AMIsim: Application-layer Advanced Metering Infrastructure Simulation Framework for Secure Communication Protocol Performance Evaluation

Vitaly Ford, *Arcadia University (vitalyford@gmail.com, https://vford.me)*
Daniel Tyler, *Tennessee Tech University* Ambareen Siraj, *Tennessee Tech University (asiraj@tntech.edu)*

Abstract

The Advanced Metering Infrastructure (AMI) is a major component of the Smart Grid. Researchers have been working to protect its communication by designing protocols that offer security and privacy in various ways to different extents. Simulation testing is a crucial part of any communication protocol development. Current simulation frameworks for power Grid experiments primarily focus on simulating the electrical components and power flow in the Grid. In this paper, we introduce a uniform AMI simulation (AMIsim) framework for evaluating secure and privacy-preserving AMI protocols. AMIsim allows researchers to conduct a performance assessment of their application-layer security protocols that are used for aggregation, privacy-preservation, and confidentiality/integrity protection of smart meter energy data. We report on the empirical results of conducting experiments in AMIsim with an existing AMI secure and privacy-preserving protocol.

1. Introduction

With the explosion of Big Data, user privacy can no longer stay under the radar in data analytics [1]. The Smart Grid [2] generates an enormous amount of sensor data containing such information as voltage, current, power, and energy consumption. Based on the recent work in the area of privacy-preserving technologies [3, 4, 5, 6], consumer privacy has been identified as a priority issue in storing and analyzing data in the Smart Grid, including energy consumption measurements.

The Advanced Metering Infrastructure (AMI) is a major component of the Smart Grid. AMI incorporates smart meters, communication networks, and data management systems [2]. Smart meters are electrical meters that support two-way communication between them and utility companies.

Researchers have developed advanced privacy-preserving protocols, addressing the core issues of smart meter data communication and handling [7, 8, 9]. The rapid growth of such protocols needs an effective testing environment, where the protocols can be evaluated individually and comparatively.

Simulation testing is a crucial part of any communication protocol development. It allows to identify potential structural and implementation flaws in the protocol and fulfill initial performance evaluation, without the need to build a real prototype of the infrastructure to test the protocol. Also, simulation frameworks allow a protocol in question to be compared with the other existing protocols with regard to their utility and performance.

Current simulation frameworks [10, 11, 12, 13] for power grid experiments primarily focus on simulating the electrical components and power flow in the Grid. However, most of the privacy-preserving protocols address privacy issues with regard to consumer data, concentrating on transferring the energy consumption data rather than electrical signals and power. Such protocols primarily operate on the application layer of the Open Systems Interconnection (OSI) model [14]. Therefore, the protocol evaluation simulation framework must support the assessment conducted on the application layer in order to achieve the desired level of the protocol performance analysis. Additionally, the framework should facilitate understanding the limitations of the protocols under investigation.

This paper introduces a uniform Advanced Metering Infrastructure simulation (AMIsim) framework for evaluating secure and privacy-preserving protocols developed for the AMI environment. The proposed framework can be used to conduct a comparative analysis of various protocols' computational and communication performance. AMIsim offers an intuitive universal platform to conduct simulations of application-layer AMI protocols used in secure communication and collection of energy consumption readings.

AMIsim allows researchers to conduct a performance assessment of their application-layer security protocols that are used for aggregation, privacy-preservation, and confidentiality/integrity protection of smart meter energy consumption data. Based on the existing AMI privacy-preserving protocol [7], we demonstrate how the framework can be utilized to conduct protocols' performance evaluation.

2. Related Work

For the Smart Grid simulation frameworks that exist today, their primary goals are clustered around simulating the electrical components, power flow, and various attack schemes that disrupt them. Following is a brief discussion of some of the existing Smart Grid simulation frameworks.

The SGsim [10] analyzes the power flow and voltage in the Smart Grid, as well as phase measurements and optimization applications.

