Preserving Data Integrity for Smart Grid Data Aggregation

Fengjun Li
Department of EECS
The University of Kansas
Lawrence, KS, 66045

Bo Luo
Department of EECS
The University of Kansas
Lawrence, KS, 66045

***Abstract*—In smart grid systems, secure in-network data aggregation approaches have been introduced to efficiently collect aggregation data, while preserving data privacy of individual meters. Nevertheless, it is also important to maintain the integrity of aggregate data in the presence of accidental errors and internal/external attacks. To ensure the correctness of the aggregation against unintentional errors, we introduce an end-to-end signature scheme, which generates a homomorphic signature for the aggregation result. The homomorphic signature scheme is compatible with the in-network aggregation schemes that are also based on homomorphic encryption, and supports efficient batch verifications of the aggregation results. Next, to defend against suspicious/compromised meters and external attacks, we present a hop-by-hop signature scheme and an incremental verification protocol. In this approach, signatures are managed distributedly and verification is only triggered in an *ex post facto* basis – when anomalies in the aggregation results are detected at the collector. The incremental verification process starts from the collector, and traces the anomaly in a breath-first manner. The abnormal node is identified within $O(logN)$ iterations. Therefore, the verification process is computationally inexpensive, while ensuring faithfulness and undeniability properties.**

I. Introduction

The smart grid is envisioned as the next-generation approach of intelligent electricity generation, transmission, distribution, consumption and control [1]–[3]. The advanced metering infrastructure (AMI) serves as an important component on the consumer side (household and local neighborhood) of the smart grid system. In AMI, smart meters equipped with computing and communication capabilities are deployed in households. They are connected with the utility company through local collector devices (a.k.a. concentrators), to collect and monitor instant usage and status information (e.g. real-time power consumption data). They are also expected to distribute dynamic pricing and remote control information to support smart energy consumption in smart appliances.

In smart grids, information aggregation is an essential function for monitoring power consumption, load balancing, resource allocation, etc [4]. Aggregation data are collected frequently (e.g. at seconds level) to support intelligent electricity distribution and management; meanwhile, it also introduces new security and privacy challenges [5]–[7]. It is critical to transmit metering data from distributed smart meters to the control center at utility in a *secure* and *privacy-preserving* manner. That is, accurate readings need to be collected without being intercepted, altered, or forged; while private usage data and behavioral patterns are protected from being revealed to irrelevant parties en-route. In [8], [9], a secure and efficient information aggregation approach is proposed for smart grid systems. It proposes an in-network aggregation mechanism that performs aggregation tasks en route, to reduce computation and communication costs, and to avoid bottlenecks at the collector. To prevent meters enroute from seeing intermediate results, it employs homomorphic cryptosystems to encrypt metering data in the aggregation, while allowing arithmetic operations to be conducted on the ciphertext domain.

However, the solutions proposed in [8], [9] adopt the "honest-but-curious" model to assume that all the smart meters follow the protocol properly. Although it protects data privacy against curious smart meters, it does not consider accidental errors or cyber-attacks that tamper with the protocol. Therefore, it is vulnerable to unintentional errors (e.g. accidental errors in network transmission, storage and computing) and compromised meters or communication channels. For instance, a malfunctioning meter may accidentally produce errors in computing the aggregation; a compromised meter may drop intermediate aggregation results, and submit a random value to its parent node; an adversary who has hijacked the connection between two meters may inject fake data into the aggregation. In these attacks, external adversaries tamper with the aggregation process, expecting to mess up with load balancing, resource allocation and smart pricing.

To protect data integrity against accidental errors, we first introduce an end-to-end authentication scheme that is compatible with the homomorphic encryption based in-network aggregation schemes proposed in [8], [9]. In particular, a homomorphic signature is generated for the aggregated metering data at each intermediate node along with the aggregation process. In the end, the collector could effectively verify the correctness of the aggregation by checking the consistency between the aggregation result and the aggregation signature. The homomorphic signature scheme requires no decryption and re-encryption at intermediate meters, to facilitate an efficient signing/verification process. Moreover, to defend against fake data injection attacks, we present a hop-by-hop signature and incremental verification scheme. In this solution, aggregated outputs from smart meters are signed, and signatures are managed in a distributed manner (instead of transmitted to the

978-1-4673-0911-0/12/$31.00 ©2012 IEEE

collector on-the-fly). Verification is only performed in an *ex post facto* basis, when anomalies in the aggregation results are detected at the collector. The incremental verification process efficiently traces the anomaly in a breath-first manner, which is computationally inexpensive. More importantly, it ensures faithfulness and undeniability properties, so that the faulty nodes are always identified with undeniable evidences.

