Cryptographic Scalability Challenges in the Smart Grid *(Extended Abstract)*

Sean W. Smith

***Abstract*—In the envisioned smart grid, massive numbers of computational devices will need to authenticate to each other. For scalability, this technology will probably rest on *public-key infrastructure (PKI)*. However, deploying PKI on an entity population this large—and doing the kinds of things we envision the smart grid doing—itself raises many scalability challenges the community will need to address. We survey some.**

I. Introduction

Most visions of the "smart grid" prognosticate vast numbers of computational devices embedded in consumer and transmission elements of the power grid and exchanging data; the visions differ in the details of exactly what the devices are, where they are, what data they exchange with whom, and what gets done with it.

Nevertheless, these visions all posit lots of devices—some predictions suggest the smart grid may have *more* new devices, somehow interconnected, than the Internet itself currently has.

Making this all function securely will require a lot of work. In this paper, we consider one component of the problem: how the devices will authenticate themselves and their transmissions to the other devices (and to the other various stakeholders in the system, such as the various power enterprises, customers, coordinating entities, etc).

In the basic view, computational devices authenticate themselves and their transmissions using a cryptographic key. ([1] surveys this area for the smart grid domain.) With conventional symmetric cryptography, both the sender and receiver must share the same key. The straightforward solution here would have each device sharing a key with each other device it might need to communicate with; for n devices, that means each device must know $\Omega(n^2)$ keys—not scalable for a device population larger than the Internet.

Oops. That should be n. (n^2 for the WHOLE system)

Consequently, the standard conclusion reached is (e.g., [2], [3]) is that the smart grid should use *public-key cryptography*: To operate on a transmission, the sender and receiver can use different keys, one not derivable from the other (at least in one direction). Initially, this brings the number of keys a device needs to know down to n: its own key pair, and the public key of each other device. However, the ability of public-key cryptography to enable digital signatures—a party can use its private key to sign an assertion verifiable by anyone knowing its public key—brings the number down to 2: a device only needs to know its own keypair, and the public key of the *trust root* it trusts to sign assertions saying what the public keys are of the other devices with which it needs to work. In either case, the number of *secrets* a device needs to know is exactly one: its own private key; this constraint limits the damage that compromise of a device can cause.

S.W. Smith is with the Department of Computer Science, Dartmouth College, Hanover NH 03755 USA e-mail: sws@cs.dartmouth.edu

Public-key infrastructure (PKI) is the catch-all term for mechanics necessary to establish, maintain, and distribute these assertions (*certificates*) and key pairs. Typically, PKI includes components to solve these problems:

- How does a keyholder obtain a certificate from a trust root?
- How does a relying party decide who its trust roots are?
- How does a relying party get its hands on a *path* from a trust root to a particular certificate?
- What exactly should a relying party conclude from discovering a path with each link apparently signed properly?
- What do we when the assertion a certificate makes (e.g., "X has public key E_x") is no longer true, and needs to be *revoked*?

X.509 denotes the family of standards and techniques (e.g., [4]) that has come to dominate the way most PKI is done in practice, although other rivals (e.g., [5], [6]) surface now and then, as do regular critiques (e.g., [7]).

Because of this small number of keys and secrets in any one device, PKI appears the natural scalable solution to device and data authentication in the coming smart grid. However, even though these particular numbers seem small and scalable, the rest of the PK infrastructure can hide costs that are not nearly as scalable—particularly when we consider the uses to which these devices may be put. We now consider some of the problem areas. (See [8], [9] for additional discussion.)

