of power loads.

Y.-J. Kim, V. Kolesnikov and M. Thottan are with Nokia Bell-Labs, Murray Hill, NJ USA (e-mail: young.kim@nokia-bell-labs.com; vladimir.kolesnikov @nokia-bell-labs.com; marina.thottan@nokia-bell-labs.com).

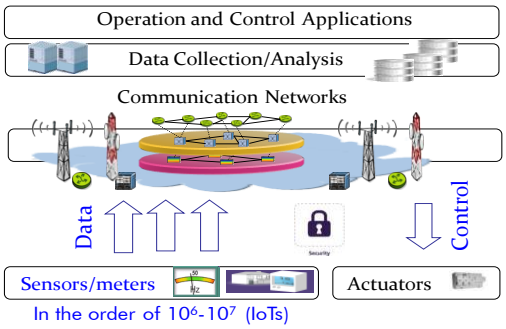

Fig. 1. The simplified system model of a large-scale CPS.

A large-scale CPS [2] requires secure and available communications as its elements are geographically distributed in a field area network and thus can be exposed to adversaries who are external to the CPS. Otherwise it can face safety (and economic) hazards. In particular, information exchanged over the large-scale CPS must be strongly protected against cyber attacks on end-to-end (E2E) (*versus* link-by-link) aspects. In communications for information collection from smart meters or distributed energy sources, the E2E security support that is necessary for protecting privacy among electric consumers or suppliers is identified as a non-negotiable security criterion. We thus emphasize that with no E2E security support, the credibility of the large-scale CPS is questionable.

However, there is, to the best of our knowledge, little study on the E2E security for the large-scale CPS (*i.e.*, smart grid). We will discuss in Sec. III that conventional security schemes applied to cyber-only infrastructures (*i.e.*, Internet) are limited to support the large-scale CPS. Conventional group security schemes have scalability advantages but are seriously weak in terms of privacy, integrity, message source authentication, and key exposure resilience. On the other hand, conventional point-to-point security schemes applied to Internet are limited in their ability to address at scale the E2E message protection for the large-scale CPS. We emphasize that the challenge is to bring the E2E security measure that provides privacy, integrity, message source authentication, and key exposure resilience for large-scale CPSs. We need to consider computational burden to account for the impact of security on resource-constrained devices and access communication networks in terms of computing ability and communication bandwidth. Note that this paper focuses on key management subject for new group security scheme that can be applied to the large CPS at scale, since the above limitations of conventional security schemes are mostly caused by inefficient key management mechanisms.

It is in this context that we devise a resilient E2E message protection (*REMP*) framework that addresses the requirements described above. The key concept of REMP is that, for a communication group where the same type of information is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2016.2613545, IEEE Transactions on Smart Grid

exchanged, its participating members have group information only and as a result, its participants as receivers do not need to have knowledge of any state for security per-sender.

Our main contributions are: (1) We survey today's security practices widely used for Internet, and show that they do not comprehensively address security strength, scalability, and thinness, in the context of the large-scale CPS. (2) We design new E2E message protection scheme for large-scale CPSs; the scheme eliminates the need for supporting costly solutions such as IPsec [3] or TLS [4] in terms of scalability and management. (3) Our security extension addresses source authentication problem that is known to be hard [5], and (4) The *O*(*1*)-state concept of REMP has benefits on performance and management aspects: the computation throughput on receivers scales with the increase of CPU frequency or the number of CPU cores as it does not depend on the number of senders; further, message flooding from a massive number of field devices during session reestablishment following server restarts or failures can be avoided.

This paper is presented as an extension of our prior work [6]. In addition to considerable editorial changes, we show the computation burden of REMP, which is built over commodity Linux PCs and open-source cryptographic library, to confirm that REMP servers scale well with a large number of client devices and also REMP clients can work well under resource-restricted conditions. By measurements, we have noticed that the original REMP [6], which consists of symmetric-key only operations and is designed for supporting secure information collection from a large number of field devices such as meters, is limited to support secure message multicast to a massive number (*i.e.*, more than 10^4 in our setting) of field devices. Motivated by this limitation, we introduce a variant of REMP, *REMP+*, where ECC (Elliptic Curve Cryptography) [7] is used to extend the message source authentication of REMP [6].

II. Properties of Large-scale CPS Communications

We now describe some of the characteristic properties of the large-scale CPS (*i.e.*, smart grid) communication [8] that is relevant for designing the REMP. First, in the large-scale CPS, fixed-size data published by a massive number of embedded field devices (data publishers) is continuously delivered to data collection servers (data subscribers), as shown in Fig. 1. This fixed-size traffic from field devices dominates the CPS communication network because control messages from CPS servers are sent to field devices only if necessary. Second, the CPS communication sessions for safe delivery of sensor data and control messages need to be *persistently-lived*. The persistent session is necessary for minimizing communication delay and for avoiding computation and communication overheads required per session establishment. Compared to the CPS traffic, Internet traffic shows heavy-tailed distribution on data size and session duration. Third, delay requirements in the CPS are communication-group specific. For example, the delay budget for phasor measurement [9] is in the order of 100 milliseconds. By contrast, the budget for smart metering is in the order of minutes. Fourth, field devices such as sensors or meters are typically purpose-built machines with constrained computing resources [10]. On the other hand, control facilities

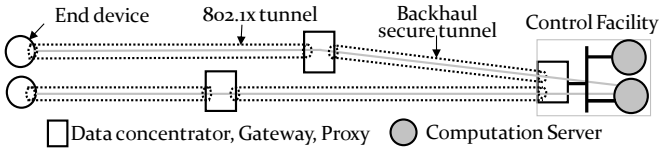

Fig. 2. A typical configuration of link-by-link secure tunnels.