The rest of the paper is organized as follows: we summarize related works in Section 2, and briefly introduce the preliminaries in Section 3. We present our solutions in Section 4, and finally conclude the paper in Section 5.

II. Related Works

Cyber security is considered as one of the biggest challenges in smart grid systems [5]–[7], [10]. A comprehensive survey is available at [11]. In this work, we are particularly interested in securing data aggregation in the Advanced Metering Infrastructure (AMI) in smart grid systems.

To protect the privacy of metering data, Efthymiou et al. proposed an anonymization based approach to hide the identity of smart meters in high-frequency metering data using pseudonyms [12]. Instead of using costly trusted third party to anonymize the metering data, a more efficient approach is to hide individual data via aggregation. Garcia et al. proposed a no leakage protocol to aggregate partial shares of smart meter readings in a neighborhood using an additively homomorphic encryption scheme [13]. However, the approach is not scalable due to the high communication overhead. Recently, Li et al. presented a distributed in-network aggregation approach [8], [9], similar to the in-network aggregation approaches in wireless sensor networks [14]–[17], to efficiently aggregate smart metering data along a spanning tree. Differing from the wireless sensor network approaches that focus on defending against misinformation, the in-network aggregation solutions in smart metering aim to protect end-to-end data confidentiality and privacy against malicious or "curious" meters en route, but neglect authentication mechanisms for data integrity protection. To address the problem, simple authentication schemes based on conventual PKI digital signature scheme [18] or cryptographic MAC [19] have been proposed. However, they are either not compatible with the privacy-preserving in-network data aggregation or introduce excessive hop-by-hop verification overhead. Therefore, we present a new homomorphic signature based authentication scheme that can efficiently re-generate signatures for aggregation results at intermediate meters but also support batch verification at the collector. Homomorphic signature scheme was first proposed in [20] to authenticate packets in network coding protocols [21], [22] and later extended to applications as delegatable data sharing and data outsourcing [23].

III. Preliminaries

A. In-network information aggregation for smart grids

An in-network information aggregation approach has been proposed for smart grid systems [8], [9]. It first constructs an aggregation tree, which is a spanning tree of the graph that covers all the smart meters in the neighborhood. The aggregation tree roots at the collector node, which initiates aggregation tasks and receives final results. To reduce computation and communication overhead, each smart meter collects data (e.g. readings) from its children in the aggregation tree, performs aggregate operations with its own data, and submits the results to its parent node in the tree. Hence, aggregation is performed en-route, instead of having each smart meter establish a peer-to-peer connection with the collector device.

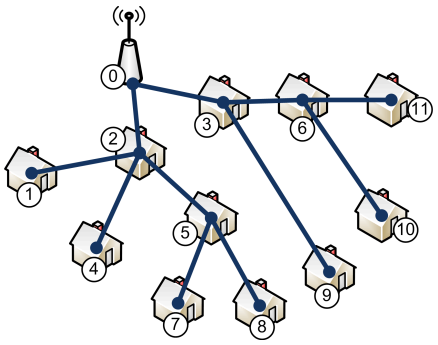

Fig. 1. An example of in-network aggregation for smart grids in a neighborhood

Example 1 *Figure 1 shows an example of a neighborhood, in which each house is equipped with a smart meter, and a collector (N_0) is deployed to cover the area. The aggregation tree (rooting at N_0) is constructed to collect realtime data from the meters. For instance, N_5 computes aggregation of data from N_5, N_7 and N_8, and sends the result to N_2.*

To prevent an intermediate node from seeing plaintext inputs from its children, homomorphic cryptosystems (e.g. [24]–[27]) are employed to encrypt the messages, while still allowing arithmetic operations on the ciphertext domain.