NeSSi2 [20] is a simulation framework developed for simulating power generation and energy consumption. NeSSi2 also provides a mechanism to perform energy-based attacks, such as falsely reporting of low energy consumption or prices.

The SmartGridLab framework [11] simulates power supply and demand, as well as real-time demand-response.

Mallapuram et al. [12] used the ns-3 [31] simulation tool to demonstrate the impact of different attacks on the Smart Grid infrastructure, simulating false-data injections, re-routing, and Denial of Service (DoS) attacks.

Yardley et al. [16] developed an ANSI C12.22 protocol dissector and a specification-based intrusion detection system and tested their tools on an AMI simulation testbed.

The PowerCyber testbed [13] simulates typical power flow in the Smart Grid, containing power systems, control centers, and substations. The PowerCyber uses the DNP3 [21] protocol for communication channels and can simulate DoS attacks.

Particularly for AMI, some publicly available simulation testbeds target AMI modeling in terms of the power flow and energy monitoring [22, 23, 24] but they do not focus much on the consumer data storage, transfer, and analysis.

3. Motivation

Most of the above mentioned open source simulation tools used in AMI protocol evaluation do not take into consideration the analysis of privacy-preserving application-layer protocols. They primarily focus on assessing electrical voltage and power variation in AMI. A prototype of AMI or a microgrid can be built [16], but it is neither practical nor feasible for every situation because of its cost and complicated structural characteristics. Therefore, there is a definite need for an AMI simulation framework that can evaluate and compare application-layer AMI protocols.

3.1. Design Goals

The design goals of the proposed AMIsim are as follows:

- It should be open source and available [19].
- It should provide a simple gateway to evaluate the implementation of AMI protocols in a simulated environment.
- Any AMI protocol can be assessed within the same environment, providing an unbiased and uniform way to compare the performance of different protocol implementations.
- It can be used to perform evaluation where computational constraints should be taken into account.

4. AMIsim Framework Architecture

The proposed AMIsim is designed and developed for analyzing any application-layer AMI security protocol in a simple and consistent manner. The core units/concepts in AMIsim are the smart meters, utility company, collector/aggregator, and trusted third party, which characterize a typical AMI environment [2]. The framework allows researchers to have some flexibility in customizing their own specialized AMI infrastructure based upon inherited abstract AMI properties, as well as adding their own AMI properties. The simulation environment feeds the Pecan Street dataset [15], which is the world's largest publicly available energy dataset. The dataset includes 15-minute and 1-minute interval circuit-level anonymized smart meter data from thousands of households in the Austin, TX area. In addition, as per need, the input data can be customized by researchers.

Each of the AMIsim core units has its own role with unique responsibilities, together providing an interface for building a customizable AMI. The generator creates realistic smart meter readings and distributes them across the smart meters. When a smart meter receives a new energy consumption reading, it performs the necessary computations according to the protocol under investigation and forwards the results of the computations to the neighborhood collector. The collector relays the received data to the utility company. Additionally, the collector's functionality can be expanded to simulate various attacks, such as replay or eavesdropping attacks.

After periodic data gathering from the smart meters, the collector forwards the data to the utility company. The utility company conducts the necessary analysis of the data, as per the requirements of the protocol under investigation. Afterwards, the utility company has a choice of sending the data to the trusted third party or keeping the measurements in the local database.

