Grid Cryptographic Simulation: A Simulator to Evaluate the Scalability of the X.509 Standard in the Smart Grid

Tucker L. Ward
Senior Honors Thesis
Dartmouth College, Hanover, NH, USA

Dartmouth Computer Science Technical Report TR2013-742

ABSTRACT

PKI may be pushed beyond known limits when scaled to some visions of the smart grid; our research developed a simulation, Grid Cryptographic Simulation (GCS), to evaluate these potential issues, identify cryptographic bottlenecks, and evaluate tradeoffs between performance and security. Ultimately, GCS can be used to identify scalability challenges and suggest improvements to make PKI more efficient, effective, and scalable before it is deployed in the envisioned smart grid.

1. INTRODUCTION

Cryptography is critically important in the envisioned smart grid; current cryptographic systems, however, present challenges that may make scaling to the potentially billions of envisioned smart grid nodes difficult or impossible. Conventional wisdom holds that the X.509 PKI standard is the preferred solution for the cryptographic demands of the smart grid [17]. X.509, however, has performed poorly at large scales; previous work has shown that in the domain of BGP routing, PKI added space and time costs that rendered real-time performance impossible in as few as 30,000 nodes [23], while certificate revocation is unworkable with only a few million Web SSL servers [4]. Numerous challenges plagued previous deployments of PKI including certificate path discovery, excessively long revocation lists, limited bandwidth, and the time and space costs of the cryptography. In the envisioned smart grid specifically there are further unresolved questions about the semantics of certificates and identity. For example, who on the grid needs to know that this device is your smart refrigerator, and how do they know that? These challenges may be particularly acute in the limited bandwidth and constantly changing environment of the smart grid.

1.1 The Smart Grid

The smart grid is a modern electrical grid that utilizes an information network to report data about efficiency, reliability, and power consumption from across an electric power system quickly and reliably. The goal of creating the information network is to ensure better power regulation, efficiency, and production while granting the consumer a wider range of choices regarding her own power consumption habits. Ultimately, the smart grid is a system for fine-granularity management of the power grid.

The basic unit of the consumer-side smart grid is the electric meter. A meter reports consumption and time data to a central data collection facility; the facility then aggregates data from many meters to make intelligent service decisions. Visions of the smart grid range from at minimum several hundred million nodes if one counts only households and businesses to several billion if individual appliances are directly connected to the grid

and monitored as some visions of the smart grid prognosticate.

The envisioned information network developed to service this data transfer will require stringent security measures. A cryptosystem is required to ensure that data is encrypted, authenticated, and non-repudiable [6]; the standard system for these requirements is Public Key Infrastructure.

1.2 Public Key Infrastructure

Public Key Infrastructure (PKI) describes a set of standards and protocols for management of digital certificates that allow secure communication in an insecure network via public key cryptography. *Trust roots* issue certificates to properly-credentialed and authorized users; these certificates attest to the validity of the binding between the public key in the certificate and the entity that owns it, provide the verifier with a means to identify the trust root, and maintain other important information about the certificate including its expiration. The user may then present the certificate and some data enciphered by the user's private key; if the data can be deciphered by the public key in the certificate, then the receiving party has reasonable proof that the certificate belongs to the sender, and the receiving party may verify these credentials and the user's public key with the trust root. Users can utilize public/private key pairs in three primary ways: to securely transmit a symmetric key used for data transfer that is shared only for the duration of the session, to provide a mechanism for digital signatures, and to ensure non-repudiation [18].

The simple and generic PKI model described above becomes significantly more complicated as network size and user base increase as do the time and space costs of the cryptographic computation. There may be myriad trust roots issuing certificates; users must therefore decide whether they trust the trust root who issued a presented certificate. *Cross-certification* between trust roots further complicates certificate validation as trust roots certify each other's users. Revocation of certificates poses a greater challenge as trust paths become increasingly complex. Although PKI is straightforward in principle, in practice PKI reveals many hidden costs and challenges as the network and user base grow. We highlight a few examples of PKI challenges and potential smart grid scenarios in Table 1.

1.3 Generic PKI Challenges

Deployments of PKI in industry revealed several unresolved challenges of PKI. Path discovery, the algorithm to find a trusted path between a user's trust root and that of a presented certificate, can be difficult in a network of myriad trust roots. Certificate revocation proved to be an even greater challenge as revocation lists grew to an order of magnitude larger than expected [5] and alternative revocation checking algorithms failed to meet demand. Although a number of measures may be taken to alleviate specific PKI challenges, careful consideration must be given to all challenges to avoid repeating the errors of past PKI deployments.