Example 2 *In Figure 1, N_8 cannot pass the plaintext output to N_5 due to security concerns. Paillier cryptosystem [24] is employed in [8] to encrypt the agregated output from N_8 (i.e. $C_{o8} = \mathsf{Enc}(P_{o8})$; since N_8 is a leaf node, we have: $P_{o8} = P_8$). N_5 takes C_{o7} and C_{o8}, performs designated arithmetic operations with its own data on the ciphertext domain, and passes the encrypted results (denoted as C_{o5}) to N_2.*

For more details on secure information aggregation in smart grid systems, please refer to [8], [9], [28], [29].

B. Homomorphic signatures

Homomorphic encryption is employed in [8] to support aggregation operations on concealed data, so that data privacy is well protected from intermediate meters. The solution in [8] focuses on data confidentiality and privacy, but lacks the capability to verify data integrity. Conventionally, authentication and integrity check are supported by appending digital signatures to the data. However, due to the malleability property of homomorphism, homomorphic encryption based schemes do not provide non-repudiation and thus cannot support verification of individual inputs at either intermediate meters or

the final destination (i.e., the collector). Therefore, additional techniques are needed for signing multiparty metering data and evaluating the integrity of the aggregation results.

Conventional digital signature schemes involve two operations, sign and verify, based on a pair of public and private keys. Each smart meter can sign the message m_i with its private key sk_i and the collector can retrieve the public keys of the smart meter and verify the integrity of each message.

Recently, homomorphic signature schemes have been proposed to support multivariate polynomial evaluation [20]–[22]. In general, a *homomorphic signature scheme* allows to sign messages m_i in a message space M and apply admissible functions f to the signed messages. In particular, with a pair of public/private key (pk, sk), the signer can sign message $m_1, m_2 \in Z_q$ as $\sigma_1 = \mathsf{sign}(sk, m_1)$ and $\sigma_2 = \mathsf{sign}(sk, m_2)$, where σ_1 and σ_2 satisfy the following properties:

1) *Homomorphic verifiability*: given σ_i for m_i, a verifier can evaluate the correctness of multivariate polynomial functions $f(m_i)$ without knowing m_i;
2) *Non-malleability*: without the secret key sk, it is impossible to generate a valid signature σ' for message $m' = f(m_i)$.

In this work, we present an end-to-end authentication scheme based on homomorphic signatures. The basis of our approach is the bilinear map [21]. For cyclic groups G_1, G_2 and G_T of prime order q, a map $e : G_1 \times G_2 \longrightarrow G_T$ is a bilinear map if it satisfies the following properties:

1) *Bilinear*: for all $u \in G_1$, $v \in G_2$, and $a, b \in Z_q$, we have $e(u^a, v^b) = e(u, v)^{ab}$;
2) *Computable*: there exists an efficient computable algorithm to compute the map $e(u, v)$ for any $u \in G_1$ and $v \in G_2$;
3) *Non-degeneracy*: for the generator g of G_1 and $\hat{g}$ of G_2, $e(g, \hat{g}) \neq 1$.

IV. The Method

A. End-to-end signature for in-network aggregation

To provide *batch authentication* in in-network data aggregation, we propose an efficient homomorphic signature scheme, similar to the short signature scheme in [20], [21] based on bilinear maps. Homomorphic authentication allows a verifier to check the integrity of the aggregation results without viewing the individual metering data involved in the aggregation. Therefore, at each relay meter, only one signature is generated and appended to the aggregation result, and at the collector, it takes only one operation to verify the entire batch of the metering data from the smart meters enroute.

The proposed scheme includes three basic components: *KeyGeneration*, *Signing*, *Verification*. More specifically, for two cyclic groups of order q, G and G_T, let g be the generator of G, then we have a bilinear map $e : G \times G \longrightarrow G_T$. Assume H is a collision-resistant hash function where $H : \{0,1\}^* \longrightarrow G$.