are made up of high-performance machines since the high volume of data that is collected from the massive number of field devices must be processed in a timely fashion. As a result, scalability is a major consideration for powerful machines in a control facility, while light-weight computation is critical for computation-constrained machines. Resilience to attacks is an essential requirement for all kinds of machines. Note that this asymmetry in the availability of computing resources and the heterogeneity in the delay requirements described above must be taken into account when security measures are developed. Fifth, CPS communication traffic is physically or virtually isolated from public network traffic due to the significant security and availability that can be encountered if messages containing mission-critical data are multiplexed with public network traffic. However, as will be described in the end of this section and in Sec. III.D, simply isolating CPS communication traffic is not sufficient to ensure secure communication, as third-party adversaries can easily exploit security holes in or intermediate communication nodes or central control facilities. Sixth, CPS communication entities that publish and consume data are governed by *one single pre-assigned* administrator: smart metering is typically operated by one single utility; wide-area situation monitoring that spans multiple utilities can be operated by a single independent organization. Lastly, CPS communications could be deployed over multiple access technologies, *e.g.*, smart metering over PLC (Power Line Communication) or IEEE 802.15.4 smart utility networks, and distribution automation over optics or cellular such as LTE (Long Term Evolution). Each access technology has its own security scheme. However, link-level (*versus* E2E) security schemes typically used by the access technologies are limited in terms of ensuring E2E message protection since they cannot guarantee secure communications among end-point devices through intermediate communication nodes such as data aggregator. Consider the scenario shown in Fig. 2, where messages encrypted by IEEE 802.1X (used for wireless mesh network security) are securely delivered over a network. The messages are decrypted in an 802.1X access gateway and then transmitted over a backhaul network using encryption schemes of the backhaul network. It is possible that the access gateway is compromised and CPS communications can be exposed to adversaries. Thus, it is critical that CPS supports security technologies that ensure E2E security.

III. Review of Known Message Protection Schemes

We here show that conventional security schemes do not meet security requirements of smart grid as a large-scale CPS.

A. Smart Grid Cyber Security Guidelines

NIST (National Institute of Standards and Technology) has published the guidelines for smart grid cyber security [11] that includes the description of cryptography and key management issues. Informatively, NIST recommends the usage of standard

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2016.2613545, IEEE Transactions on Smart Grid

	Smart meter	Sensor (PMU)	DER (PV)	Plug-in EV	Admin. msg.
E2E privacy	√		√	√	
E2E integrity	√	√	√	√	√
Message source authentication	√	√	√	√	√
Latency req.	min	sub-sec	sec	sec	sub-sec

Fig. 3. Security and latency requirements of smart grid applications.

symmetric ciphers such as AES (Advanced Encryption Standard) [12] for message encryption and CMAC (Cipher-based Message Authentication Code) [13] for message integrity on which most security schemes including REMP rely today. More importantly, NIST guidelines have identified as an R&D priority enabling key management on a scale involving, potentially, tens of millions of credentials and keys as well as cryptographic processing on resource-constrained sensors such as encryption and digital signatures. Also, it explicitly notes: "low bandwidth channels may be too slow to exchange large certificates, and thus if the certificate-based key-establishment exchange is time critical, protocols such as IKE [14] that exchange multiple messages before arriving at a pre-shared key may be too costly, even if the size of the certificate is minimal". Among an amount of grid applications having various security and availability requirements, we here focus on supporting grid applications that are listed in Fig. 3.

B. Group Communication Security

We notice that the per-group key management property of conventional group security schemes is well aligned with the Publish-Subscribe (hereafter caller pub-sub) communication [15] property as the first property of the CPS communications described in Sec. II. In the context of publish-subscribe group communications, group communication security has different pros and cons in terms of communication message protection.

Consider the incorporation of pub-sub communications into smart grid, where all field devices (*i.e.*, smart meters or sensors) as publishers are divided into a small number of groups, and control facility head-end servers as subscribers participate in all the groups. Pub-sub communications refer to *unidirectional many-to-many* communications among group members (see Fig. 4.). We can find significant advantages in scalability and management in terms of E2E communication message protection as, for a group, all members share single symmetric group key for message encryption [16], or rely on public keys. Pub-sub group communications inherently protect end-point servers in a control facility against attacks that can be launched from field devices, since the field devices have no knowledge of the end-point servers and the end-point servers do not setup sessions with the devices themselves due to time and space decoupling between publishers and subscribers [15].

However, existing group security schemes for pub-sub systems have limitations of security strength and/or efficiency due to the use of either one single symmetric key per group or public keys despite ensuring authentication at group join time.

For the use of one symmetric key per group [16], we can identify the following limitations. First, legitimate publishers in a group can listen to messages from other publishers in the group (*privacy violation*), *i.e.*, existing group security schemes cannot be used for privacy-preserving infrastructures such as smart metering. Second, a compromised subscriber in a group

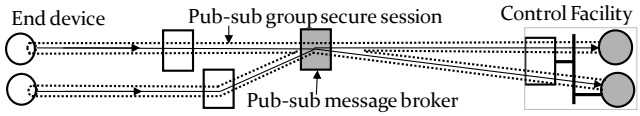

Fig. 4. A pub-sub communication configuration for a large-scale CPS.

can send messages to other subscribers because it can disguise as a legitimate publisher (*source authentication problem*). This is a well-known open problem in group communications. The problem was tackled in TESLA [5], which supports multicasting of one sender rather than pub-sub communications of multiple senders. TESLA has limitations on delay and memory scale as a receiver temporarily stores messages at its buffer until keys for authenticating the messages are revealed. It also requires an external time-synchronization scheme where the clock drift of sender and receiver must be very small, and receivers periodically resynchronize the time with its sender. Third, accidental or incidental exposure of a group secret key to attackers may result in whole system failures (*key exposure resilience problem*). Lastly, group secret keys must be updated to ensure forward-backward secrecy whenever a member joins or leaves the group (*key refresh problem*). For a pub-sub group having N members, refreshing a key needs O(N) message exchanges in a brute-force fashion and O($log\ N$) in tree-based approaches such as LKH [17]. However, both of O(N) message exchanges and key tree managements are costly for a large-scale CPS communication network such as an advanced metering infrastructure that consist of a massive number of smart meters and is likely to be built over narrow-band communication technologies including PLC or IEEE 802.15.4.