1.3.1 Path Discovery

Path discovery is the challenge of finding a trusted link between a presented certificate's issuing authority and the set of certificate authorities trusted by the checking node. RFC 3280 [9] defines the standard path discovery algorithm for X.509. The algorithm validates most fields in a given certificate, including the electronic signatures, before proceeding to validate the certificate issuer's certificate information; the cycle continues until the algorithm discovers a common trust root or lack thereof [9]. In a network of more than a

few trust roots this problem becomes nontrivial as trust roots cross-certify, roots go out of business, or the graph of trusted roots simply becomes so large as to require nontrivial time to find a common trust root.

1.3.2 Revocation Lists

Trust roots routinely revoke certificates for numerous reasons including identity compromise, certificate alteration, a request for a new certificate, periodic reissue to reduce the risk of certificate compromise, or even relatively mundane events such as an employee leaving a company. Revoked certificates must be tracked to prevent improper use. Industry has deployed two primary revocation systems: *certificate revocation lists* (CRLs) and the *Online Certificate Status Protocol* (OCSP).

Certificate revocation lists (CRLs), now the most common system, are lists of revoked certificates a user can search to determine if a presented certificate is still valid. The search process itself is not difficult however the lists themselves become untenably long. Johnson and Johnson, which maintains a company-wide PKI for its 165,000 employees, discovered that its CRLs grew to an order of magnitude larger than expected, largely because employees would forget passwords and thus required a new certificate and the old certificate revoked. As CRLs grow, the bandwidth and memory overhead necessary to maintain CRLs across a network becomes significant. A modified CRL system, known as delta CRL, attempted to address this problem by only sending updates to CRLs across the network. Although this update significantly reduced the overhead required, transmissions still exceeded one megabyte weekly in a network of only 160,000 nodes [5]. The overhead requirements on the multi-million node smart grid would be much larger, yet the memory and bandwidth available in the smart grid is significantly smaller [2]; moreover previous work has also shown that using revocation lists alone fails in only one million properly-certified nodes [4] and that revocation introduced significant security holes in client-side SSL. This scalability problem will be significantly worse in visions of the smart grid that may exceed one billion nodes.

OCSP is a server-based validation protocol that obviates the need for CRL transmission by requiring all users to transmit certificates to a centralized server for validation. In theory OCSP's centralized model would alleviate the bandwidth and memory challenges of CRLs, however in practice OCSP introduced a new set of equally challenging problems. Users swamped the validation servers with requests, destroying real-time performance and occasionally crashing servers [15]. Given that reliability is a crucial requirement of the electric grid, OCSP may be unlikely to see widespread deployment in the envisioned smart grid.

1.3.3 Countermeasures

Network topology or other implementation-specific choices can significantly influence the degree to which these challenges affect performance. Imposing a single trust root on the entire network would solve the path discovery problem, for example, but also results in a single point of failure or compromise for the entire network and a workflow bottleneck. Thus note that specific design choices may alleviate some of the potential problems identified in this section; we therefore do not argue that all of the discussed problems will necessarily be manifest in the smart grid, but all of them require careful consideration to ensure that the system deployed to the smart grid does not repeat the errors and omissions of past PKI deployments.

1.4 PKI Challenges in the Smart Grid

In addition to the generic PKI challenges identified we envision several challenges specific to the smart grid. The smart grid may have a fundamentally different granularity of certificate issue than normal PKI networks, there is significant industry pressure to minimize overhead cryptographic costs, and data in the smart grid must be read, aggregated, and acted upon more frequently than in standard PKI systems.

1.4.1 Non-Static Entities

Some visions of the smart grid envisage granularity of control and data collection at the appliance level rather than the aggregated electric meter [17]. Such functionality would appear to require that appliances possess appropriate cryptographic credentials to take part in these communications. While meters are relatively static and are generally owned and operated by an individual or company, appliances are dynamic and may change ownership frequently; moreover, there may be myriad certificate sources depending on where the client buys the appliance and who issues the certificate. Each time an appliance changes ownership a new certificate must be issued and the old certificate revoked; this problem exacerbates the revocation challenges identified earlier. Moreover, there are a number of open questions regarding these "appliance" certificates – who needs to know that it is your refrigerator, how do they know that, and what are the privacy implications of such transparency? Who enforces a change-of-ownership certificate update? Who are the relying parties who must work with those appliances? These unresolved issues require careful thought and exploration before such granularity of control is deployed in the smart grid.