KeyGeneration: The key generation algorithm creates a pair of public and private keys to be used in the homomorphic signature scheme. With the above parameter settings, the tuple $< e, q, G, G_T, g, H >$ are the global parameters known by all the parities in the scheme. A random $a \in Z_q$ is selected as the private key sk shared by all the smart meters, and the corresponding public key is generated as $pk = g^a$.

Notations	Definitions
N_0	the collector, i.e., root of the spanning tree
N_i and ID_i	smart meter in the NAN and its unique identifier
P_i and P_{o_i}	input and aggregation output of N_i in plaintext
C_i and C_{o_i}	homomorphic encrypted form of P_i and P_{o_i}
σ_{agg_i} and σ_{o_i}	aggregation signature and output signatire for C_{o_i}
T_i	timestamp

TABLE I
NOTATIONS USED IN THE EXAMPLES.

Signing: For smart meter N_i (whose unique identifier is ID_i), let $C_{o_i} \in Z_q$ be the encrypted form of the plaintext output $P_{o_i} \in Z_q$ after homomorphic encryption for in-network aggregation. N_i computes $h_i = H(ID_i)$ and outputs the signature $\sigma_{agg_i} = (h_i C_{o_i})^a \in G$ for $< ID_i, C_{o_i} >$.

Verification: With the public key pk, an encrypted message C_{o_i}, an identifier ID_i, and a signature σ_{agg_i}, the verification is performed by checking if $e(\sigma_{agg_i}, g) = e(h_i C_{o_i}, pk)$, based on the bilinearity property such that $e((h_i C_{o_i})^a, g) = e(h_i C_{o_i}, g^a)$.

Similar to other signature schemes based on bilinear map, the proposed scheme provides full homomorphism in signature verification, which is also compatible with the in-network aggregation schemes [8] based on homomorphic encryption. In particular, consider a relay meter N_s who receives a set of encrypted messages $\{C_{o_1}, ..., C_{o_n}\}$ and the corresponding signatures $\{\sigma_{agg_1}, ..., \sigma_{agg_n}\}$ from its child nodes $\{N_1, ..., N_n\}$. N_s aggregates its own input C_s to generate an aggregation output $C_{o_s} = C_{o_1} \oplus ... C_{o_n} \oplus C_s$ and an aggregation signature $\sigma_{agg_s} = (h_s C_{o_s})^a$, where "$\oplus$" denotes the homomorphic operators in ciphertext domain (in [8], "$\times$" is considered for "$\oplus$"). Then, any node can verify the correctness of the intermediate aggregation output of N_s, C_{o_s}, by checking if $e(\sigma_{agg_s}, g) = e(h_s C_{o_s} \prod_{j=1}^{n} h_j, pk)$.

If the verifier is the collector, it enables an efficient *batch verification* for all the messages included in the aggregation. Such end-to-end integrity check is able to ensure the correctness of the aggregation results against accidental communication or meter errors. However, as we will see, it is insufficient in the presence of cyber-attacks.

B. Security vulnerabilities

As we have introduced, the in-network aggregation approach [8], [9] and the batch verification scheme proposed above do not provide the capability to verify the inputs from individual nodes. That is, they adopt the honest-but-curious model to assume that all the participating nodes properly follow the protocol, without any attempt to alternate or forge inputs/outputs. However, a malfunctioning or compromised smart meter, or an outside attacker, may generate fake data to tamper with the aggregation process.

Example 3 (Compromised meters.) *Smart meters in smart grid systems are vulnerable to a number of attacks [5], [30]. When smart meter N_6 in Figure 1 is compromised, it may report very large realtime data to N_3, expecting to mess with real-time load balancing and pricing. N_3 continues with the aggregation without knowing that the data has been exaggerated, since it only sees encrypted data. When N_0 decrypts the result, it may be able to recognize that input from N_3 is abnormal; however, it is unable to identify whether N_3 or any of its descendants produced the fake data.*

Example 4 (Compromised communication.) *Instead of directly attacking smart meters, external attackers can tamper with the communication between smart meters (especially the vulnerable wireless communication channel typically in SG communication). For instance, man-in-the-middle (MITM) attacks, replay attacks or reflection attacks could be launched in the Advanced Metering Infrastructure (AMI) [30]–[32]. Attackers could further inject fake data into the system, using forged identity and compromised communication channels.*

Although well designed authentication and key management schemes could be adopted to defend against attacks on network communication (e.g. [33]), integrity check at the application layer is still expected in various tasks. Moreover, as shown in the example, compromised smart meters could always inject forged data into the aggregation without being detected by other smart meters or the collector device.