For the use of public keys, the following limitations of thinness and performance are identified. First, public-key operations are difficult to implement on low-powered field devices due to their intensive computation, as will be specifically described later. Second, public-key operations unnecessarily consume limited communication resources. Consider two end-point servers that are interested in data published by field devices (See Fig. 4). Public keys of the end-point servers are likely to be distributed to field devices since the number of end-point servers is typically much less than the number of field devices. An end device must twice encrypt the same data using the two public keys and send the twice encrypted data for both the servers to decrypt the data. Note that the size of the certificates [4] that are exchanged among communicating parties to ensure authenticated public keys is typically greater than 2K bytes, and public-key operations (*i.e.*, encryption-decryption and/or sign-verification) incur almost hundred-to-thousand times more computing resources (see Fig. 10) than symmetric-key operations. Thus, it is difficult to implement these schemes on field devices with constrained computing power or bandwidth, *e.g.*, sensor platforms with 16-bit 10MHz processors [10] and IEEE 802.15.4 modules. For a large-size message more than 2K bytes, the end-to-end delivery delay over IEEE 802.15.4 or PLC networks is known to be sometimes more than one minute due to high-losses and scheduling delays that avoid MAC collisions. Accordingly, we cannot ensure the feasibility of implementing certificate-based security schemes over narrow-band communication networks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2016.2613545, IEEE Transactions on Smart Grid

C. Point-to-Point Communication Security

Unlike existing group security schemes, existing point-to-point security schemes have no serious weakness of resilience and security strength since they protect each secure session per two communicating parties other than per group. We here consider four well-known point-to-point security schemes that are widely used for Internet, IPsec [3], TLS [4], DTLS [18], and SRTP [19]. However, we show in this section that the key management of all the schemes can reduce scalability and/or thinness required to support large-scale CPS on E2E aspects.

IPsec (IP Security): For establishing an IPsec tunnel, it performs mutual-authentication between two communicating parties and then exchange keys to be used for protecting the session. However, this is accomplished by an external scheme, IKE [14], which needs certificates and public-key operations.

TLS (Transport Layer Security) and **DTLS** (Datagram TLS): These schemes also require certificates and public-key operations for mutual-authentication and key management.

SRTP (Secure Real-time Transport Protocol)**:** It relies on an external key management protocol, ZRTP [20], to establish one master key for deriving session keys. Moreover, it needs additional time-synchronization for key derivation.

Scalability Requirement: All of the above protocols share one limiting property. *E.g.,* consider the scenario of data collection from *N* smart meters embedded in smart grid. For ensuring E2E secure communications, data collection (and computation) servers in a central control facility must maintain O(*N*) secure sessions. When *N* is large (*i.e.,* in the order of millions), a big chunk of memory of the computation servers is occupied by security tasks assigned to handle messages from *N* smart meters, computation intensive activities such as real-time data analysis and control algorithms may face a temporary shortage of runtime memory during their computation that could result in missed deadlines. Also, exposing computation servers to a large number of smart meters must be avoided as adversaries can easily develop cyber attacks via smart meters. In short, enforcing E2E message protection to eliminate security holes necessarily involves deploying cost-effective scalability in the system. Further, all of the above referred protocols can incur message flooding when the computation servers have abruptly failed or are restarted for upgrading. Meters associated with these servers will simultaneously send thousands of control messages to re-setup their secure sessions as soon as possible.

Thinness Requirement[1]: All the protocols except for SRTP have dependencies on certificates and public-key operations for mutual-authentication and key management. Therefore, IPsec, TLS, and DTLS are inherently not light-weighted in terms of computations for resource-constrained field devices such as smart meters, as already described in Sec. III.B.

Extensibility Requirement: For an extensible deployment in a large-scale CPS, a newly installed or rebooted field device must have knowledge of the name of its pre-assigned control facility rather than the IP address of its associated end-point server. This is necessary to establish the necessary security associations. Therefore, a massive number of distributed field devices must perform the name resolution for end-point servers. It poses a challenge for the seamless replacement of end-point servers. In addition, using existing security schemes needs secure name resolution systems such as DNSSEC (DNS Security Extensions) [21]. However, DNSSEC relies on costly public-key operations to ensure message source authentication and moreover has no confidentiality for DNSSEC messages.

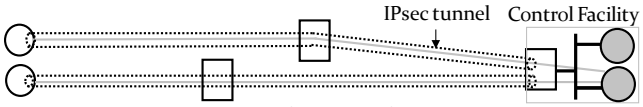

Fig. 5. A typical configuration of IPsec tunnels.