1.4.2 Speed versus Capacity

PKI is expensive in time, memory, and computation; as we mentioned, PKI renders real-time control impossible in as few as 30,000 nodes (in BGP) [23]. In the smart grid many control decisions must be made nearly in real time to avoid blackouts or other severe service interruptions. There is thus a tension between real-time performance and security; PKI itself may destroy real-time performance, but with no security the grid is unprotected. Even in grid contexts where real-time performance is not critical, such as sending meter data, the overhead time, bandwidth, and memory costs of PKI may prove prohibitive. There is thus significant room for research into the overhead costs of PKI and discovering what bottlenecks may be easily relieved to improve performance.

Capacity, defined as the number of clients that can be served by a single server within a given time period, is also a critical component of the envisioned grid. Maintaining servers is expensive and there is therefore significant economic pressure to increase the capacity of servers in the network [2]. This incentive is at cross-purposes with security considerations as security operations often add considerable time and computational overhead. Thus a careful balance must be struck between security and capacity; the relatively large overhead demand of PKI may thus be prohibitive. Many industry contacts have informed us that most utilities choose to maximize capacity at the expense of security [2].

1.4.3 Aggregation

Smart meter or appliance data must be read and potentially aggregated numerous times, from the utility level to the regulatory level to even potentially the Federal Energy Regulatory Commission. In the general PKI

setting, at every point where the data is read or aggregated, the certificate of each sending node must be verified to ensure integrity. The computational overhead required to service so many requests grows quickly as reporting network size increases. Even in the cryptographically simplest case, where data is aggregated at each step and re-encrypted with a new local key pair, in the fully general case the computational overhead in the envisioned billion-node grid is potentially greater than in any other network in the world. We must carefully measure these aggregation challenges before deployment.

PKI Challenge	Scenario
Path Discovery	Smart meter certified by trust root A reports usage data to regulatory authority B which is certified by trust root C.
Revocation	Utility A upgrades 10,000 smart meters requiring new certificates. The old certificates are added to the growing revocation list.
Non-Static Entities	Alice purchases a refrigerator from store A with an initial certificate issued by trust root B. Alice then sells her refrigerator to Bob requiring revocation of the refrigerator's "Alice" certificate from trust root A and issuance of new "Bob" certificate to the refrigerator from trust root B.
Speed versus Capacity	Utility A must choose whether to connect 10,000 or 100,000 smart meters to a single server resulting in an order of magnitude difference in initial and maintenance costs.
Aggregation	Utility A owns 100,000 smart meters. Groups of 10,000 nodes report to local data collection centers; the data is then retransmitted to the utility and finally on to the regulatory authorities. At each step the data is decrypted, parsed, and re-encrypted. If at each step the data is aggregated, then a single compromised aggregator could cause significant harm to the grid. If signatures are simply accumulated, then the messages become significantly longer, as does the task of validating each signature.

Table 1: Scenarios in the Envisioned Smart Grid for each Potential PKI Challenge

1.5 Previous Work

Several researchers have contributed significantly to our understanding of the limitations of PKI.

Zhao, Smith, and Nicol studied the practical constraints of using PKI to create secure BGP. Zhao created a PKI simulation based on the SSFNet framework that simulated nearly 35,000 autonomous systems [21]; the simulated autonomous systems used the X.509 standard for authentication and integrity checks. Zhao found that in as few as 30,000 nodes the practical constraints of PKI significantly increased latency and rendered real-time control impossible [22]. The envisioned smart grid will include potentially several billion nodes. With significant performance degradation and no real-time control in only 30,000 nodes [23], we hypothesize that the performance effects in a multi-billion node network will be much more severe.

Grant studied the relative benefits of numerous proposed changes to the standard X.509 PKI architecture; her conclusion that the recommendations of the Certification

Authority and Browser Forum (CABF) should be strengthened influenced our selection of network protocols for the PKI simulation and will provide an excellent starting point for further research and analysis of alternative grid PKI systems [4].

Smith later condensed these potential smart grid challenges in his extended abstract elucidating potential challenges to the smart grid; our work is built directly on Smith's probing questions [17].

