Design Principles for Power Grid Cyber-Infrastructure Authentication Protocols

Himanshu Khurana, Rakesh Bobba, Tim Yardley, Pooja Agarwal and Erich Heine
Information Trust Institute, University of Illinois at Urbana-Champaign
{hkhurana, rbobba, yardley, pagarwl, eheine}@illinois.edu

Abstract

Recently, there has been an increased focus and a sense of urgency in developing standards for Power Grid systems centered on the need for interoperability. Given the threat against these systems an important goal is the development of effective cyber security standards. However, past experience shows that security protocols are prone to design errors. Focusing on authentication protocols, in this work we discuss key design principles and engineering practices that we believe can help ensure the correctness and effectiveness of standards for authentication in Power Grid protocols; e.g., DNP3. This work builds on past work in the area of principles of authentication in Internet protocols but focuses the discussion on the constraints of the Power Grid; in particular, the need for efficient and highly available systems[1] .

1. Introduction

Effective integration of renewable energy sources, energy conservation and better usage of the high voltage transmission system are major driving forces for an enhanced electric grid often called the "Smart Grid." The vision for a modernized Smart Grid involves the use of an advanced computing, communication and control cyber infrastructure for enhancing current grid operations by enabling timely interactions among a range of entities. The coupling between the power grid and its cyber infrastructure is inherent, and the extent to which the Smart Grid vision can be achieved depends upon the functionality and robustness of the cyber infrastructure. To that end there has been increased focus and a sense of urgency around the development of standards that promote interoperability between various Smart Grid systems. Given the prominent threat against the electric grid cyber infrastructure an important part of this effort is devoted to development of cyber security protocols and their standardization.

Cyber attacks launched against the power grid may aim to disrupt operations modifying or inserting messages. For example, malicious entities might change the set points at outstation devices, by pretending to be a master device at a control center, and cause instability in the grid. Adequate protection involves authenticating both the devices and control commands. In this work we take a closer look at protocols and standards for *authentication* built on cryptographic tools and discuss key design principles that can help ensure the correctness and effectiveness of the developed standards.

Authentication is an important recognized problem for the grid. In fact, researchers and industry practitioners have recently focused on developing authentication protocols and standards for the electric grid, for example, the DNP Users Group is currently developing a standards for DNP3 Authentication [13] and the IEC is developing the 62351-5 standards [19] for securing Transmission Protocols in Telecontrol Equipment and Systems. These two standards are also some of the primary cyber security standards identified by the National Institute of Standards and Technologies (NIST) as pertinent to their ongoing interoperability effort.

Intuitively, in designing such authentication protocols for the grid existing authentication protocols and standards for Internet systems should be leveraged. Development of authentication protocols for Internet systems using cryptographic tools and techniques has been a practice for more than two decades. However, prior work has shown that if adequate care is not taken the process is prone to significant errors [23, 3, 18, 2, 16, 30, 1]. To help avoid such errors researchers have developed design principles to serve as guidelines that have subsequently been quite effective in helping avoid major errors in new protocols.

In this work we argue that while these design principles are relevant to authentication protocols for the

[1] To appear in Proceedings of the Forty-Third Annual HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, Hawaii, January 2010

electric grid they do not address some of the unique constraints and properties of the grid that distinguish it from Internet systems. For example, the real-time nature of grid applications, the need for high availability and the typical long-term deployed life of grid devices. To address these gaps we build upon prior work in design principles for authentication protocols but focus the discussion on grid-specific issues. In Section 2 we identify fundamental constraints and properties of the power grid that should influence protocol design. In Section 3 we review some interesting attacks against previous authentication protocols as well as sound design principles proposed in the literature. For each such principle we comment on its applicability to the grid. In Sections 4 and 5 we discuss design issues for efficiency, availability and evolvability that lie at the heart of grid constraints. We conclude with a brief look at the process of designing secure protocols beyond these design principles.