II. Generic PKI Issues

We start with some of issues relevant to PKI for *any* large-scale population.

a) Trust Roots: In the textbook view of PKI, a single trusted entity acts as the universal *certification authority*. This one party issues all certificates; this one party's public key is the only trust root any relying party ever needs to know. Unfortunately, even in the PKIs that have emerged so far (relatively small, compared to the smart grid), this simplifying property has failed to hold. For reasons logistical, economic, and otherwise, multiple CAs emerge serving various parts of the population. Consider: as of this writing, the population of SSL-protected Web servers are served by over 100 different trust roots—and the smart grid will have far more devices in far more homes and businesses than the world currently has SSL-protected Web servers. So, for the smart grid, we will have to either figure out how to solve a problem no one's solved yet—having one entity sign certificates for a vast

population—or we'll have to deal with the reality of myriad trust roots.

Having myriad trust roots raises the questions of how these roots should be organized. One might imagine a strict hierarchical tree, where higher-level roots certify lower-level ones, and the lowest-level certify devices. One might also imagine a system where peer roots have their own user populations, but *cross-certify* each other. ("My users can trust R_2 to talk about devices in subpopulation P_2.") One might imagine having *bridge* authorities who exist only to cross-certify. One might imagine just a loosely-organized *oligarchy* of independent roots. All of these approaches have emerged in current PKIs, with varying degrees of implementation and engineering complexity.

b) Trust Paths: An artifact of moving from a single universal root to a more complex network complicates the notion of *trust path*: the route from a relying party's root to a target certificate. In the simple model, the trust path *is* the certificate: the One True Root signed this, so we believe it. In a more complex model, we need to figure out how to construct the trust path, and what the semantics mean (for example, consider the composition of two cross-certificates above). When a relying party P needs to make a judgment about certificate C whose trust path may be long, we also need to figure out how to get all the other certificates P might need to P. Suddenly we might need directories and repositories, and the space in protocols and handshakes and tables we implicitly assumed would hold one certificate may now hold several. (Indeed, current PKI-based tools can break when an extra certificate gets introduced into paths.)

How we make this scale to a population the size of the smart grid is not trivial.

c) Revocation: Secrets become non-secret. In the current world, people lose credit cards and college/employer ID cards; people divulge passwords; activist hackers (and presumably more secretive malicious ones) penetrate systems to obtain keys (even high-value private keys). Even in these cases, human individuals perceive a motivation to keep their credit cards or login dongles close at hand.

In the envisioned smart grid, we will have far more devices distributed in more uncertain environments. (Who exactly has access to that box? Will they have motivation to protect it—or to compromise it?) Furthermore, the state of the art in having physical devices protect their own secrets is a continual game of cat-and-mouse between attack and defense technology (e.g., [10], [11]); if grid devices are to be affordable and long-lived, it's probably safer to assume that adversaries will be able to extract secrets if they get destructive physical access.

Consequently, a PKI needs to allow for the fact that any given certificate may need to be suddenly *revoked*: "oops, it's no longer true that the thing which knows the private key matching the public E_x is necessarily X." The necessity of potential revocation gives rise to a new problem: how will does a relying party know if a given certificate has been revoked? (If non-trivial trust root structure has given us non-trivial trust paths, the problem is compounded: the relying party needs to do this for each certificate in a potential trust path.)

The traditional PKI approach to this problem has been for a CA to regularly publish a *certificate revocation list (CRL)*. In theory, a relying party regularly obtains a fresh CRL, assumes it's valid until the next CRL, and during the meantime checks if each new target certificate is present in the list. In practice, this hasn't worked too well: in domains ranging from enterprise (e.g., [12]) to defense (e.g., [13]) to industrial infrastructure (e.g., [14], [15]) to Web SSL, CRLs have consistently proven significantly much larger than expected, straining bandwidth and exacerbating latencies.

Researchers have explored alternatives, such as *online certificate status protocol (OSCP*—where the relying party checks a certificate's revocation status with a trusted entity in real time—or hash-chained schemes (e.g., [16]), to reduce the bandwidth for revocation data. Nonetheless, even in current large-scale PKIs, revocation is a challenge. It doesn't scale. How is a PKI for a smart-grid-sized population going to fare better?