Numerous researchers are also investigating alternatives to PKI in the smart grid. The most compelling active research comes from Los Alamos National Laboratory where a research team demonstrated the viability of quantum cryptography in the smart grid, although the hardware remains prohibitively expensive [50]. Other cryptographic systems, including several symmetric key algorithms, are already used in the grid but there is no clear consensus for specific protocols or usage.

2. MOTIVATION FOR GCS

There is significant potential for error or cryptosystem failure if standard X.509 is simply scaled to a network the size of the smart grid without modification. We therefore created a simulation to model PKI in the envisioned smart grid and identify possible bottlenecks and challenges before PKI is widely deployed. Our goal is to build a simulation that will quantify those bottlenecks, provide a framework to evaluate cryptographic and performance tradeoffs, and explore the viability of alternatives to generic PKI.[1] Ultimately our goal is to ensure that whatever cryptosystem is deployed in the smart grid will have the best chance of success with minimal performance or security compromise.

[1] There is significant precedent for large-scale simulation in the smart grid [1] [3][24]

3. SSFNET

3.1 Previous Work in SSFNet

Discrete-event simulation models a system as a sequence of scheduled events that change the state of the system [16]. The PRIME family of discrete event simulators, comprised of PRIME, SSF, SSFNet, PrimoGENI,[2] and S3F, is a standard for academic research in network simulations [19]. We thus decided to begin our work in SSFNet, the PRIME simulation framework with the most support for large-scale discrete-event simulation with TCP/IP support [11]; moreover, numerous other scholars have used PRIME simulators—in 2005 Zhao created a simulation of PKI in Prime SSF to evaluate the security implications of PKI in BGP [24].

In 2011 Mercado unsuccessfully ported Zhao's PKI components to PRIME SSFNet; despite his best efforts the PRIME SSFNet framework was not compatible with Zhao's code [13]. Our work in turn builds on the efforts of both Zhao and Mercado; our first attempt to build a simulation of PKI in the envisioned smart grid was to work with Mercado's code and again attempt to port it into SSFNet. We initially decided to attempt to continue Mercado's work porting Zhao's code to PRIME SSFNet because PRIME's real-time simulation strategy allows it to interact with real network applications [12], a potentially powerful tool that would enable us to use deployed industry network code in our simulations. Despite some initial success building on Mercado's work, our attempts to transfer the code to SSFNet also proved unsuccessful for the same compatibility

[2] PrimoGENI is a version of SSFNet adapted to work with the Global Environment for Network Innovations (GENI). PrimoGENI was newly released when we started our work and now that it has been thoroughly tested would likely be the best simulator for further research [20]

issues Mercado encountered [13]. We thus decided to build our own PKI simulation module in SSFNet[3] from scratch.

3.2 SSFNet Simulation

Creating a PKI module within the PRIME SSFNet framework proved more challenging than anticipated. The base code failed to compile on any of our lab machines; errors ranged from #include failures to unknown data types. After nearly six weeks of experimentation, code checks, and support from Dr. Miguel Erazo at FIU, we discovered that the PRIME SSFNet framework could only be compiled in the deprecated g++ 4.1.2 in Ubuntu 10.04.[4]

After successful compilation we began to write a PKI module for SSFNet. Modules in SSFNet are not trivially defined; each new module must implement its own network stack and application layer using prototypes defined within SSFNet's base simulation code. We therefore first attempted to build an application-layer module on a copy of the existing TCP/IP network stack; however the existing TCP/IP stack failed to communicate with our application. After a series of discussions with Dr. Erazo, we were unable to identify the source of the error although we believed that the existing TCP stack code relied on a number of hard-wired application-layer function calls that we either could not locate or were running in the background simulation support code and would be dangerous to alter. Dr. Erazo advised us to write our own network stack to avoid these issues. David Rice proceeded to begin work on the TCP/IP stack while Tucker Ward continued to develop the application-layer PKI module.

Writing a new module proved equally problematic as compiling. For nearly 15 weeks we continued to write and test module code; despite continued support from FIU and long debugging hours our module code simply did not communicate with the SSFNet simulation framework. We eventually traced this error to a function type conflict that could not be resolved. At run time the SSFNet base code expects the user to define a function that calls the network functions that move packets through the TCP/IP stack. However, in order to compile, the base code required the user to implement this same function with a completely different set of parameters. The result was that we were forced to make code that either did not compile or would call the wrong function at runtime. We confirmed this issue with Dr. Erazo and together were unable to find a solution.