2. The Grid: Today and Tomorrow

Developing standards is a time-consuming process so a key motivation is to ensure that the resulting standards are relevant for a long period of time. In this section we consider some constraints and properties of the electric grid that impact the effectiveness of authentication standards over time.

The electric grid today comprises a wide range of computation and communication systems that support data exchange using multiple formats and protocols for a wide range of grid applications. Networking technologies in use range from 1200 baud serial links to high-speed fiber-optic TCIP/IP routed networks. Computation devices range from low end 8-bit processors to modern high-end 64-bit multiprocessor servers. These devices and networks support data exchange with legacy and new protocols. For example, protocols used in transmission systems for supporting data exchange include MMS, DNP3, ICCP and IEC 61850. These data exchanges support grid applications with varying timing requirements. The transmission system, for example, operates at granularities ranging from milliseconds to whole days. Protection and control protocols have full data exchanges in milliseconds. State estimators operate on the order of minutes, contingency analysis on the order hours, and day-ahead power markets on the order of days.

For authentication standards that need to be deployed across the grid, one can therefore characterize common constraints of the grid today as follows: devices have limited computation capabilities, networks have limited bandwidth, there is a need to integrate legacy protocols and systems, and there is a lack of a system-wide cyber security infrastructure. That said, the entire cyber infrastructure of the grid is being modernized for the past few years and, in fact, the process has picked up pace given recent investments. These modernization efforts can be expected to significantly change the landscape and within a few years, say 5 to 10 years, eliminate many of these common constraints. These efforts will not only improve the cyber infrastructure in terms of high-speed networks and advanced computation devices, they will also result in the deployment of a core cyber security infrastructure and advanced multi-layer gateways that provide protocol translation and tunnels much in the way Internet systems have evolved. After a period of time one can conceive of a grid that largely comprises high-speed high-bandwidth networks, high-performance devices, protocol translators and gateways that help deal with legacy problems and a core cyber security infrastructure for trust establishment and enforcement.

Whether the grid comprises an advanced or an outdated cyber infrastructure (or perhaps a combination), it is important to realize that the grid will always be different from Internet systems in fundamental ways [29, 15]. These differences arise largely from the critical and real-time nature of electric grid systems, their impact on the safety of equipment and personnel and their long component life. These differences lead to constraints and properties that impact cyber security protocols, in general, and authentication protocols, in particular. Specifically, grid applications require 1) high performance in terms of latency and jitter in message exchange, 2) high availability in terms of tolerating faults and failures (e.g., they need graceful degradation of services), 3) timeliness in that computation and communications subsystem must meet real-time requirements of applications, 4) comprehensive security design as they are likely targets for sophisticated cyber attacks and 5) adaptable and evolvable designs because components typically have a lifetime of 15 or more years once deployed. We argue that protocols must be developed with these fundamental constraints and properties and not significantly influenced by today's constraints that are likely to be eliminated with modernization.

3. A Principled Approach to Authentication

Authentication mechanisms are used to "corroborate that an entity is the one that is claimed" according to the international standard ISO/IEC 9798-1 [20]. In [16], Gollmann stipulates this high level goal to be equivalent to message origin authentication and replay prevention, and discusses some common approaches for achieving authentication. For example, when an entire session has to be authenticated, a common approach is to establish