III. Smart Grid Devices

Traditional PKI focuses on binding a public key to the keyholder's identity, which is implicitly assumed to be a well-defined, relatively static thing (such as individual's full name or email address, or the hostname of a public webserver). However, in the envisioned smart grid, the relevant properties of a the keyholder are not just the device's identity ("this is a meter made by ACME"; "this is a refrigerator made by GE") but its *context*: "this is a refrigerator in the apartment rented by Alice, who buys power from X." This context information will not necessarily be known until device installation. This information may change dynamically. (What if Alice sells her fridge on Craigslist or sublets her apartment to Bob? What if repair personnel Cathy replaces Alice's meter?), This information may also may not be particularly simple. (What if Alice's landlord owns many apartment buildings, and changes power vendors to get a better rate?)

If our cryptographic infrastructure is going to enable relying parties to make the right judgments about these smart grid devices, this additional information needs to be somehow available. We can try to modify a traditional identity-based PKI to attest to these more dynamic kinds of identities; we could try instead to adapt the largely experimental world of *attribute certificates* (e.g., [17]) to supplement the identity certificates in the smart-grid PKI. Either of these approaches breaks new ground. Alternatively, we can leave the identity PKI in place and use some other method of maintaining and distributing this additional data; this requires supplementing our scalable PKI with a non-scalable database.

In any of these approaches, we also need to think about who is authorized to make these dynamic updates. Who witnesses that Alice has sold her refrigerator? Thinking about this organizational structure of smart grid devices also complicates the revocation problem. If we can't quite figure out who it is that speaks for where a device currently lives, how will we figure out who it is who is authorized to say it has been compromised?

IV. Smart Grid Applications

Another family of cryptographic scalability issues emerges when we consider the kinds of things the community envisions that smart grid devices will be doing with their data.

To start with, consider the legacy grid. Devices exchange data, sometimes on overtaxed networks, and need to make decisions on this data in near-real-time in order for the grid to maintain overall stability. If we suddenly add PKI-based authentication to these communications, we suddenly increase sizes: of data transmitted (we need to allow room for signatures and for certificates and for certificate paths and for revocation lists) and of processing time (to verify signatures, to discover paths, to verify paths, to check revocation). In exploratory work my lab in did in the relatively simpler world of Internet routing [18], we discovered the cumulative effect of fairly basic PKI on near-real-time infrastructure stability was not trivial. What will it be like when we scale up to the smart grid?

Another set of issues arises if we think of the *aggregation* that the community expects smart grid devices to perform. Various visions place the aggregation in various places, but they usually posit some set of devices collecting and aggregating information from a large set. What will the cryptographic burden these aggregators incur when they verify all their inputs? With its own output, will an aggregator need to forward on all the inputs and signatures it receives, in order for its own relying parties to verify provenance? If so, how will that scale? (Will we need to consider elliptic-curve cryptography tricks such as aggregate signatures (e.g., [19])?) If not, how will relying parties verify provenance?

Yet another set of issues arises if we think about *privacy.* The power community may see the bright side of how the smart grid will bring increased resiliency and efficiency; however, many in the larger community worry about how this new infrastructure will enable more prying eyes to see more private details of a household's or enterprises's activities. On the application level, aggregation and other blinding techniques may help address these concerns. However, even with these techniques, the underlying PKI level may compromise privacy. For example, the trust path supporting the certificate meter X may betray the path of ownership, landlord, and subletting; the revocation or directory queries a substation server S receives from neighborhood consumer devices can betray their activity, even if the application blinds details. The research community has explored some privacy-enhancing solutions here—but will they scale to the smart grid?

V. Conclusion

The cryptographic infrastructure underlying the smart grid the community envisions will likely require PKI, for scalability—but this is the beginning, not the end, of the solution.

Acknowledgment

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 Eric Hacker for his helpful discussion on this topic, over lunch at the 2011 at the TCIP-G Summer School.