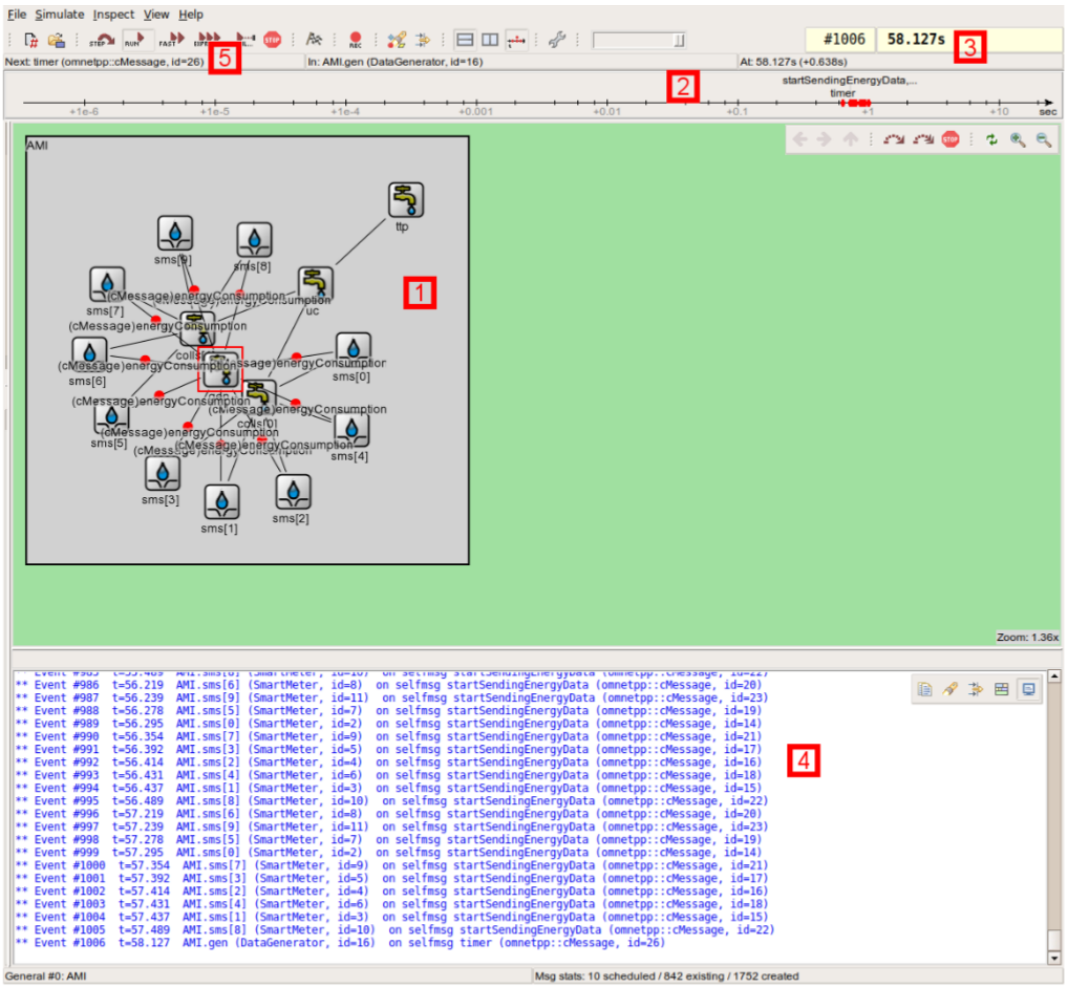


Fig. 1. Protocol evaluation in the framework: (1) visualization of the running simulation; (2) event timeline; (3) simulation timing; (4) event log; (5) simulation controls.