Table 1. Examples of Attacks on Security Protocols

Protocol	Attack Description	Comments
Authentication protocol based on symmetric key by Woo and Lam in 1992 [32]	This protocol was analyzed in [2] and it was shown that due to the absence of a principal's name in a particular message there existed insufficient connection between messages which lead to a successful impersonation attack on the protocol.	This protocol violates Principle 1 given in this paper. A similar impersonation attack known as the Canadian attack was found in a draft of ISO/IEC 9798-3. It was also shown that an early version of the SSL protocol failed to provide client authentication due to the lack of explicitness of names.
STS (Station to Station Protocol) by Diffie, Oorschot, and Wiener in 1992 [12].	An attack on this protocol was described in [8] which was based on concurrent protocol runs and application environment. It was originally based on Shoup's attack [27] and Lowe's attack on STS [22]. In this, it was shown that when the given application protocol used STS to establish a session key then the obtained mechanism is vulnerable to impersonation attack. The attack used a weakness in STS which was not exploitable until it was used within the application environment.	This protocol violates Principle 2 and Principle 5 given in this paper. The attack shows the importance of analyzing the concurrent protocol runs and application environment before deploying any protocol.
Kerberos System Version 4 by Steve Miller and Clifford Neuman in 1987	Gong [17] showed that authentication in Kerberos fails not only due to slow clocks but also due to fast clocks. It was shown that incorrect timestamps can lead to replay attacks even when the clocks are resynchronized by using a postdated message (when a fast clock is used to timestamp messages).	This protocol violates principle 4 given in this paper. Note that in [4] Bellovin and Merritt state that Kerberos depends on the mutual trust between client, server, authentication server, and time server. These trust relations are founded on the policy of the protocol, hence trust relations in kerberos are an example of Principle 3.
TMN scheme by Tatebayashi, Matsuzaki and Newmann in 1989 [31].	This scheme was proposed to execute a key exchange between two users by the help of a trusted server. The protocol was analyzed by Simmons in [28]. Simmons found that if two adversaries collude or if one of the users generates predictable random numbers then an attack can be launched wherein the supposed shared secret key can be derived. In this, the server becomes the oracle for gaining the secret key information.	This protocol fails to observe Principle 6 given in this paper. Another subtle example of oracle attack is discussed in [30] wherein oracle attack lead to leaks of critical information.

a session key and use an integrity protection mechanism that leverages the established session key. Another common approach is to use a challenge-response mechanism to authenticate an entity for message exchange.

For power grid authentication protocols there are three potential approaches. First, existing and preferably standardized protocols could be used "out-of-the-box". This is likely to work only in rare cases as at the very least the grid environment is typically different from that envisioned for existing protocols. Second, existing protocols could be customized or adapted to suit the needs of the grid application in mind. Third, if the requirements are fairly unique then a new protocol could be designed.

For all of these approaches it is important to keep in mind that authentication protocols, and security protocols in general, can be surprisingly hard to design correctly as has been observed in practice [17, 28, 2, 22, 30, 27]. This is true even when an existing protocol is placed in a new environment. Learning from experiences afforded by flaws discovered in designed protocols, some of which are shown in Table 1, security researchers have developed several principles that can be very helpful in avoiding pitfalls and common errors [23, 3, 18, 2, 16, 30, 1, 8] both when designing new and adapting existing security protocols. Table 1 shows examples of protocols that were considered secure but were later found to be flawed. Further, we identify sound design principles proposed in the literature in Table 3 and refer to these principles when discussing attacks in Table 1.

While we note that design principles are neither necessary nor sufficient for protocol correctness, we believe that studying them for control systems such as the power grid can significantly help with designing new and customizing existing protocols for the Smart Grid. Below we discuss principles identified in Table 3 that have been gathered from literature and their applicability for designing protocols for the Smart Grid. These principles are discussed with the following in mind: 1) traditional cryptographic tools such as encryption, message authentication codes (e.g., SHA-1), HMAC, symmetric cryptosystems (e.g., AES) and asymmetric cryptosystems (e.g., RSA), 2) cyber attacks such as man-in-the-middle attacks, impersonation, forgery and modification, and 3) protocol goodness properties such as replay prevention, message freshness, and complete and effective state management.