D. Operation Practice Issues

Security breaches may be developed when E2E security policy is not strictly enforced. Consider a central control facility whose servers communicate with field devices through gateway routers in the subnet where the servers are located. By typical security operation practices, when the number of field devices is large (*i.e.,* in the order of millions), due to hardware limitations, secure tunnels with the field devices are likely to be terminated at a gateway router other than a server that is the expected destination node of messages from a field device. The reason for the termination of secure tunnels at the gateway is primarily due to performance and management issues, as illustrated in Fig. 5. In such cases, communications between the server and the gateway are unprotected unless there are other security measures ensuring confidentiality and integrity for the communications. In addition, security threats can be caused by internal adversaries in control facilities that include compromised (or disgruntled) employees or malwares that can intrude via infected portable devices of employees.

E. Related Work

We now review relevant work to REMP in terms of key management. There is a considerable amount of prior work on key management for smart grid, as shown in [22]. However, none of prior work meets all the following requirements: thinness, scalability, extensibility, and E2E security policy. Specifically, most prior schemes require the use of PKI (Public Key Infrastructure) certificate; otherwise, hierarchical communication trees or abundant computational resources are required [23][24]; in some cases, the E2E security policy or multicast (or group) communications are not supported [25] [26]. More importantly, all the proposed security schemes are classified into stateful approaches where a receiver must keep per-sender security state. Therefore, none of the schemes is similar with REMP in terms of design goals.

On the other hand, SRTP [19] is similar to REMP in the sense of message encryption using different short-term keys rather than a single long-term key. In SRTP, communicating parties share a master key and extract each short-term key using a key derivation function, the master key, and a sequence number. However, establishing the master key relies on extra schemes such as ZRTP [20] that need a non-negligible number of control message exchanges. Importantly, the master key exposure or out-of-ordered sequencing can cause security failures. In addition, SRTP supports only RTP.

[1] Collecting data from real-time measurement devices such as PMUs should be timely but typically requires neither reliable-guarantees nor low jitter. Thus, UDP along with a message integrity scheme such as CMAC [13] will suffice. By contrast, using TLS or SRTP, which need TCP or RTP respectively, will result in unnecessarily-costly CPS communications.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2016.2613545, IEEE Transactions on Smart Grid

TABLE I. TERMINOLOGIES USED FOR EXPLAINING REMP

Pub_i	Publisher with identity *i*	*pi*	A publishing key for Pub_i
Sub_j	Subscriber with identity *j*	*ai*	An authentication key for Pub_i
$\|\|$	Concatenation	*aj*	An authentication key for Sub_j
E_x	Encryption using key *x*	*pm*	A single publishing master key of *g*
D_x	Decryption using key *x*	*tk*	A single access-ticket key of *g*
M_k	The *k-th* message of Pub_i	T_i	An access-ticket for Pub_i, $E_{tk}(i \,||\, ai \,||\text{'W'})$
rn	The *k-th* random number	T_j	An access-ticket for Sub_j, $E_{tk}(j \,||\, aj \,||\text{'R'})$
sk	A session key for M_k	*kp*	A public key of message broker *B*
g	A group identity	*kr*	A private key of message broker *B*
X_k	A message, $\{E_{sk}(M_k), rn, g\}$	H_x	Cipher-based hashing using key *x*
ts_j	A time stamp of Sub_j	A_i	An encryption, $E_{ai}(j \,||\, ts_j \,||\, j$'s IP address)

IV. DESIGN OF REMP

We propose the design of REMP in a top-down modular manner for simplicity of presentation and understanding.

A. Overview of the approach

The design goals of REMP as new pub-sub group security scheme are to improve end-to-end security strength in terms of privacy, integrity, message source authentication, and key exposure resilience, and to accommodate resource-constrained environments, while preserving the scalability and extensibility inherited from pub-sub group communications[2].

One encryption key per-message: Each publisher executes encryption using a separate session key per message sent. This approach addresses privacy among publishers in a group. As it provides forward-backward secrecy, key update caused by new member join/leave can be avoided and key exposure resilience is inherently improved. Further, it prevents attackers from collecting and replaying large amounts of cipher text encrypted with a single session key on a per-group basis.

***Subscriber's state independent of the number of publishers*:** One novelty of REMP is that a subscriber computes a one-time decryption key when a message arrives. This capability is enabled by the use of a long-term master key. In this way, a subscriber does not need to keep security state per-publisher. In addition, this idea helps avoid extreme overloading in the face of subscriber restarts or failures. Thus, we can obviously observe the scalability of REMP in a group with a massive number of publishers, *e.g.*, a data collection group consisting of meters (as publishers) and utility head-ends (as subscribers).

Message source authentication extension: To tackle message source authentication, which is known to be a hard problem [5], we exploit the notion of *E2E authenticators* and *message brokers* that multicast messages from publishers in a group to subscribers in the group (see Fig. 8 and Fig. 9). For a message sent, its E2E authenticator consists of the sender's identity and a ciphered message authentication code [13] of the message.

***Symmetric-key based approach for resource constraints*:** As discussed in Sec. II, in CPS communications, end devices communicate with end servers under a single administrative domain. Thus, one pre-shared key (PSK) [27] per end device and symmetric ciphers such as AES [12] or 3DES [28] will suffice. In this setting, using costly public-key credentials has no benefit of system-wide reduction in the number of keys. Symmetric-key only operations are thus used in most REMP security extensions including confidentiality and integrity. For the message source authentication, unlike our prior work [6], we now use ECC [7] together with symmetric key operations for supporting secure multicast to a massive number of subscribers and still preserve symmetric-key only operations for supporting secure data collection from a large number of publishers, as will presented in Sec. IV.C.

[2] Pub-sub communications are well-aligned with CPSs on terms of scalability and extensibility, as discussed in Sec. III. But, without optimized protection schemes such as REMP, pub-sub communications will not scale to CPSs.