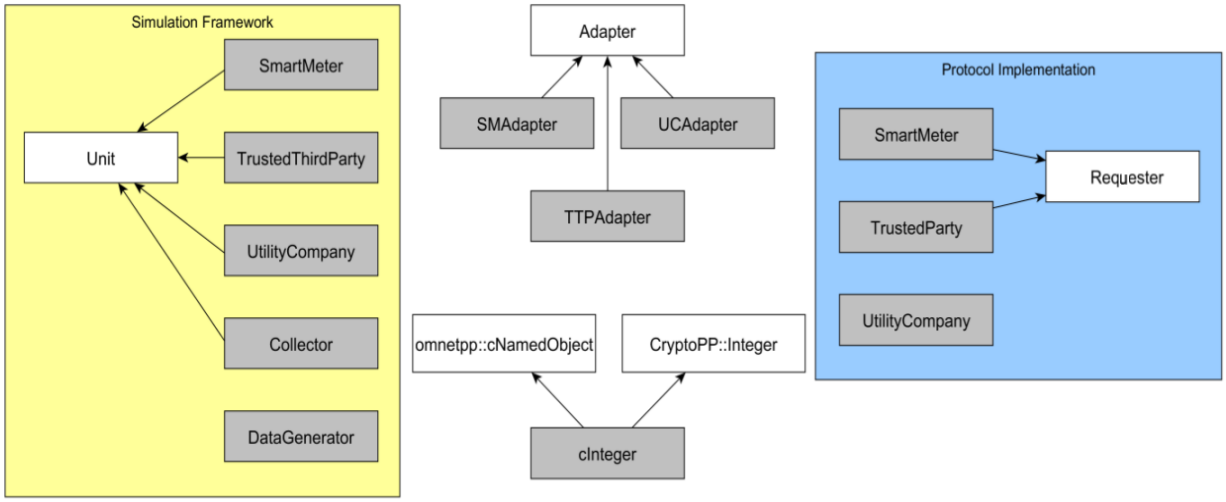


Fig. 2. Class hierarchy.

The selection depends on the protocol that is being tested in the AMIsim framework. An example of a running protocol evaluation is illustrated in Figure 1.

The features of the proposed AMIsim are as follows:

- It is a customizable, modular, and simple-to-use application-layer protocol simulation environment.
- It collects and produces the necessary performance data without user interaction.
- It provides opportunities for assessing network security against such attacks as fuzzing.

In order to develop a realistic smart metering network, we studied the literature to determine the typical network specifications used in AMI. The base wireless communication standard used in AMI is IEEE 802.15.4 [25]. One of the most common implementations of the standard is ZigBee [26]. To simulate a ZigBee network and better understand its performance, we utilized the data from the literature. IEEE 802.15.4 networks are expected to transmit data in a range from 10 to 75 meters [27]. According to the practical experiments in [27], the packet error rate of an IEEE 802.15.4 outdoor network at a distance of ~70 meters is approximately 0.1. Additionally, an average IEEE 802.15.4 data rate is ~163 kbps [28], whereas the maximum is 250 kbps.

- 1-minute interval data samples are fed to the smart meters.
- The trusted third party is created, which stores all the data and fulfills energy consumption analysis based upon granular energy measurements.
- The utility company is created, which connects the smart meters with the trusted third party.
- Smart meters are created as grouped together in neighborhoods. For each neighborhood, we create one collector that connects directly to the utility company. The collector can be adapted to work as an attacker if required in the experiment.
- Collectors forward all smart meter data to the utility company periodically in a certain preset interval as per the protocol under investigation (for example, every 15 minutes). Alternatively, the collectors can send the data at other intervals varying from 10 to 20 minutes, if needed, to decrease the load on the network. These intermittent intervals enable the collectors to have less interference with each other during data transfer.
- Time delays are created to simulate limited smart meter computational capabilities while executing the following smart meter activities:
    - Registering a new smart meter with the utility company.
    - Encrypting the energy readings by smart meters.
    - Sending smart meter data to a collector.

4.2. Implementation

AMIsim utilizes OMNeT++ [17] providing a discrete time-based simulation interface to evaluate any arbitrary protocol. OMNeT++ has been recommended by various experts after comparison with several different networking frameworks [18]. AMIsim provides adapter (wrapper) classes for an easy integration of AMI protocols in the testing environment.

AMIsim framework is developed in C++ language, using an Object-Oriented Programming approach. Figure 2 describes the object class hierarchy integrated into the framework. The object classes within the Simulation Framework category manage the higher-level network interactions among the core units of AMI. The object classes used in Protocol Implementation category are interchangeable, depending on the protocol under evaluation. The object classes at the center of Figure 2, which inherit from the Adapter class, are used to bridge the gap between the Simulation Framework and Protocol Implementation object types. The cInteger class is a helper class, which extends both the OMNeT++ cNamedObject class and the Crypto++ Integer class, providing a convenient mechanism to convert between those specific types. The AMIsim code can be found in [19].