Table 2. Selected Design Principles for Security Protocols

Principle Name	Principle	Comment
Principle 1: Explicit Names	*If the identity of a principal is essential to the meaning of a message, it is prudent to mention the principal's name explicitly in the message [2].*	Specific instantiation of *Explicit Communication* principle of [2].
Principle 2: Unique Encoding	*If an encoding is used to present the meaning of a message, then it should be possible to tell which encoding is being used. In the common case where the encoding is protocol dependent, it should be possible to deduce that the message belongs to this protocol, and in fact to a particular run of the protocol, and to know its number in the protocol [2].*	Similar to principles that appear in [18, 8] and related to principles in [3].
Principle 3: Explicit Trust Assumptions	*The protocol designer should know which trust relations his protocol depends on, and why the dependence is necessary. The reasons for particular trust relations being acceptable should be explicit though they will be founded on judgment and policy rather than on logic [2].*	
Principle 4: Use of Timestamps	*If timestamps are used as freshness guarantees by reference to absolute time, then the difference between local clocks on various machines must be much less than the maximum age for a message to be deemed valid. Furthermore, the time maintenance mechanism everywhere becomes part of the trusted computing base [2].*	Abadi and Needham [2] provide additional insights into using nonces and counters for message freshness that may be useful.
Principle 5: Protocol Boundaries	*Often the specification of a protocol and its verification focus on the core of the protocol and neglect its boundaries. However, these boundaries are far from trivial; making them explicit and analyzing them is an important part of understanding the protocol in context. These boundaries include: (1) interfaces and rules for proper use of the protocol,(2) interfaces and assumptions for auxiliary functions and participants, such as cryptographic algorithms and network services, (3) traversals of machine and network boundaries,(4) preliminary protocol negotiations, (5) error handling [1].*	Principles in a similar vein are discussed in [8]
Principle 6: Release of Secrets	*A protocol should be designed as much as possible to be resilient against the release of secrets, even potential or obsolete secrets [8].*	A similar principle is also discussed in [2].
Principle 7: Explicit Security Parameters	*Be explicit about the security parameters of crypto primitives. A key generation routine should be claimed as good for so many keys; a threshold schemes for resistance to so many conspirators; a block cipher for so many blocks; and so on [3, 2].*	

First, we discuss a few overarching principles. As discussed in [8], when designing or adapting a protocol it is important to know the threat environment in which the protocol is expected to operate and ensure that it is designed to be secure in such an environment. A common mistake is to adopt an existing protocol into a threat environment that is different from what the protocol was designed to be secure against without re-evaluating the security of the protocol in the new environment. Another overarching principle is that of explicitness of communication and is stated as follows in [2]: *Every messages should say what it means: the interpretation of the messages should depend only on its content. It should be possible to write down a straightforward English sentence describing the content—though if there is a suitable formalism that is good too.* Notions similar to this are also found in [6, 33]. Some of the principles, *e.g.*, Principles 1 and 2, discussed below are specific instantiations of this overarching principle.

Principle 1: Explicit Names *If the identity of a principal is essential to the meaning of a message, it is prudent to mention the principal's name explicitly in the message*. Abadi and Needham presented this principle [2] after observing that several leading protocols at that time omitted doing so and, consequently, were vulnerable to attacks. In particular, they show that protocols omitting to include the principal's name can be vulnerable to impersonation attacks as the message recipient cannot cryptographically determine who they are talking to. For authentication in power grid protocols this principle brings to light several issues. First, it is essential that the principle be applied to avoid potential impersonation or other cyber attacks. Second, entities in the power grid are becoming more interactive and may need to communicate with previously unknown entities. This implies that every entity a principal interacts with must have a unique name. Currently, there is no global naming scheme for principals in the power grid. In fact, it is common to find multiple entities with the same name. There are alternative approaches besides a global naming scheme to address this. For example, one can use local naming schemes like those designed by SPKI/SDSI [14] or one can encode pair-wise relations in keys by using unidirectional pair-wise symmetric keys as recommended by the ISO/IEC 9798-4 standard [21]. The draft DNP3 Secure Authentication Supplement uses unidirectional pair-wise symmetric keys to address this problem

[13].