The goal of the paper is to provide information accountability in the in-network information aggregation mechanism in smart grids. In particular, we provide novel designs so that: (1) the collector device is enabled to check the validity of inputs from individual nodes, when anomaly is detected in the inputs; and (2) smart meters tampering with the protocol are held accountable (or undeniable) for the forged inputs; and (3) the computation and communication overhead are minimized, especially, to avoid bottlenecks.

C. Preliminaries: anomaly detection

In in-network aggregation [8], the collector (e.g. N_0 in Figure 1) issues queries for aggregation tasks, and receives results that are distributedly computed from inputs from all (or a subset of) the smart meters in the neighborhood. A straightforward solution for authentication and data integrity is to let the collector perform verification for every aggregation result it receives. However, this is impractical because of the computation and communication overhead. In practice, the collector examines every aggregation result locally, assesses the likelihood of anomalies, and only calls the verification process for abnormal results.

Most of the aggregation tasks are repetitive, thus, the results are viewed as time series data. Temporal patterns could be learned from such data, and abnormal points could be identified. For instance, in [34], a *Hidden Markov Model* (HHM) was used to model real time electricity pricing data received by smart household appliances. Abnormal data (possibly injected by attackers to trick the appliances) are detected by a *dynamic Bayesian network* implemented with particle filtering. Further details of anomaly detection mechanisms in time series data is outside the scope of this paper. Please refer to [35]–[38] for more statistical learning methods for anomaly detection, and [31], [32] for more on intrusion detection in smart grid systems and AMI.

As we have mentioned, when a node N_i tampers with the aggregation process by injecting a $\Delta d_i \in (-\infty, \infty)$ in its output to the parent node, none of the smart meters enroute will be able to detect the forged data; meanwhile, the controller is unable to identify the source of the altered data, even though it identifies that the result is abnormal. In practice, when a compromised node injects a large Δd_i, it is easier for the collector to detect the intrusion, and drop the abnormal data. Meanwhile, such intrusion becomes harmful if it is undetected. On the contrary, a small Δd_i is less likely to be detected, however, it is also less harmful to the grid.

D. Incremental verification

Intuitively, data verification requires each node to sign its data and transmit the signature with the data to the collector for evaluating the input against the signature. However, this approach will not work for in-network aggregation since the collector only receives the aggregation result but not individual data from each smart meter. If asking smart meters to do so, it will essentially compromise all the advantages of the in-network aggregation approach by sending a message from each smart meter to the collector. It becomes a paradox of eliminating individual sessions between the collector and the smart meters, while achieving individual verifiability/undeniability.

To tackle the problem, we present an *incremental verification* approach, which stores digital signatures along the aggregation path, instead of sending them to the collector instantly. In our solution, each node sends ciphertext output along with a timestamped signature to the parent node, while the parent node verifies the signature and stores it locally. The signature is only transmitted to the collector on-demand, i.e., when the collector detects inconsistency or suspicious data, it requests for signatures for incremental verification and intrusion localization. The detailed protocol is as below:

KeyGeneration: The key generation algorithm remains largely the same to select $sk = a \in Z_q$ for all smart meters and $pk = g^a$ as its public key for verifiers. Moreover, each smart meter is assumed to generate its own private/public key (sk_i, pk_i) shared between neighboring nodes.

Signing: In the incremental protocol, a smart meter N_i generates two signatures, the *aggregation signature* σ_{agg_i} and the *output signature* σ_{o_i}, for its encrypted aggregation output C_{o_i}. σ_{agg_i} remains the same as $(h_i C_{o_i})^a$, and $\sigma_{o_i} = \mathsf{sign}(sk_i, h_i || C_{o_i} || \sigma_{agg_i})$, where $h_i = H(ID_i || T_i)$ and "$||$" denotes concatenation. Any PKI based signature scheme works for $\mathsf{sign}(\cdot)$. Here, we use $\sigma_{o_i} = (H(h_i || C_{o_i} || \sigma_{agg_i}))^{sk_i}$. Then, N_i sends σ_{agg_i} and σ_{o_i} along with C_{o_i}, ID_i, and T_i to its parent node N_j.