As mentioned before, the AMI classes described above were developed based on OMNeT++. In OMNeT++, the network is defined in a special *NED* language (topology description language [32]). The *NED* file example of a network definition with 10 smart meters, 2 collectors, ZigBee and Wireless communication among the meters, collectors, utility company, and trusted third party is presented below.

4.3. NED File

```
network AMI
{
  parameters:
    int smNum = default(10);// Define 10 me-
ters
    int colNum = default(2);// Define 2 col-
lectors
    // Set a default delay for ZigBee
    volatile int smDelay @unit(ms) =
         default(exponential(100ms));
  types:
    // Datarate is 250 kbps max for ZigBee,
    // average is ~163 kbps on 70m distance
    channel ZigBee extends ned.DatarateChannel
    {
      delay = smDelay;
```

```
      datarate = 163kbps; // Set a default da-
tarate
      per = 0.1; // Set a default packet error
rate
    }
    channel Wireless extends ned.DatarateChan-
nel
    {
      delay = 1s;
      datarate = 30Mbps;
    }
  submodules:
    // Define an array of meters
    sms[smNum]: SmartMeter;
    // Define an array of collectors
    colls[colNum]: Collector;
    uc: UtilityCompany;// Define the utility
company
    // Define the trusted third party
    ttp: TrustedThirdParty;
    // Define the energy readings generator
    gen: DataGenerator;
  connections:
    for i=0..smNum-1 // For every meter, do
    {
      // Connect meters with the generator,
directly
      sms[i].generatorLine <--> gen.smLine++;
      // Connect meters with the collectors
      // via ZigBee
      sms[i].radio <--> ZigBee <-->
          colls[floor(i/(smNum/colNum))].ra-
dio++;
    }
    for i=0..colNum-1
    {
      // Connect collectors with the utility
company
      // via Wireless
      colls[i].ucLine <--> Wireless <-->
uc.radio++;
    }
    // Connect the trusted third party with
the
    // utility company, directly
    uc.ttpLine <--> ttp.ucLine;
}
```

4.4. Evaluation Metrics

There are several evaluation measurements that AMIsim can automatically collect for comparing the performance of various AMI security protocols. It can calculate the time taken by a smart meter to perform the required operations, as well as the communication overhead. It is important to note that smart meters are limited in their computational capabilities; therefore, computation time is the most significant performance metric that needs to be considered, when comparing various protocols. Additionally, the amount of data that needs to be sent across the network from the smart meters to the collector should be within acceptable margins in compliance with the limited smart meter storage and low-bandwidth wireless protocols, such as ZigBee.

The summary of the evaluation metrics that can be analyzed in the AMIsim framework is presented below.

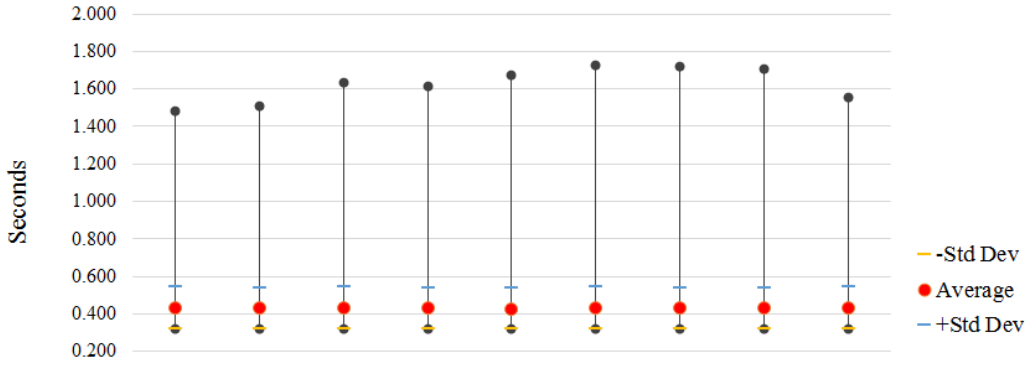
- The total packet size that a smart meter transmits.
- Encryption, decryption, and aggregation times taken by a smart meter.
- Congestion analysis based on the number of packets in a sending queue on the smart meter side.
- Feasibility of the protocol in terms of ZigBee network's throughput (the average size of the packet queue on the smart meter side) and error rate.