Principle 2: Unique Encoding *If an encoding is used to present the meaning of a message, then it should be possible to tell which encoding is being used. In the common case where the encoding is protocol dependent, it should be possible to deduce that the message belongs to this protocol, and in fact to a particular run of the protocol, and to know its number in the protocol*. Abadi and Needham presented this principle [2]. It aims to prevent, 1) interleaving attacks where messages from one run of the protocol are inserted into another run of the protocol or into a different protocol, and 2) parsing ambiguity attacks [10] where an adversary forces an entity to parse a message in a different way. Similar notions to this are discussed in [18, 8] and related principles are discussed in [3]. For authentication in power grid protocols this principle brings to light several issues. First, security is being built into existing legacy protocols which don't seem to have any protocol identifiers. Second entities in the power grid are becoming more interactive and are interacting using many different protocols increasing the potential for interleaving attacks.

Principle 3: Explicit Trust Assumptions *The protocol designer should know which trust relations his protocol depends on, and why the dependence is necessary. The reasons for particular trust relations being acceptable should be explicit though they will be founded on judgment and policy rather than on logic*. Abadi and Needham [2] argue for caution and clarity on which entities are trusted for the correct execution of the authentication protocol and to what extent. For example, when using timestamps the time server must be trusted to maintain accurate time. In the context of the power grid, this principle argues for the need to clearly state and analyze all trusted entities and the extent of trust in them; e.g., users that manage keys and passwords, servers that manage keys or time, software that generates nonces and sequence numbers, and networks that deliver messages.

Principle 4: Use of Timestamps *If timestamps are used as freshness guarantees by reference to absolute time, then the difference between local clocks at various machines must be much less than the allowable age of a message deemed to be valid. Furthermore, the time maintenance mechanism everywhere becomes part of the trusted computing base*. This principle by Adadi and Needham [2] argues that if timestamps are used for message freshness guarantees then the timestamp services must be trustworthy and reasonably well synchronized. In the power grid several protocols already provide time synchronization services (e.g., DNP3) and, therefore, it may be tempting to use timestamps for message freshness. However, there are two issues that must be carefully analyzed before doing so. First, deeming the timestamping and synchronization as trustworthy enough for authentication protocols may require additional monitoring and verification capabilities than what is currently available. Second, the granularity of time synchronization may not be sufficient for the high frequency messaging often found in the grid; e.g., UTC provides millisecond level synchronization, however, an entity may send or receive more than one message per millisecond. Abadi and Needham [2] provide additional insights into using nonces and counters for message freshness that may be useful. Also, we note that ISO/IEC 9798-4 [21] provides example authentication protocol sequences that use timestamps, nonces or counters for message freshness.

Principle 5: Protocol Boundaries *Often the specification of a protocol and its verification focus on the core of the protocol and neglect its boundaries. However, these boundaries are far from trivial; making them explicit and analyzing them is an important part of understanding the protocol in context. These boundaries include:(1) interfaces and rules for proper use of the protocol,(2) interfaces and assumptions for auxiliary functions and participants, such as cryptographic algorithms and network services,(3) traversals of machine and network boundaries,(4) preliminary protocol negotiations,(5) error handling*. Abadi [1] points out that security protocols do not execute in isolation but rather in an environment. Therefore, it is crucial to understand this environment and ensure that the protocol can function correctly in that environment. For the power grid, this principle has far reaching consequences and to the extent possible a thorough analysis of the environment is essential. This potentially includes the underlying messaging protocols, real-time nature of control systems, legacy integration issues, choice of cryptographic primitives, networking constraints and error handling.

Principle 6: Release of Secrets *A protocol should be designed as much as possible to be resilient against the release of secrets, even potential or obsolete secrets*. Canetti *et al.*[8] derive this principle from the commonly occurring attacks like blinding attacks and compromise of old secrets. It suggests that one should not only be careful about the current keys but also about the secrets used in the past. In the design of power grid protocols, one should keep this principle in mind, as there are many remote devices that could be compromised. For example, if common keys are used across the system then a

compromise of a small number of remote devices could result in the compromise of a large number of keys. Similar principles are also discussed in [3, 2].