Verification: The verification supports three functions, 1)*per-hop verification*, 2)*signature table*, and 3)*batch verification*:

1) N_j takes N_i's public key pk_i to verify the encrypted input C_{o_i} and the aggregation signature σ_{agg_i} by computing $h'_i = H(ID_i||T_i)$, and comparing if $e(\sigma_{o_i}, g) = e(H(h'_i||C_{o_i}||\sigma_{agg_i}), pk_i)$.
2) If local verification succeeds, N_i stores the tuple $< ID_i, T_i, C_{o_i}, \sigma_{agg_i}, \sigma_{o_i} >$ to its local table. Assume that on average the collector receives and detects any abnormal input within T_0 seconds, we set a live period $T_{live} = T_0$, and drop any stored signature at t when $t > T_i + T_{live}$ to avoid unnecessary storage waste.
3) For any received aggregation result C_{o_x} and the corresponding aggregation signature σ_{agg_x}, the collector N_0 (and any node enroute if necessary) can perform batch verification by checking if $e(\sigma_{agg_x}, g) = e(C_{o_x} \prod_{N_j \in \{N_x\}} H(ID_j), pk)$, where $\{N_x\}$ is the set of nodes that are included in this aggregation task.

Incremental.Verification: Once the anomaly detection mechanism at the collector N_0 identifies a suspicious input, N_0 initiates the *incremental verification* process, which propagates to descendent nodes until the faulty node (i.e., a node whose output is abnormal, but all the outputs from its direct children are normal) is identified. Assume N_x is a direct child of N_0, the collector verifies the validity of the message received from N_x as follows:

1) N_0 first checks the correctness of C_{o_x} against σ_{o_x} according to step 1) of the **Verification** algorithm, and the timestamp to ensure it is a valid input.
2) N_0 recovers P_{o_x} that is the plaintext aggregation result for the subtree rooting at node N_x as: $P_{o_x} = \mathsf{Dec}(sk_0, C_{o_x})$, where sk_0 is the private key of the homomorphic encryption scheme for in-network aggregation (see [8] for details).
3) N_0 calls the anomaly detection module to verify the validity of P_{o_x}. If P_{o_x} is valid, the entire subtree rooting at N_x is considered valid, and N_0 continues with the next sibling node of N_x.
4) Otherwise, the faulty node is considered within the subtree rooting at N_x so that N_0 requests N_x to submit the local signature tuples $< ID_i, T_i, C_{o_i}, \sigma_{agg_i}, \sigma_{o_i} >$ that N_x obtained from its direct child nodes $\{N_i\}$.
5) N_0 verifies the correctness of the tuple according to **Verification**.(1), and decrypts C_{o_i} to get P_{o_i}, which is the plaintext aggregation result for the subtree rooting at node N_i. Then, N_0 repeats step (3) to call the anomaly detection module to verify the validity of P_{o_i}.
6) N_0 repeats steps (3) to (5) until the abnormal input is located.

The proposed verification protocol is efficient that only $O(logN)$ iterations are needed to locate a faulty node in a tree of N nodes. In particular, when the aggregation output at node N_x (i.e., P_{o_x}) is confirmed to be valid, all the descendants of N_x are eliminated from the verification process. Meanwhile, with a slight modification of the protocol, N_0 could opt to verify the input at each smart meter, instead of checking the aggregated output. To do so, the collector calls the reverse of the aggregation function to recover the input (P_x) at N_x from its output (P_{o_x}) and the output of its child nodes ($\{P_{o_i}\}$).

