Preserving Data Integrity for Smart Grid Data Aggregation

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

The smart grid is envisioned as the next-generation approach of intelligent electricity generation, transmission, distribution, consumption and control. 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. realtime 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 saving communication cost, monitoring power consumption, load balancing, resource allocation, etc. Meanwhile, it also introduces new security and privacy challenges. Private usage data and behavioral patterns need to be protected from being revealed to irrelevant parties en-route. Thus, secure homomorphic encryption based data aggregation approaches have been introduced to efficiently collect aggregation data. Nevertheless, it is also important to maintain the integrity of aggregate data in the presence of accidental errors and internal/external attacks. Accurate readings need to be collected without being altered or forged. In smart grids, values received by the collector provide a basis for critical decisions; hence, false or biased values may cause catastrophic consequences.

To protect data integrity against accidental errors, we first introduce an end-to-end authentication scheme that is compatible with the homomorphic encryption based aggregation schemes. In particular, a homomorphic signature is generated for the aggregated metering data at each intermediate node during the data aggregation process. In the end, the collector could effectively verify the correctness of the aggregation by checking the consistency between the aggregation result and its corresponding signature.

However, a malfunctioning or compromised meter, or an outside attacker, may generate fake data to tamper with the aggregation process. A malfunctioning/compromised node may launch false data injection attacks by changing the data from its children or forging its own value. In order to defend against false data injection attacks and identify the compromised node, we propose an Incremental Verification scheme to perform a two-phase verification for each suspicious node. The scheme creates a hop-by-hop signature and stores them at each intermediate node. Verification is only triggered in an *ex post facto* basis through a top-down process. The collector recovers the corresponding plaintext for each aggregated result from its children and calls the anomaly detection module to verify the validity of the plaintext. If the collector detects an anomaly, it launches the first-phase verification: first, it requests all children of the suspicious node to submit their aggregation outputs. Then it recomputes the aggregated result and signature by itself and compares them. If the check succeeds, we consider that the suspicious node doesn't change its children's aggregation output, but it is still suspicious to forge its own reading. Thus, the collector starts second-phase verification for this suspicious node. It decrypts the input data of the suspicious node, which are aggregated output results of its children. If all children submit valid aggregate outputs, we consider the suspicious node as malfunctioning or compromised one. Otherwise, a node that submits invalid data is considered as the new suspicious node. The scheme recruits the nodes that pass previous verification as new verifiers which repeat the two-phase verification to verify the new suspicious node in a top-down manner until identifying the malicious node. All participants of the verification submit their aggregate data to the new verifier to perform the first-phase verification, thus, the verifying computation is distributed to the intermediate nodes without a centralized verifier. Our top-down verification scheme identifies the malfunctioning or compromised node without introducing high communication and computation overhead.