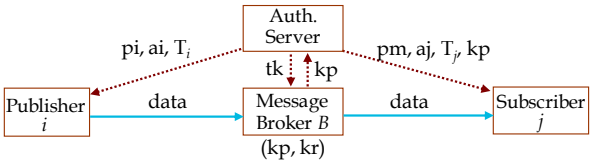

Fig. 6. Our system model of pub-sub group communications.

B. System Architecture

We begin with the discussion of our pub-sub group communication framework [29] that is leveraged by REMP. Consider Fig. 6. For all pub-sub groups, authentication servers are responsible for member authentication and key distribution to members. In a group, each member (publisher, subscriber, or message broker) must be authenticated and then assigned keys. A message broker authenticated for a group maintains state for subscribers authenticated for that group. Whenever it receives a message, it verifies if the source of the message is a publisher permitted to access the group and thus filters out any unauthenticated message. We emphasize that one of our design principles is to improve E2E security strength for group communications in terms of confidentiality and integrity. Thus, in a group, messages encrypted and hashed by publishers can be decrypted and verified by only subscribers. Even message brokers in the group are not allowed or able to decrypt the message. Each node can participate into more than one group but only play either as a publisher or subscriber in each group.

C. Design Details

Symmetric-key member authentication: For participating in a certain group, each member must be authenticated by an authentication sever under the same administrative domain [16]. Conventional approaches that use certificates and public-key ciphers or require many message exchanges, are not always suitable for CPS communications where end devices or access networks can be resource-constrained. By contrast, symmetric key based approaches (*i.e.*, PSK-based scheme [30] or just-in-time key computation scheme [31]) are appropriate for CPS communications, by virtue of the properties described in Section II. Please refer to [30] and [31] for details. An authenticated member can safely acquire secret information over a secure channel with its associated authentication server.

Long-term key assignment and access-ticket: For a group, an authentication server creates and/or distributes five kinds of long-term keys: a single publishing master key, a single access ticket key, a public key of a message broker, an authentication key per member, and a publishing key per publisher, as shown in Fig. 6. For a group *m*, given a single publishing master key *pm*, a publisher with identity *i* is assigned a publishing key *pi* = AES_{pm} (*i*). Security properties of AES function guarantee that none of the publishing keys can be distinguished from a random string, even if the adversary obtains publishing keys of all other publishers. Thus, these keys are safe to use. On the other hand, publishing master *pm* is given to all subscribers. *pi* and *pm* are subsequently used to compute keys for message encryption and decryption respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2016.2613545, IEEE Transactions on Smart Grid

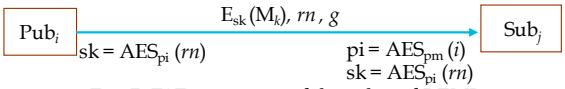

Fig. 7. E2E message confidentiality of REMP.

For message source authentication and access control, we use the notion of "access-ticket" borrowed from Kerberos [32]. An authenticated member with identity *i* is given access-ticket T_i, authentication key *ai*, and a message broker *B*'s public-key *kp* (only for subscribers), as shown in Fig. 6. T_i is a cipher text that contains identity *i*, authentication key *ai*, and access-right for group *m*, and whose encryption is done by one single access-ticket key *tk* that is assigned to group *m*.

E2E message confidentiality and E2E message integrity: We compute encryption keys using a key derivation function for message encryption and decryption. A key is pseudo-randomly generated from a publishing key and a random number. As illustrated in Fig. 7, given a publishing key *pi*, for a random number *rn*, a publisher *i* computes the key *sk* = $AES_{pi}(rn)$. When a subscriber gets an encrypted message from publisher *i*, it first computes the publishing key *pi* using the publishing master key *pm* that was given by applying the AES function; from *pi* it can derive the session key *sk*. Thus, it does not store this session key, since he can readily compute it given a publishing key and a random number. In addition to session key derivation, publishing keys support E2E message integrity using MAC functions such as CMAC [13], as shown in Fig. 8 and 9. Using key derivation functions over publishing keys and random numbers improves privacy among publishers in a group. We emphasize that REMP supports confidentiality and integrity on aspects of E2E security policy.

Message source authentication extension: We now describe an extension for addressing message authentication: without this extension, we cannot prevent a compromised subscriber in a group from disguising as a legitimate publisher in the group since the subscriber has a publishing master key for the group.

For a subscriber with identity *j* of group *g*, given a message broker *B* (details of message broker selection is presented in [29]), we first establish a security association between *B* and *j*. Thus, subscriber *j* can ensure that any message received really comes from its associated message broker *B*. Consider Fig. 8. After it is authenticated, subscriber *j* creates an encryption A_j, which contains identity *j*, time stamp ts_j, and IP address of *j*. A_j is encrypted with an authentication key *aj* of *j*. Subscriber *j* sends a control message containing A_j and an access ticket T_j (given by an authentication server) to message broker *B*, who holds an access ticket key *tk* given by an authentication server (see Fig. 6). When message broker *B* receives the message, it can extract both the authentication key *aj* and access-right of subscriber *j* from T_j using *tk*, and then verify A_j using *aj*. As a result, a message broker can establish a security association with each legitimate subscriber.