The incremental verification mechanism enables that all the inputs from the smart meters are verifiable – the collector device verifies the inputs to the aggregation task on an *ex post facto* basis. More importantly, the proposed verification protocol ensures two important properties:

1) **Faithfulness.** Since the signature σ_{o_i} is collected in the aggregation process and stored at the parent node, it faithfully demonstrates the authenticity of C_{o_i}, which is precisely the data injected into the aggregation by node N_i. The smart meters cannot "lie" in the verification process by injecting forged C_{o_i} in the aggregation but submitting a different signature in the verification process.
2) **Undeniability.** The signature σ_{o_i} is encrypted by the private key of node N_i, which is only known to itself. No other node is capable of producing a valid signature for a forged C'_{o_i}. Therefore, the signatures serve as undeniable evidence when the collector accuses a smart meter for injecting suspicious data, i.e., N_i is guaranteed to be the source of C_{o_i} once a signature is verified.

Last but not least, when compromised or faulty meters are identified, it is important to temporarily eliminate them from future aggregation tasks. To isolate a node N_x, the aggregation tree needs to be reconstructed: (1) direct child nodes of N_x are moved away from N_x to become children of normal nodes; (2) N_x is disconnected from its parent node in the aggregation tree. Depending on the type of the attack, new keys may need to be regenerated/redistributed.

E. Discussions

In industrial adoption of smart grid systems, cost is one of the major factors to be considered. In practice, computing power in state-of-art smart meters are not fully utilized, hence, it is possible to accommodate applications with moderate computation. On the contrary, communication capability is limited and the cost of communication is more sensitive.

The in-network information aggregation and end-to-end signing approaches save communication costs by eliminating overlapped links and conducting aggregations en-route. They also avoid bottlenecks (at the collector device) by distributing computations to smart meters. It is a big waste if the collecting device collects all the signed verification messages along with the aggregation message. With the incremental verification scheme, the collector only initiates the (expensive) verification protocol when anomalies are detected. The verification process efficiently traces the abnormal data, and terminates when the faulty meter is identified. In normal states, signatures are not transmitted to the collector, which effectively saves network resources. The computation and communication overhead introduced by the signing process (i.e., Protocol.Signing) is similar to the overhead of the aggregation process, which

is the price for ensuring the faithfulness and undeniability properties.

V. Conclusion

The secure in-network information aggregation mechanism for smart grid systems [8] does not have capabilities for integrity check. Hence, it is vulnerable to accidental errors, as well as compromised/dishonest meters and other fake data injection attacks. In this paper, we first introduce an end-to-end signature scheme using homomorphic signatures. A checksum of the aggregation is generated and updated along with the in-network aggregation process. With minimum overhead, it enables the collector to check the integrity of the aggregation result. However, such mechanism becomes insufficient in the presence of cyber-attacks, i.e., when forged data is injected by compromised meters or communication channels. We further present a hop-by-hop signature scheme and an incremental verification mechanism to defend against such attacks. In this solution, output from each smart meter is signed, and signatures are kept at parent nodes. The collector device initiates an incremental verification of signatures when suspicious aggregation results are received. The *ex post facto* verification process is computationally inexpensive, while ensuring faithfulness and undeniability properties.

References

[1] S. Massoud Amin and B. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *Power and Energy Magazine, IEEE*, vol. 3, no. 5, pp. 34 – 41, sept.-oct. 2005.

[2] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, 2010.

[3] M. Hashmi, S. Hanninen, and K. Maki, "Survey of smart grid concepts, architectures, and technological demonstrations worldwide," in *IEEE PES Conference on Innovative Smart Grid Technologies*, 2011.

[4] W. H. Sanders, "Progress towards a resilient power grid infrastructure," in *Proceedings of the IEEE Power & Energy Society General Meeting (PES GM)*, July 2010.

[5] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *Security Privacy, IEEE*, vol. 7, no. 3, pp. 75 –77, 2009.

[6] H. Khurana, M. Hadley, N. Lu, and D. Frincke, "Smart-grid security issues," *Security Privacy, IEEE*, vol. 8, no. 1, pp. 81 –85, jan.-feb. 2010.

[7] A. R. Metke and R. L. Ekl, "Security technology for smart grid networks," *Smart Grid, IEEE Transactions on*, vol. 1, no. 1, pp. 99 –107, june 2010.