3.3 NS3 Simulation

In December 2012, we finally decided to abandon PRIME altogether. In consultation with Professor David Nichol at the University of Illinois Urbana-Champaign, we elected to continue simulation development in the NS3 simulation environment. Work on the new NS3 simulation began in January 2013; Tucker Ward completed all further work on this project.

4. GCS SPECIFICATIONS

4.1 Overview

GCS is built on the ns-3.16 framework publicly available at http://www.nsnam.org/. New users wishing to run the simulation code should download the simulation tarball and build NS3 using the instructions on the

[3] We also briefly considered changing to the second-generation SSF simulation, S3F [14], however the S3F project is still relatively new and untested

[4] It may be possible to compile SSFNet in other operating systems however the g++ 4.1.2 is a firm requirement.

NS3 website.[5] Please contact the author of this report for the additional GCS code. Waf will automatically build the GCS simulation assuming NS3 is correctly installed.

We determined simulation protocols and parameters from best estimates from conversations with industry contacts and from open-domain sources. We designed the code to allow for easy parameter or protocol changes in the event that the user generates new best estimates.

4.1 Modules

GCS adapts two primary modules from NS3. The TCP/IP network implementation is used unaltered from the base NS3 code. The application-layer PKI code, however, is an extensively edited version of the OnOff Application from the base NS3 code. All PKI code executes in the application layer.

4.2 Software Architecture

GCS builds extensively on the NS3 base code. There is no native support for PKI simulation in NS3 excepting support for a standard TCP/IP network stack. GCS is thus built on heavily modified applications from the NS3 base. The simulation is generally speaking built on a standard client/server model in which the client reports grid data to a server; the server then either aggregates data and sends it to higher-order server or collects the data for analysis. For modularity and ease of modification the GCS code is contained entirely within two application files and one topology file. We altered no other NS3 base code.

Client

The GCS client is an extensively modified version of NS3's native OnOff application. OnOff offers the nearest proxy to a smart meter as its purpose is to report data to a central server at regularly-scheduled intervals. The code, however, was almost entirely replaced to support PKI and now only the transmission scheduling function and the destroyer are original to NS3; less than 10% of the existing code is taken from the OnOff base.

Server

The GCS server is an extensively modified version of NS3's native PacketSink application. PacketSink is a server model that accepts data from any other application in NS3. PacketSink is less extensively altered than OnOff, but is still less than 30% original to NS3.

Topology

The topology file, pki_grid.cc, is entirely our work although it was in part inspired by examples found in the NS3 codebase. The default topology is a star network with default size and parameters described below.

4.3 Protocols

4.3.1 Overview

GCS simulates the X.509 PKI protocol. Each node maintains a simplified certificate struct that contains an expiration time, the certificate issuer, the node's ID number, and some bookkeeping information—all of the information necessary for a simulation to emulate real PKI but simplified to conserve memory. GCS conducts most functions of PKI except cryptography; GCS verifies, issues, revokes, and transmits certificates and data as in a real X.509 deployment. We simulate encryption and decryption simply by adding constant time to the simulation clock and transmission by adding a constant to the bandwidth usage in proportion to the number of certificates in the certificate chain of a given transmission. Once two nodes

[5] http://www.nsnam.org/docs/release/3.14/tutorial/singlehtml/index.html

validate each other's certificates a session key is randomly generated and data transmitted between the two nodes, although the actual encryption/decryption again is simulated simply by adding constant clock time. Digital signatures and non-repudiation thus may be simulated by simply ignoring the key pair and validating the certificates.

4.3.2 Certificate Issue

Simulated trust roots issue new certificates to nodes when a node's certificate is revoked either through expiration or failure to authenticate. The trust root gives the certificate an expiration time, records is trust root ID, and assigns the certificate a sequential number for individual certificate identification. The trust root sends the new certificate to the respective node and the node adopts it for use. Identity checks are simulated by adding constant time to the simulation clock before the node may use the new certificate; although this practice inverts the real-world order of operations, the net effect is the same and it simplifies certificate issue.