Principle 7: Explicit Security Parameters *Be explicit about the security parameters of crypto primitives. A key generation routine should be claimed as good for so many keys; a threshold schemes for resistance to so many conspirators; a block cipher for so many blocks; and so on*. Anderson and Needham proposed this principle, in [3], keeping in mind that most cryptographic primitives have limitations. Such limitations should be made explicit and taken into account, for example when specifying key refresh times. There are many remote devices in the power grid that operate in deploy-and-forget mode. Making the security parameters explicit and taking them into account in prescribing key refresh intervals for example will help balance security needs with the desire for low management overheads.

4. Design for Efficiency

A few key issues to consider when developing secure authentication protocols for the smart grid environment are efficiency of the protocol, availability of both the authentication mechanism and the system incorporating it and evolvability of the protocol. Efficiency and availability are important given that real-time critical applications need to be supported by the grid. Evolvability is important as devices deployed in the grid will likely be in operation for a long time. In this section we discuss some approaches and trade-offs for achieving efficiency while in the next section we consider availability and evolvability.

In the grid, high priority is given for deadline driven processes that support real-time critical grid applications. While it is essential that messages exchanged in such environments be authentic, the authentication protocol itself must not prevent timely execution of the processes. That is, the specific cryptographic primitives and parameters and protocol sequences must be sufficiently efficient in the context of the grid application where the authentication protocol is employed. Efficiency is gauged in terms of computation and communication overhead imposed by the authentication protocol as they add additional latency. Computation overhead arise from expensive operations such as cryptographic operations (e.g., HMAC, symmetric or asymmetric encryption, digital signatures) while communication overhead arise from additional rounds of messages and message expansion due to the authentication protocol (e.g. added sequence numbers, nonces, digital signatures).

Communication Overhead Communication overhead can be reduced by minimizing the additional number of messages needed by the authentication protocol. For example, there exist both one-pass, *i.e.*one message, and two-pass, *i.e.*two message, unilateral authentication mechanisms (only one entity is authenticated to the other). One could opt for a one-pass unilateral authentication mechanism instead of a two-pass one to reduce communication overhead. However this choice has implications on the necessary capabilities of devices using the mechanisms and the environment, and on the properties achieved by the protocol. For example, one-pass unilateral authentication can be achieved using timestamps or sequence numbers, requiring a trusted time synchronization base or state maintenance respectively, while a two-pass unilateral authentication can be achieved using random numbers that avoid the need for additional resources [21]. Similarly, in some scenarios it may be sufficient to just use unilateral authentication as opposed to mutual authentication (both entities are authenticated to each other) that typically needs a larger number of messages.

Communication overhead can also be reduced by minimizing the size of additional messages needed by the authentication mechanism and the size of additional fields added to existing messages by the authentication mechanism. For example, one could use smaller digital signatures such as the 326 bit Elliptic Curve Digital Signature Algorithm (ECDSA) signatures versus the 1024 bit RSA signatures. It is important to note that this choice may affect the computational efficiency of the protocol as verifying an RSA signature is significantly faster than verifying an ECDSA signature on common platforms. Similarly one could opt for using symmetric key based Keyed Message Authentication Codes (MACs) instead of asymmetric key based digital signatures to reduce byte overhead. For example, using HMAC-SHA256 produces a 256 bit MAC versus using 1024 bit RSA that produces 1024 bit signatures. Furthermore, HMACs are about 3 orders of magnitude more efficient to compute than digital signatures. Even when using HMACs one could further reduce the communication overhead by using truncated outputs. Interestingly, there are no formally established results regarding the security of a truncated HMAC but standards such as NIST Special Publication 800-107 [11] currently recommend 64 to 96 bit output as providing sufficient security for most applications.

Another way to reduce communication overhead is by reducing the size of the sequence number field and re-keying more often to keep the sequence number from repeating for any given key. This may turn out to be more efficient in terms of bandwidth consumption de-

pending on the reduction in the field size, number of messages exchanged before a re-key is needed and the cost of re-keying.