[8] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *SmartGridComm, 2010 First IEEE International Conference on*, oct. 2010, pp. 327 –332.

[9] ——, "Secure and privacy-preserving information aggregation for smart grids," *Int. J. Secur. Netw.*, vol. 6, no. 1, pp. 28–39, 2011.

[10] A. Metke and R. Ekl, "Smart grid security technology," in *Innovative Smart Grid Technologies (ISGT)*, jan. 2010, pp. 1 –7.

[11] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid - the new and improved power grid: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 99, pp. 1–37, 2011.

[12] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," *2010 First IEEE International Conference on Smart Grid Communications*, pp. 238–243, 2010.

[13] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Proceedings of the 6th international conference on Security and trust management*, ser. STM'10, 2011, pp. 226–238.

[14] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, 2002, pp. 575–578.

[15] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, 2002.

[16] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks," in *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 278–287.

[17] K. B. Frikken and J. A. Dougherty, IV, "An efficient integrity-preserving scheme for hierarchical sensor aggregation," in *WiSec*, 2008, pp. 68–76.

[18] P. Deng and L. Yang, "A secure and privacy-preserving communication scheme for advanced metering infrastructure," *Innovative Smart Grid Technologies, IEEE PES*, vol. 0, pp. 1–5, 2012.

[19] Y. Yan, Y. Qian, and H. Sharif, "A secure and reliable in-network collaborative communication scheme for advanced metering infrastructure in smart grid," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, 2011, pp. 909 – 914.

[20] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *ASIACRYPT '01*.

[21] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *EUROCRYPT'03*.

[22] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *INFOCOM 2008*.

[23] M. Barbosa and P. Farshim, "Delegatable homomorphic encryption with applications to secure outsourcing of computation," in *Proceedings of the 12th conference on Topics in Cryptology*, 2012, pp. 296–312.

[24] P. Paillier, "Public-key cryptosystem based on composite degree residuosity classes," in *Proceedings of Eurocrypt '99*, 1999, pp. 223–238.

[25] P. Paillier and D. Pointcheval, "Efficient public-key cryptosystems provably secure against active adversaries," in *Advances in Cryptology – Proceedings of Asiacrypt '99*. Springer-Verlag, 1999, pp. 165–179.

[26] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proceedings of CRYPTO 84 on Advances in cryptology*. Springer-Verlag New York, Inc., 1985, pp. 10–18.

[27] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Proceedings of Theory of Cryptography (TCC)*, 2005, pp. 325–341.

[28] X. He, M.-O. Pun, and C.-C. Kuo, "Secure and efficient cryptosystem for smart grid using homomorphic encryption," in *IEEE PES Conference on Innovative Smart Grid Technologies (ISGT)*, 2012.

[29] S. Ruj, A. Nayak, and I. Stojmenovic, "A security architecture for data aggregation and access control in smart grids," http://arxiv.org/abs/1111.2619, 2011.

[30] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy theft in the advanced metering infrastructure," in *CRITIS'09*.

[31] F. M. Cleveland, "Cyber security issues for advanced metering infrastructure (ami)," in *Proceedings of the IEEE Power and Energy Society General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century*, 2008.

[32] R. Berthier, W. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010.

[33] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 375–381, 2011.

[34] Y. Chen and B. Luo, "S2a: secure smart household appliances," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, ser. CODASPY '12, 2012, pp. 217–228.

[35] D. Dasgupta and S. Forrest, "Novelty detection in time series data using ideas from immunology," in *Proceedings of The International Conference on Intelligent Systems*, 1995.

[36] S. Salvador and P. Chan, "Learning states and rules for detecting anomalies in time series," *Appl. Intell.*, vol. 23, no. 3, pp. 241–255, 2005.

[37] E. Keogh, J. Lin, and A. Fu, "Hot sax: Efficiently finding the most unusual time series subsequence," in *Proceedings of the Fifth IEEE International Conference on Data Mining*, ser. ICDM '05, 2005, pp. 226–233.

[38] L. Wei, N. Kumar, V. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Assumption-free anomaly detection in time series," in *SSDBM'05*, 2005, pp. 237–242.