For a given authentication key *ai* and an encrypted message $X_k := \{E_{sk}(M_k), rn, g\}$, publisher *i* can create an E2E authenticator, $i||H_{pi}(X_k)$, which contains identity *i* and a cryptographic hash of X_k using its publishing key *pi*, and then encrypts the E2E authenticator using *ai* (A weak form of non-malleability of the latter encryption is needed and adversaries can't meaningfully modify a message under encryption. The use of AES as the encryption scheme is sufficient.). X_k, the

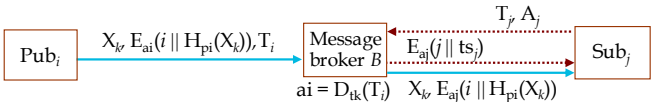

Fig. 8. Message source authentication of REMP.

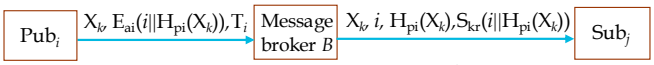

Fig. 9. Message source authentication of REMP+.

encrypted authenticator, and access ticket T_i are together sent to message broker *B*, as shown in Fig. 8 and 9. Upon reaching *B*, we first extract identity *i* and authentication *ai* from T_i using the access ticket key *tk*, and then identity *i* from $E_{ai}(i||H_{pi}(X_k))$ using *ai*. If the two identities are equals and the number of subscribers associated with *B* is not large, message broker *B* follows up the original message source authentication process described in [6]. The details of the process are omitted due to space constraints. Please refer to [6] for the details.

In fact, the message source authentication of REMP is limited to accommodate applications that require message multicast to a large number (*i.e.*, more than 10^4) of subscribers. For an E2E authenticator received, a message broker performs N_S per-subscriber encryptions of the E2E authenticator if it has associated with N_S subscribers. Hence, the performance of a REMP message broker depends on the number of subscribers associated with the message broker, as shown in Fig. 10.

REMP+: Motivated by this limitation of [6], we introduce a variant of REMP, REMP+, whose message source authentication procedure uses ECC (a public key cryptography) [7]. In REMP+, for an E2E authenticator received, message broker *B* signs the E2E authenticator with its private key *kr* paired with its public key *kp*, and sends the same signature to its associated subscribers, as shown in Fig. 9. For the signature received along with a message, all subscribers of *B* can verify the signature since they already hold public key *kp* given by an authentication server at group member authentication time.

The performance of a REMP+ message broker using ECC signing algorithm is independent of the number of subscribers, compared to REMP message brokers. As a trade-off, we can experience some performance degradation in subscriber sides due to computation-intensive ECC verification (see Fig.10). Even with ECC verification, REMP+ can account for resource constrained devices as subscribers [33]. The only difference of REMP+ against REMP is this message source authentication procedure among a message broker and its subscribers.

D. Discussion on Replay Attacks

The message broker of REMP+ (or REMP) is essentially stateless for publishers and thus REMP+ may be vulnerable to *replay attacks*. We provide a mechanism to protect the broker from this type of attack. First, we remark that publisher's authentication key is unforgeable and with proper formatting and care we can prevent adversaries from presenting an E2E authenticator generated for a publisher as an E2E authenticator of other publisher. Further, re-encrypting an E2E authenticator with no knowledge of an authentication key is not possible either. Hence, the only venue of the replay attack is the verbatim replay of one of the previously encrypted E2E authenticators with a possibly different session message. Recall that each message is cryptographically hashed with its publisher's publishing key and so is implicitly tied with an E2E authenticator. As a result, such a replay attack can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2016.2613545, IEEE Transactions on Smart Grid

always detected in subscribers (E2E integrity). Thus, the only replay attack that remains to be considered is the verbatim replay of the entire publisher's message. And indeed, our presentation so far is vulnerable to this attack. We discuss our protection method. Firstly, in typical CPS settings (see Fig. 3), the number of messages that can arrive in the time period of several seconds is not very large, and so we can afford to keep the history of their hashes. Thus, for each new message, we will check it against the small recent history of hashes, and reject it if it is found in the history; if not found, we proceed as before. This will protect our systems against accidental replay. We cannot eliminate malicious replay attacks at a protocol level due to our state restriction. Recall, however, that for our application scenarios, messages from devices embedded in a large scale CPS are time stamped. We can thus delegate the final timestamp and duplication checks to the application layer, where this can be done much more efficiently.

V. EVALUATION

A. Security Strength

We now recap the security strength of REMP+. Keep in mind that like other security measures, we rely on standard ciphers, AES [12], CMAC [13], and ECC [7]. First, the long-term key generation in our proposed scheme is secure due to the properties of the AES function. The follow-up message encryption and signature, based on symmetric ciphers [12] [28] and ECC ciphers chosen from the standards, is also secure. Next, the derivation of short-term session keys from long-term keys and random numbers results in the following benefits. It prevents attackers from developing attacks by passively collecting large amounts of cipher text encrypted with one long-term session key. Further, it provides forward-backward secrecy in the sense that a compromised session key does not compromise other session keys derived from the same long-term key. Another possibility is to compromise a subscriber having a single publishing master key for a group. However, such an attack will only result in the (expected) ability of attackers to listen to group's messages. In particular, attackers cannot disguise as a legitimate publisher of the group due to our message authentication extension. One remaining potential attack is to compromise each publisher. However, we can confine the effect of such an attack to only the publisher.

B. Scalability, Availability, and Message Overhead

Due to the property that subscribers in a group do not directly communicate with publishers in the group, we can outperform most point-to-point security schemes (*i.e.*, IPsec, TLS, DTLS, SRTP, etc) in terms of scalability and availability, and as shown in Table II. In terms of communication overhead for message protection, REMP+ is comparable to alternatives. For a given message, REMP+ consumes additional bandwidth for three extra fields, 2-bytes random number, 2-bytes group identity, and either 12-bytes access ticket (from a publisher) or 32-bytes ECC signature (from a message broker) [7]. On the other hand, alternatives except for SRTP is more than 40 bytes additional overhead and SRTP running over RTP has more than 32 bytes additional overhead. Note that the $H_{pi}(X_k)$ field as a cipher-based message authentication code (MAC) is not an additional overhead of REMP+ since alternatives typically add a MAC to every message for message integrity. Its size is 8 or 12 bytes when CMAC [13] is used as a MAC generator.