4.3.3 Certificate Revocation and Expiration

Certificate reissue occurs because either the original certificate expired or GCS randomly selects it for revocation. The trust root handles certificate renewal by simply issuing a new certificate to the node. The trust root furthermore adds the old certificate to the certificate revocation list (CRL) until it expires. Expired certificates are not kept on the CRL as nodes are expected to verify the expiration date of all certificates presented for authentication. Random revocation occurs by default with a frequency determined by a simulation constant. Nonrandom revocation is also supported but must be specified in the topology file. Revocation simulates the range of real-world scenarios in which a certificate may be revoked in the smart grid: a user forgets a password, a user breaks an appliance, a hacker compromises a utility, or a utility goes out of business, among others. GCS provides extensive capabilities to specify revocation policies and events.

4.4 Simulation Parameters

Based on conversations with industry contacts, we used the following default parameters for consumer-side simulation in a small geographic network:

Data Payload: 2 kB
Report Frequency: 15 minutes to 6 hours
Transport/Network Protocol: TCP/IP
Certificate Revocation Rate:
0.01% - 0.10% per day
Bidirectional Communication: 95% uplink, 5% downlink communication
Network Topology: Star or Radio Mesh

Best-estimate parameters:

Neighborhood Size:[6] 10,000 nodes
Utility Size:[7] 1 – 100 neighborhoods
Total Network Nodes: 10 million
Relying Parties: Utilities and a "Regulator"

4.5 Statistics Gathering

GCS gathers the following statistics:

- *Average bandwidth used per link over time*. Average bandwidth indicates how much bandwidth will be required for a given link in the network.

[6] "Neighborhood" simply denotes a set of data collection nodes that roughly corresponds to a base unit – either residential, industrial, or commercial. The size may be trivially adjusted to reflect any particular network size desired.
[7] "Utility" denotes the size of a nontrivial power production unit, such as NSTAR in New England

- *Average latency per link*. Latency data will assist in emergency and control planning where response time must be in milliseconds.
- *Peak bandwidth per link*. Peak bandwidth will help engineers plan for worst-case failure scenarios in which a data overload saturates a network.
- *Peak latency per link*. Peak latency gives the researcher a sense of the upper tolerances of the network under extreme conditions.
- *Certification revocation list size*. CRL size is a critical metric for determining how much computational and bandwidth resource is required in a given network simply to support the PKI operations.
- *PKI computational overhead time cost*. Overhead cost helps engineers determine minimum hardware and software benchmarks for successful deployment in the smart grid.
- *Certificate validation time and memory cost*. Time and memory cost indicate how feasible PKI may be on smaller machines with limited bandwidth access.
- *Packet traces*. Packet traces assist in simulation debugging and any fine granularity analysis.

GCS can be easily extended to gather or analyze additional statistics as desired.

4.6 Current Simulation Status

The GCS code modules are approximately 1500 lines of C++ code built on the NS3 framework; simulation topologies vary from only dozens of lines of code to several hundred depending on user specifications. The basic star network topology is already ready for demonstration and thorough testing while the radio mesh topology is under development. Unusual network topologies or deviations from standard PKI protocols are trivially possible by modifying the topology or protocol files. We designed GCS to abstract away many of the underlying NS3 code allowing a relatively new programmer to edit and run the simulation without knowing the details of NS3. GCS is ready for data collection but we will continue to expand its repertoire of simulation models and topologies.

4.7 Simulation Testing and Topology

GCS has been thoroughly tested on a number of small and large grid topologies. We verified the accuracy of statistics gathering by running two small-scale simulations of 12 nodes in three stars network topology with predictable network traffic that was verified manually. We then scaled testing to a much larger 10,000 nodes in 100 stars topology and ran the simulation for three simulated days. The larger test simulates a metropolitan topology of at most two electric regulators collecting data solely from smart meters reporting to designated data collection servers as might be found in most medium-sized cities in the United States. More or less complex topologies, including data collection from or control of appliances, additional electric regulators, or stress conditions such as a high number of trust roots may be simulated by carefully changing simulation constants, although network traffic protocol or routing changes would require minor code alterations.