Computation Overhead Computation overhead can be reduced by minimizing the number of expensive (*e.g.*cryptographic) operations in an authentication mechanism. Cryptographic operations such as exponentiations in finite groups are orders of magnitude more expensive than additions or multiplications. For example, Abadi and Needham demonstrate in [2] (Example 6.1) how they were able to reduce the number of encryption operations needed by Otway and Rees protocol [26] by explicitly adding participating entity names in messages. This also reduced the bandwidth overhead of the protocol. Another way to reduce overall computational overhead is by making the common operations faster. For example, if signature verification is a lot more common operation than signing then it pays to pick a signature scheme whose signature verification operation is much more efficient. RSA signature scheme is one such example where verification operation is an order of magnitude faster than its signing operation.

Typically, symmetric key based cryptosystems are considered to be more efficient than their asymmetric counterparts. For example, digital signatures can be two orders of magnitude more expensive than HMACs in terms of computation. Therefore, for authentication in networks with frequent messaging (e.g., SCADA), digital signatures must only be used after careful analysis. However, they do have distinct benefits over symmetric key based cryptosystems. For example, if non-repudiation of messages is needed then signatures may turn out to be more efficient overall when compared to non-repudiation solutions with symmetric cryptosystems. Similarly, key distribution and management can be simpler when using asymmetric cryptosystems.

Clearly there is some tension between optimizing an authentication protocol for efficiency and its security as seen from some of the trade-offs discussed above. This is also implied by the more general and overarching principle that all communication must be explicit. Thus optimization of a security protocol should be undertaken with care.

5. Design for Availability and Evolvability

Given the critical nature of the power grid, it is clear that availability is a crucial concern. In fact, it is often argued that among the three key cyber security properties of availability, integrity and confidentiality, availability is the most important for grid systems and other critical infrastructure. In terms of protocols such as those for authentication, availability implies three major requirements. First, the protocol must be efficient in its use of computation and communication resources so that resources do not get overwhelmed and all requests can be handled. Second, the protocol must have good error management built into it to ensure proper handling of failures (e.g., those resulting from bad messages). Furthermore, the error management functions must be fail-safe in nature so they do not themselves lead to resource exhaustion even in the face of adversarial action. Third, the protocol should support auxiliary security functions that may be deployed in the grid cyber system to detect to and respond to cyber attacks. The issue of efficiency was addressed in detail in Section 4 above. We now discuss the remaining two requirements below.

Error Management Error management represents a significant boundary when designing a protocol. There are many aspects to it, and frequently interactions of error conditions can lead to subtle, but exploitable, errors. Increased complexity in the error management process, leads to an increase in edge cases as well, so there exists a tension in design for this component as well. Common techniques for smart error management include back-off timers, limits on number of events reported, event reporting compression and suppression techniques, and both in-band and out-of-band reporting. Overall, the idea of correct error handling is highly contextual and includes many trade-offs that need careful consideration. In practice, most protocols are built upon the idea of a state machine. Such state machines allow for easy error detection and handling, as well as providing other benefits to protocol designers. In such cases, all input to the state machine that does not lead to a good state should create an error condition. In a state machine this means entering an error state or returning to a previous safe state. In both cases the correct state transition needs to be carefully analyzed.

Consider a common error management issue. Protocols which include weak or no mechanism for handling malformed or unexpected packets can frequently have implementations which crash or execute arbitrary code after receiving such packets. Such protocols are commonly discussed as "lacking appropriate bounds checking", which means checking for errors on functions related to allocation or assignment. When bounds checking is included care is needed because such bounds checking may cause problems itself, for example, protocols with error counters and lockouts or fall-back mechanisms can suffer denial of service attacks; e.g., from an adversary that forces an error counter increase using malformed messages until lockout occurs. Further, it is helpful to be aware that error reporting can be used

as a mechanism to facilitate information gathering. This could be used for attacks, reverse engineering, or general reconnaissance for the later use. For example, in penetration testing the technique of fuzzing protocols relies on the error reporting mechanisms to determine when it has succeeded in finding an unhandled condition or edge case.