5. A Case Study: A Protocol's Performance Analysis

We conducted simulation experiments on a machine with Intel Core i7-3610QM 2.3 GHz processor. To simulate smart meter's limited computational capabilities, we calculated the number of clock cycles needed for a particular function with a certain set of tasks. Given that number of clock cycles and the average smart meter's processor speed (for instance, 120 MHz, 32-bit ARM Cortex-M4 [29]), we then compute the time it can take a smart meter to finish that specific function. ARM Cortex-M4 is approximately 100 times slower than Intel Core i7-3610QM (according to [30]) in performing floating-point operations per second (FLOPS). Therefore, we assume that a smart meter's processor is about 100 times slower than the processor that we have used for our experiments. To conduct experiments on machines with different processors, we recommend determining the difference between the processor that is being used and Intel Core i7-3610QM. That way one can apply that deviation to the constant (which was equal to 100 in our case) and identify how slower the smart meter would be in comparison with the processor that is being used.

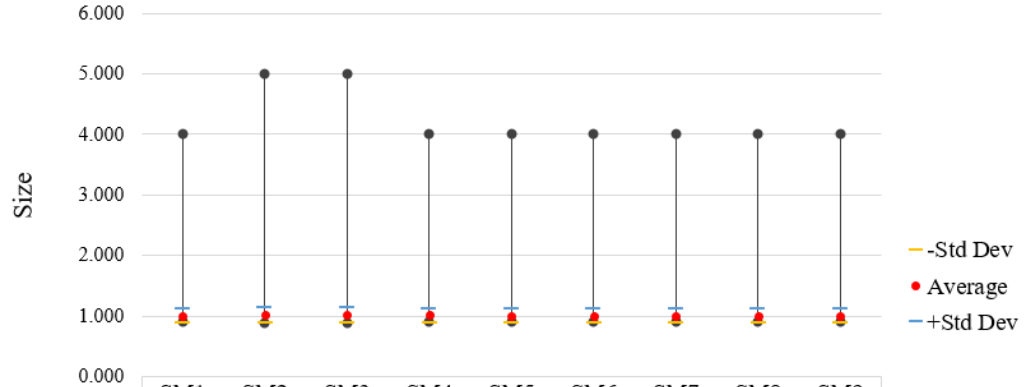
We selected a privacy-preserving AMI protocol that we had previously designed and developed [7] to evaluate its performance within AMIsim. We implemented the protocol with the appropriate classes using the framework. The simulation experiments contained two collectors and nine smart meters. It was not necessary to increase the number of meters and collectors because AMIsim is based on the discrete time-based simulator where a higher number of the meters would not influence the overall meters' performance metrics, as long as the number of collectors is increased as well.

While running the simulation, we collected data corresponding to the performance, particularly computational and communication overheads. Using the protocol, the total size of the packets sent by the smart meters at the data transmission phase was 300 bytes, which included encryption of the message, integrity verification, and timestamp.



	SM1	SM2	SM3	SM4	SM5	SM6	SM7	SM8	SM9
Minimum	0.343	0.341	0.343	0.340	0.340	0.344	0.343	0.342	0.342
-Std Dev	0.323	0.322	0.323	0.321	0.320	0.323	0.322	0.322	0.322
Average	0.433	0.431	0.432	0.430	0.429	0.433	0.432	0.431	0.432
+Std Dev	0.543	0.541	0.542	0.540	0.537	0.544	0.541	0.541	0.542
Maximum	1.483	1.510	1.631	1.617	1.677	1.726	1.718	1.705	1.554

Fig. 3. Computation overhead.