The flexibility and testing of GCS thus provides for a wide variety of further simulations and network topologies. The benefits and costs of collecting data from or even controlling individual appliances may be assessed simply by adding an additional

layer of endpoints in the simulation and requiring those nodes to report to the connected smart meter node. Larger networks, such as those for large cities, states, or even an entire country may be simulated fairly trivially by increasing the size of the simulation, the number of trust roots, the number of electric regulators, and modifying simulation traffic to more closely match a national average than any regional variations. GCS may also prove useful on smaller scales if, for example, a researcher desired to test the viability of multiple trust roots within a relatively small and localized network—as may be the case in some sort of regulatory headquarters to which many utilities report data. Ultimately, with minimal modification GCS may be used to simulate nearly any network that may be represented as a star or mesh network of nodes using X.509 PKI.

5. FUTURE WORK

The star topology simulation is ready for deployment and large-scale data collection, although periodic refinements of the simulation parameters and topology are expected. Future researchers may use this simulation to collect data and discover the practical performance of PKI in a smart grid network. Once bottlenecks are identified, alternative PKI schemes, network topologies, or other heuristics may be explored to identify simple solutions to the complex problems PKI may pose in the envisioned smart grid.

6. CONCLUSION

The envisioned smart grid will require a robust cryptosystem to meet the demands of confidentiality, integrity, and availability. Conventional wisdom holds that Public Key Infrastructure is the best solution. However, deployed PKIs demonstrated a number of challenges including path discovery and revocation issues while we envision several smart grid-specific challenges such as non-static entities, speed versus capacity tradeoffs, and aggregation. In addition to those challenges, the potential billion-node network of the smart grid will be larger than any PKI previously deployed; in short, in the smart grid, PKI may be pushed beyond known limits and must go where no PKI has gone before. We therefore constructed a simulation of PKI in the smart grid, GCS, that based on the NS3 simulation environment and can identify which of those challenges may pose serious threats to grid integrity and performance.

GCS can simulate a wide variety of grids, topologies, protocols, and even non-grid networks. GCS can be modified to simulate nearly any level of grid communications infrastructure. The GCS default is to simulate a network of smart meters reporting to centralized utility data servers; with minor modifications to the protocols and frequency of reporting, GCS can simulate, for example, the command and control functions of generation, transmission, and power distribution, a utility-wide data/regulatory communications network, a federal regulatory authority's communication and control network, or even a combined data, control, and regulatory network of any reasonable size. GCS can also simulate at the finer granularity of smart appliances or even smart appliance command and control. Each of these network types may be arbitrarily large, constrained only by the computational resources available. In addition to network type and topology, GCS may be easily modified to simulate protocols other than the X.509 PKI standard; other variations of PKI or even a different authentication system, such as symmetric key, may be examined by

re-tooling the protocol code within GCS. GCS can even simulate non-grid networks; although nodes in GCS are considered to be smart meters, with minor modifications their behavior may be adjusted to match personal computers or any other computational device to simulate a wide range of real-world networks. GCS is a robust and malleable tool for simulating a wide range of networks.

GCS simulations can clarify many potential problems in large-scale cryptosystems and evaluate potential tradeoffs. Potential tradeoffs include the effect of imposing a single trust root on the grid versus security, the effect of only checking for certificate revocation every *n* times a node is presented with the certificate, the effect of increasing or decreasing the expiration time of a certificate, the effect of enforcing separate PKIs for data and control systems, the effect of data aggregation at each regulatory level or forwarding the data directly, or even the effect of enforcing command-only PKI and not encrypting usage data. Simulations will also quantify the range of potential issues discussed in this paper, from bandwidth utilization to CRL growth to latency. Using the data collected when evaluating the above-mentioned potential tradeoffs can help industry and government make better decisions based on clear, quantifiable criteria for the envisioned smart grid.

We hope to use GCS to identify and alleviate PKI issues before PKI is widely deployed in the smart grid. GCS is a robust base that can be easily modified to simulate nearly any PKI deployment in the envisioned smart grid, and our ultimate goal is to identify and quantify potential challenges and bottlenecks before any cryptosystem is deployed to the envisioned smart grid.

ACKNOWLEDGEMENTS

This material is based in part upon work supported by the Department of Energy (under Award Number DE-OE0000097) and by the NSF (under grant CNS-0448499). The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The author would also like to thank Sean Smith at Dartmouth College, David Nichol at University of Illinois Urbana-Champaign, Jason Liu a Florida International University, Robert Lee of General Electric, the Los Angeles Department of Power and Water, and Tim Yardley of UIUC for their advice and guidance on this project.

BIBLIOGRAPHY

[1] Anderson, R. J. Security Engineering: A Guide to Building Dependable Distributed Systems, 1st ed. John Wiley & Sons, Inc., New York, NY, USA, 2001, Ch. 20, p. 428.