References

[1] H. Khurana, R. Bobba, T. Yardley, P. Agarwal, , and E. Heine, "Design Principles for Power Grid Cyber-Infrastructure Authentication Protocols," in *Hawaii International Conference on System Sciences*, 2010.

[2] J. Benoit, "An Introduction to Cryptography as Applied to the Smart Grid," Cooper Power Systems, February 2011.

[3] A. R. Metke and R. L. Ekl, "Security Technology for Smart Grid Networks," *IEEE Transactionson Smart Grid*, vol. 1, 2010.

[4] R. Housley and T. Polk, *Planning for PKI*. John Wiley and Sons, 2001.

[5] D. Clark, J. Elien, C. Ellison, M. Fredette, A. Morcos, and R. Rivest, "Certificate Chain Discovery in SPKI/SDSI," *Journal of Computer Security*, vol. 9, no. 4, pp. 285–322, 2001.

[6] R. Rivest and B. Lampson, "SDSI - A Simple Distributed Security Infrastructure," April 1996, http://theory.lcs.mit.edu/~rivest/sdsi10.html.

[7] P. Gutmann, "PKI: It's Not Dead, Just Resting," *IEEE Computer*, vol. 35, no. 8, pp. 41–49, 2002.

[8] "NISTIR 7628: Guidelines for Smart Grid Cyber Security," The Smart Grid Interoperability Panel Cybersecurity Working Group, 2010.

[9] J. Tiller, "Smart Grid PKI: The Hidden Security Challenge," http://www.realsecurity.us/weblog/?e=124, August 2010.

[10] S. Smith, "Fairy Dust, Secrets, and the Real World," *IEEE Security and Privacy*, vol. 1, no. 1, pp. 89–93, 2003.

[11] S. Weingart, "Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses," in *Cryptographic Hardware and Embedded Systems—CHES 2000*. Springer-Verlag LNCS 1965, 2000, pp. 302–317.

[12] R. Guida, R. Stahl, T. Bunt, G. Secrest, and J. Moorcones, "Deploying and Using Public Key Technology: Lessons Learned in Real Life," *IEEE Security and Privacy*, vol. 2, 2004.

[13] R. Nielsen, "Observations from the Deployment of a Large Scale PKI," in *4th Annual PKI R&D Workshop*. NIST, 2005.

[14] IBM, "Significant delay when starting .NET client application," https://www-304.ibm.com/support/docview.wss?uid=swg21394928, 2009.

[15] P. Kehrer, "Defective By Design? - Certificate Revocation Behavior In Modern Browsers," http://blog.spiderlabs.com/2011/04/certificate-revocation-behavior-in-modern-browsers.html, 2011.

[16] S. Micali, "NOVOMODO: Scalable Certificate Validation And Simplified PKI Management," in *1st Annual PKI Research Workshop—Proceedings*. NIST Special Publication 800-62, 2003.

[17] D. Chadwick, A. Otenko, and E. Ball, "Role-Based Access Control with X.509 Attribute Certificates," *IEEE Internet Computing*, March-April 2003.

[18] M. Zhao, S. Smith, and D. Nicol, "Evaluating the Performance Impact of PKI on BGP Security," in *4th Annual PKI Research and Development Workshop*. NIST/NIH/Internet2, April 2005.

[19] D., C. Gentry, B. Lynn, and H. Shacham, "A Survey of Two Signature Aggregation Techniques," *RSA CryptoBytes*, vol. 6, no. 2, 2003.

Sean W. Smith Prof. Sean Smith has been working in information security—attacks and defenses, for industry and government—since before there was a Web. As a post-doc and staff member at Los Alamos National Laboratory, he performed security reviews, designs, analyses, and briefings for a wide variety of public-sector clients; at IBM T.J. Watson Research Center, he designed the security architecture for (and helped code and test) the IBM 4758 secure coprocessor. In July 2000, Sean left IBM for Dartmouth, since he was convinced that the academic education and research environment is a better venue for changing the world. His current work investigates how to build trustworthy systems in the real world.