	SM1	SM2	SM3	SM4	SM5	SM6	SM7	SM8	SM9
Minimum	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
-Std Dev	0.899	0.888	0.894	0.897	0.897	0.902	0.904	0.897	0.903
Average	1.010	1.012	1.012	1.012	1.010	1.010	1.009	1.010	1.009
+Std Dev	1.121	1.137	1.130	1.126	1.124	1.117	1.115	1.124	1.116
Maximum	4.000	5.000	5.000	4.000	4.000	4.000	4.000	4.000	4.000

Fig. 4. Packet queue size.

Figure 3 illustrates different statistical metrics of the time taken to perform the necessary cryptographic computations by each of the nine smart meter's processor (SM1 to SM9) during the simulation in the AMIsim. As it can be seen that the average lies in the range of 0.431 to 0.433, which signifies that smart meters can successfully perform the necessary computations within a second.

After gathering the data corresponding to the computational overhead on the Intel processor, we multiplied the time measurements by 100 to simulate the ARM Cortex-M4 processor used in the smart meters that performs floating-point operations approximately 100 times slower than the processor that ran our simulation. We calculated statistical metrics including the minimum, average, maximum, and standard deviation from the mean, provided the time performance data that have been changed by 100. As a result, we deduced the approximate time it would take to run the necessary computations on the smart meters.

Figure 4 demonstrates different statistical metrics of the packet queue size on the smart meters due to simulated packet loss and network congestion in AMIsim with one-

second smart meter data transmission intervals. The packet error rate was set to 0.1 to represent the ZigBee's data transmission properties. It should be noted that the average packet queue size was nearly one during the simulation, meaning that the computational overhead did not cause any network congestion at the end nodes (smart meters). It can also be concluded that performance wise, the protocol under consideration is feasible to be deployed in ZigBee networks. The accuracy of the protocol under investigation is outside of the scope of this work and demonstrated in [7] for interested readers.

Conclusion and Future Work

The current simulation frameworks mostly focus on simulating the electrical components and power flow in the Smart Grid. However, there is a need for comparing the performance of privacy-preserving application-layer protocols in AMI. We have developed a simulation framework AMIsim based on OMNeT++. AMIsim allows comparing existing application-layer protocols in terms of their computational and communication performance. It provides a set of high level abstract conceptual modules to simulate communication among a utility company, trusted third party, and smart meters.

We performed experiments in AMIsim and demonstrated its effectiveness with the protocol under investigation. It can serve as a benchmarking tool for comparing application-layer AMI protocols' performance overheads.

As a future work, we plan to explore an integration of the AMIsim framework with other simulation frameworks to extend the existing features and share the developed framework with the Smart Grid community. We plan to conduct multiple experiments, where different existing protocols applicable in this environment will be implemented to demonstrate AMIsim's applicability in the protocol performance evaluation. Additionally, an assessment of network security against attacks such as fuzzing will be conducted in AMIsim.

Acknowledgment

We would like to thank the Center for Manufacturing Research, as well as the Cybersecurity Research and Outreach Center at Tennessee Tech University, for financial support during design and development of this project.

References

[1] A. Cavoukian, and J. Jonas. Privacy by design in the age of big data. Information and Privacy Commissioner of Ontario, Canada, 2012.

[2] The Department of Energy's Office of Electricity Delivery and Energy Reliability, "What is SG?". [Online]. URL: https://www.smartgrid.gov/

[3] K. Birman, et al. "Building a secure and privacy-preserving smart grid." ACM SIGOPS Operating Systems Review 49.1 (2015): 131-136.

[4] Y. Gong, et al. "A privacy-preserving scheme for incentive-based demand response in SG" IEEE Transactions on Smart Grid 7.3 (2016): 1304-1313.

[5] L. Chen, L. Rongxing, and C. Zhenfu. "PDAFT: A privacy-preserving data aggregation scheme with fault tolerance for SG communications." Peer-to-Peer networking and applications 8.6 (2015): 1122-1132.

[6] F. Farokhi, and S. Henrik. "Fisher information as a measure of privacy: Preserving privacy of households with smart meters using batteries." IEEE Transactions on Smart Grid (2017).