TABLE II. Qualitative comparison of REMP+ against alternatives

	Point-to-Point	Group Security	REMP+
Memory scale	$O(N)$ in a receiver	$O(1)$	$O(1)$
Restart of receivers	Message flooding	Zero message	Zero message
Seamless deployment	No	Inherently support	Inherently support
Privacy protection	Inherently-support	No	Support
Source authen.	Support	No	Support
Authentication & key distribution	External protocols or complex proc.	Inherent but public key operations	Inherent and simple symmetric key ops
# of messages exchanged for key refresh	Typically unnecessary	$O(\log N)$ per member- join (or leave)	Zero

TABLE III. Additional computations of REMP+ against alternatives

	Publisher	Message Broker	Subscriber
Confidentiality	One AES op to compute *sk*	None	Two AES ops to compute *pi* & *sk*
Integrity	None	None	None
Message Source Authentication	One AES op to encrypt an E2E authen.	For an E2E authenticator, two AES ops and one ECC signature	For an E2E authenticator, one ECC verification

C. Computational Burden

Table III shows the extra computations of REMP+ against alternatives for a given sent message M_k. Note that encryption for confidentiality and cryptographic hashing for integrity are common across all alternatives. For a communication message, a publisher has two additional AES operations and both of a subscriber and a message broker additionally have two AES additional operations and one additional ECC operation. The additional computational burden for REMP+ stems mostly from message source authentication process. The performance degradation introduced by REMP+ is quite small since even in low-powered sensors [33], the speed of ECC signature and verification is in the order of hundred milliseconds, and the speed of the symmetric cipher AES is in the order of microseconds for small-size data such as $i||H_{pi}(X_k)$ whose size is 12~16 bytes when CMAC performs a cryptographic hashing. According to [34], the AES operation time of 40-bytes data is about 100 μsec over a 10 MHz micro-processor [10] as AES is known to spend about 50 clock cycles for processing one-byte data. Accordingly, we can notice that REMP+ well supports resource-constrained devices. In addition, we remark that the computation burden in message brokers typically having high-powered computing ability is negligible, *i.e.*, about 100 μsec.

On the other hand, we have performed a measurement for confirming the scalability advantage of REMP and REMP+ on aspects of performance. In particular, for a communication message, we measured the computation time of the symmetric cipher AES with a key size of 128 bits and the *additional computation time* of a publisher, a subscriber, a message broker with one subscriber, and a message broker with 1500 subscribers respectively. The measurement was performed in the following setting: 1) a Linux PC with Intel Core i7-3770 quad-core CPU 3.40 GHz and 16 GB RAM, and 2) OpenSSL-1.0.1f [35] that is an open-source cryptographic library.

Fig. 10 shows that both of a publisher and a subscriber have sub-microsecond additional computation time since one AES operation time for a 36-bytes data is about 0.24μsec. On the other hand, the additional computation time of a message broker is about 0.57 μsec for one subscriber and about 83.15 μsec for 1500 subscribers. The result is caused by the fact that in a pub-sub group with N_S subscribers, a REMP [6] message broker must perform N_S per-subscriber encryptions for an E2E

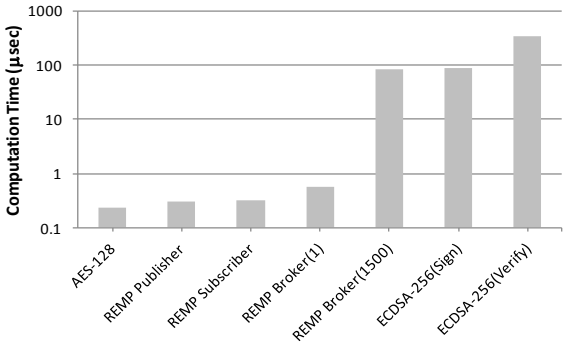

Fig. 10. Additional Computational Time of REMP/REMP+.

authenticator received from a publisher (see Fig. 8). We thus confirm that the performance of a REMP message broker is a function of the number of subscribers of the message broker.

Recall that REMP+ designed for addressing the limitation of REMP [6] uses ECDSA (Elliptic Curve Digital Signature Algorithm) [7] for signing or verifying a message (see Fig. 9). Fig. 10 indicates that the performance of a REMP+ message broker is independent from the number of subscribers of the message broker since the message broker can sign a message received from a publisher and then once multicast the signed message to its all associated subscribers. Interestingly, in the measurement, we notice that a REMP+ message broker shows similar performance to a REMP message broker that has 1500 subscribers.

VI. Conclusion

In this work, we show that conventional security schemes do not meet the security requirements of smart grid as a large-scale CPS. We propose REMP and REMP+ that achieve scalability and thinness, without compromising on the E2E security strength[3], and also account for resource-constrained devices in terms of communicating and computing capability, *e.g.*, low-powered sensors using PLC. Specifically, REMP is designed for supporting secure information collection (*i.e.*, smart metering) from a large number of devices, and REMP+ is designed for supporting secure multicast of a query message (or a control command) to a large number of devices. Lastly, we emphasize that our approaches are well aligned to NIST guidelines for smart grid security, as described in Sec III.A.