[2] Bharat, Bruce, Senior Engineer for Smart Grid engineering at General Electric. Interview April 25, 2013.

[3] Erazo, M. A., and Liu, J. On Enabling Real-Time Large-Scale Network Simulation in GENI: The PrimoGENI Approach. In Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 39:-39:2.

[4] Grant, A. Search for Trust: An Analysis and Comparison of CA System Alternatives and Enhancements. Tech. Rep. TR2012-716, Dartmouth College, Computer Science, Hanover, NH, January 2012.

[5] Guida, R., Stahl, R., Bunt, T., Secrest, G. Deploying and Using Public Key Technology: Lessons Learned in Real Life. IEEE Security & Privacy, 2, 4 (July-Aug. 2004), 67-71.

[6] Hauser, C., Bakken, D., and Bose, A. A Failure to Communicate: Next Generation Communication Requirements, Technologies, and Architecture for the Electric Power Grid. IEEE Power and Energy Magazine, 3, 2 (March-April 2005), 47-55.

[7] Hauser, C. H., Bakken, D. E., Dionysiou, I., Gjermundrød, K. H., Irava, V. S., and Bose, A. Security, Trust and QOS in Next-Generation Control And Communication For Large Power Systems. International Journal of Critical Infrastructures (2007).

[8] Housley, R., and Polk, T. Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure, 1st ed. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[9] IETF. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF: RFC 3280. http://www.ietf.org/rfc/rfc3280.txt. Accessed May 24, 2013.

[10] LANL. Quantum Cryptography Put to Work for Electric Grid Security. http://www.lanl.gov/newsroom/news-releases/2013/February/02.13-quantum-cryptography.php

[11] Liu, J. A Primer for Real-Time Simulation of Large-Scale Networks. In Proceedings of the 41st Annual Simulation Symposium (April 2008), pp. 85-94.

[12] Liu, J., Li, Y., and He, Y. A Large-Scale Real-Time Network Simulation Study Using PRIME. In Proceedings of the 2009 Winter Simulation Conference (WSC), (Dec. 2009), pp. 797-806.

[13] Mercado, A. Adapting SSFNet for PKI Scalability in the Smart Grid. Senior Thesis, Dartmouth College, June 2012.

[14] Nicol, D., Jin, D., and Zheng, Y. S3F: The Scalable Simulation Framework Revisited. In Proceedings of the 2011 Winter Simulation Conference (WSC), (Dec. 2011), pp. 3283-3294.

[15] Poynter, I. To Extend OCSP or Not? Network World 18, 9 (Feb. 2001), 74.

[16] Robinson, S. Conceptual Modeling for Discrete-Event Simulation. Boca Raton: CRC Press, 2011, pp 1-5.

[17] Smith, S. Cryptographic Scalability Challenges In The Smart Grid (Extended Abstract). In Innovative Smart Grid Technologies (ISGT), IEEE PES (Jan. 2012).

[18] Smith, S., and Marchesini, J. The Craft of System Security. Safari Tech Books Online. Addison-Wesley, 2008.

[19] SSFNet. Why We're Here. http://www.ssfnet.org/homePage.html, 2002. Accessed 29 April 2013.

[20] Van Vorst, N., Erazo, M., and Liu, J. PrimoGENI: Integrating Realtime Network Simulation And Emulation in GENI. In IEEE Workshop on Principles of Advanced and Distributed Simulation (PADS), (June 2011), pp. 1-9.

[21] Zhao, M. Performance Evaluation of Distributed Security Protocols Using Discrete Event Simulation. Tech. Rep. TR2005-559, Dartmouth College, Computer Science, Hanover, NH, January 2005.

[22] Zhao, M., Smith, S., and Nicol, D. The Performance Impact of BGP Security. IEEE Network, 19, 6 (Nov.-Dec. 2005), 42-48.

[23] Zhao, M., Smith, S. W., and Nicol, D. M. Aggregated Path Authentication for Efficient BGP Security. In Proceedings of the 12th ACM Conference on Computer and Communications Security (New York, NY, USA, 2005), CCS '05, ACM, pp. 128-138.

[24] M. Zhao, S.W. Smith. Modeling and Evaluation of Certification Path Discovery in the Emerging Global PKI. Public Key Infrastructure: EuroPKI 2006.