[7] V. Ford, A. Siraj, and M. A. Rahman. "Secure and efficient protection of consumer privacy in AMI supporting fine-grained data analysis." Journal of Computer and System Sciences 83.1 (2017): 84-100.

[8] S. Tonyali, et al. "Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled Smart Metering systems." Future Generation Computer Systems (2017).

[9] L. Zhu, et al. "Privacy protection using a rechargeable battery for energy consumption in smart grids." IEEE Network 31.1 (2017): 59-63.

[10] A. Awad et al. "SGsim: A simulation framework for smart grid applications." In "IEEE Energy Conference," pages 730-736. 2014.

[11] G. Lu et al. "Smartgridlab: A laboratory-based smart grid testbed." In "Smart Grid Communications, 2010 First IEEE International Conference on," pages 143-148. IEEE, 2010.

[12] S. Mallapuram et al. "On a simulation study for reliable and secured smart grid communications." In "SPIE Defense+ Security." International Society for Optics and Photonics, 2015.

[13] A. Hahn et al. "Cyberphysical security testbeds: Architecture, application, and evaluation for smart grid." IEEE Transactions on Smart Grid, volume 4, no. 2, pages 847-855, 2013.

[14] H. Zimmermann. "OSI reference model--The ISO model of architecture for open systems interconnection." IEEE Transactions on communications 28.4 (1980): 425-432.

[15] Pecan Street. "Energy and Water Dataport." [Online]. URL: https://dataport.pecanstreet.org/

[16] T. Yardley et al., "Smart grid protocol testing through cyber-physical testbeds," 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, 2013, pp. 1-6.

[17] OMNeT++. "Discrete Event Simulator." [Online]. URL: https://omnetpp.org/

[18] M. Koksal. "A survey of network simulators supporting wireless networks." [Online]. URL: http://www.ceng.metu.edu

[19] AMIsim. [Online]. URL: https://github.com/dolphinhats/Smart-Grid-Advanced-Metering-Infrastructure-privacy-preserving-protocol-implementation

[20] J. Chinnow et al. "A simulation framework for smart meter security evaluation." In "Smart Measurements for Future Grids (SMFG), 2011 IEEE International Conference on," pages 1-9. IEEE, 2011.

[21] G. R. Clarke et al. Practical modern SCADA protocols: DNP3, 60870.5 and related systems. Newnes, 2004.

[22] Power World Corporation. "The visual approach to electric power systems." [Online]. URL: http://www.powerworld.com/

[23] IEEE Power and Energy Society. "A Virtual Smart Grid." [Online]. URL: http://magazine.ieee-pes.org/january-february-2012/a-virtual-smart-grid

[24] GridLAB-D. "Power Distribution System Software." [Online]. URL: http://www.gridlabd.org/

[25] A. F. Molisch, et al. "IEEE 802.15. 4a channel model-final report." IEEE P802 , volume 15, no. 04, page 0662, 2004.

[26] Alliance, ZigBee et al. "Zigbee specification.", 2006.

[27] M. Petrova et al. "Performance study of IEEE 802.15.4 using measurements and simulations." In "IEEE Wireless Communications and Networking Conference, 2006", volume 1, pages 487-492. IEEE, 2006

[28] Latre, et al. "Throughput and delay analysis of unslotted IEEE 802.15. 4." Journal of networks, volume 1, no. 1, pages 20-28, 2006.

[29] Atmel Microchip. "Metering." [Online]. URL: http://www.atmel.com/products/smart-energy/power-metering/

[30] Geek Magazine. "Comparison of compilers for development on microcontrollers with ARM Cortex-M kernel." [Online]. URL: http://geek-mag.com/posts/264558/

[31] T.R. Henderson et al. "Network simulations with the ns-3 simulator." SIGCOMM demonstration 14 (2008).

[32] Network Description (NED) language of OMNeT++. [Online]. URL: http://www.ewh.ieee.org/soc/es/Nov1999/18/ned.htm