REFERENCES

[1] E. Lee, "Cyber physical systems: design challenges," in *Proc. IEEE Symposium on Object Oriented Real-Time Distributed Computing*, May 2008.
[2] M. Ilic, L. Xie, U. Khan, and J. Moura, "Modeling of future cyber physical energy systems for distributed sensing and control," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, pp. 825 –838, Jul. 2010.
[3] S.Kent, "IP encapsulating security payload," *IETF* RFC 4301.
[4] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.3," *IETF* draft-ietf-tls-tls13-03, Oct. 2014.
[5] A. Perrig, R. Canetti, D. Song, and J.D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. Internet Society NDSS*, Feb. 2001.
[6] Y.-J. Kim, V. Kolesnikov, and M. Thottan, "Resilient end-to-end message protection for large-scale cyber-physical system communications," in *Proc. IEEE SmartGridComm.*, Oct. 2012.
[7] NIST, "Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography," *NIST special publication 800-56A*, Mar. 2006.
[8] NIST, "Role of IP in the smart grid," *NIST SGIP*, July 2011.
[9] NASPI, "Data bus technical specifications for North American synchophasor initiative network," *NASPI*, May 2009.
[10] Texas Instruments, "MSP430 product brochure," *ti.com*, 2014. [Online]. Available: http://www.ti.com/lit/sg/slab034y/slab034y.pdf.
[11] NIST, "Introduction to NISTIR 7628 guidelines for smart grid cyber security," *NIST*, Sep. 2010. [Online]. Available: http://www.nist.gov/smartgrid/upload/ni stir-7628_total.pdf.
[12] "Announcing the advanced encryption standard (AES)," *NIST* publication 197, Nov. 2001.
[13] J.H. Song, R. Poovendran, J. Lee, and T. Iwata, "The AES CMAC algorithm," *IETF* RFC 4493, June 2006.
[14] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, "Internet key exchange protocol version 2 (IKEv2)," *IETF* RFC 5996, Sep. 2010.
[15] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 2, June 2003.
[16] J. Bacon, D. M. Eyers, J. Singh, and P. R. Pietzuch, "Access control in publish/subscribe systems," *ACM DEBS*, July 2008.
[17] C. K. Wong, M. Gouda, and S. S. Lam "Secure group communications using key graphs," *IEEE/ACM Trans. Networking*, vol. 8, no. 1, Feb. 2000.
[18] N. Modadugu and E. Rescorla, "The design and implementation of datagram TLS," in *Proc. NDSS Symposium*, Feb. 2004.
[19] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The secure real-time transport protocol (SRTP)," *IETF* RFC 3711, Mar. 2004.
[20] P. Zimmermann, A. Johnston, and J. Callas, "ZRTP: media path key agreement for unicast secure RTP," *IETF* RFC 6189, Apr. 2011.
[21] O. Kolkman, W. Mekking, and R. Gieben, "DNSSEC operation practices," *IETF* RFC 6781, Dec. 2012.
[22] M. Badra and S. Zeadally, "Key management solutions in the smart grid environment," in *Proc. Joint IFIP WMNC*, Apr. 2013.
[23] N. Liu, J. Chen, L. Zhu, and Y. He, "A key management scheme for secure communications of advanced metering infrastructure in smart grid," *IEEE Transactions on Industrial Electronics*, Aug. 2012.
[24] J.Y. Kim and H.K. Choi, "An efficient and versatile key management protocol for secure smart grid communication," in *Proc. IEEE Wireless Communications and Networking Conference*, 2012.
[25] S. Das, Y. Ohba, M. Kanda, D. Famolari, and S.K. Das, "A key management framework for AMI networks in smart grid," *IEEE Communications Magazine*, vol. 50, no. 8, pp. 30–37, Aug. 2012.
[26] J. Xia and Y. Wang, "Secure key distribution for the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1437–1443, 2012.
[27] M. Badra and I. Hajjeh, "ECDHE_PSK cipher suites for transport layer security (TLS)," *IETF* RFC5489, Mar. 2009.
[28] W. Barker, "Recommendation for the triple data encryption algorithm block cipher," *NIST publication 800-67*, May 2008.
[29] Y-J. Kim, J. Lee, G. Atkinson, H. Kim, and M. Thottan, "SeDAX: a secure, resilient and scalable platform," *IEEE JSAC*, vol. 30, no. 6, July 2012.
[30] Y.-J. Kim, V. Kolesnikov, H. Kim, and M. Thottan, "SSTP: a scalable and secure transport protocol for smart grid data collection," in *Proc. IEEE SmartGridComm.*, Oct. 2011.
[31] Y.-J. Kim, V. Kolesnikov, and M. Thottan, "TSAF: tamper-resistant and scalable mutual authentication framework for plug-in EV charging," in *Proc. IEEE SmartGridComm.*, Oct. 2013.
[32] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The kerberos network authentication service (V5)," *IETF* RFC 4556, July 2005.
[33] A. Liu and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," in *Proc. ACM IPSN, SPOTS Track*, April 2008.
[34] B. Schneier, J. Kelsey, D. Whiting, and D. Wagner, C. Hall, and N. Ferguson, "Performance comparison of the AES submissions," in *Proc. the Second AES Candidate Conference*, April 1999.
[35] OpenSSL Toolkit. [Online]. Available: http://www.openssl.org.

[3] REMP can enhance protection offered by network intrusion detection systems (NIDS) that inspect all traffic to identify malicious attempts or policy violations, since it enforces communication paths and hides network structure. Further, E2E encryption will not hamper the performance of most NIDS, as REMP headers remain unencrypted and by contrast REMP payloads are checked by the application-layer for screening data anomaly. Finally, REMP authentication makes it harder to inject invalid messages into the network.