Supporting Cyber Attack Detection and Response While a carefully designed authentication protocol can offer strong protection against attacks, it cannot by itself provide all the necessary security in an operational environment. For example, if an adversary penetrates the network and launches a denial-of-service attack by flooding the network, the authentication protocol can easily get overwhelmed. Given that the protocol will eventually be deployed in an operational power grid environment, designing it to support detection and response services will greatly strengthen the overall system security. For example, consider an adversary that attempts to lock out users by forcing errors via malformed messages. If the authentication protocol includes detailed error counting and can record such reports in a data object accessible by supporting security services (deployed in the operational environment), then the system can be significantly more effective overall at detecting and responding to cyber attacks. Creating such detailed error counters, data objects for maintaining the counters and an interface for accessing the objects is one good strategy for supporting cyber attack detection and response.

Design for Evolvability In the power grid, cyber systems and embedded security protocols are expected to last in the field much longer than those in typical Internet systems. Over such a long period of time the security protocols may need to be updated or modified in order to ensure continued secure operation. Changes may be required if the protocol is found to have vulnerabilities or inefficiencies. Alternatively, the underlying cryptographic primitives may be deemed weak in light of future discovered vulnerabilities, or because the key sizes are no longer considered appropriate. For example, primitives such as MD5 and DES are now considered too weak because practical attacks against them have been discovered. More recently, researchers have identified weaknesses in SHA-1 [9] and AES [5] suggesting that practical attacks against these newer primitives may be discovered in the future. In terms of key sizes NIST provides guidelines on recommended lifetimes of common key sizes for common primitives [25].

It is important to note that just because a primitive is deemed weak or a particular particular protocol step can lead to newly discovered attacks, the entire protocol need not be discarded from a security point of view (or from a business point of view). Instead, if the protocol is designed in a *modular* manner then it will be possible to replace weak components of the protocol with strong ones, upgrade the protocol through necessary hardware and software changes, and continue the use of the protocol. That is, modularity can allow for evolvability of the protocol.

Researchers have studied protocol modularity extensively. Cannetti [7] provides the following simple modular design methodology for secure protocols:

> "(1) Design a high-level protocol for the given task assuming that other, simpler, subtasks can be carried out securely.
> (2) Design protocols that securely carry out these simpler subtasks.
> (3) Construct a full-fledged protocol for the given task by plugging the simpler protocols as subroutines in the high-level protocol."

Such an approach creates the possibility of substituting embedded steps of the simpler tasks without affecting the major tasks. However, the environment in which the protocol(s) are intended to be used in should be taken into account during the design process as discussed in Principle 5 and [8]. Similarly, object-based design approaches can allow for the substitution of specific cryptographic primitives and ciphers without rewriting the entire protocol.

6. Conclusion

In this work we propose design principles, best practices and guidelines for the development of power grid cyber security authentication protocols. We first reviewed past work in both development of authentication protocols and vulnerabilities and attacks discovered against authentication protocols. We then summarize key design principles that others have proposed in the path to help improve protocol security. Motivated by the power grid we then focus on three key aspects of power grid cyber security protocols, namely, efficiency, availability and evolvability. For each of these aspects we present an analysis of potential hurdles and pitfalls and high-level approaches for addressing them. While our analysis is focused on authentication protocols it applies to most cyber security protocols such as those for key exchange, encryption and non-repudiation. However, each of those protocols deserves a similar analysis to develop a comprehensive set of key design principles.

Protocol development with the design principles is very helpful but not sufficient for ensuring security of

protocols. As recommended by Syverson [30] design principles should be used at the beginning middle and at the end of designing a protocol. They will guide preliminary design, specification evaluation and final verification at the end. In addition, protocol security can be greatly enhanced by having it reviewed by relevant experts and/or via the use of formal verification tools. External expert reviewers can use a variety of tools (including formal verification) to ensure that security goals of the protocols are met and potentially provide enhancements to mitigate vulnerabilities. Automated formal verification tools such as model checkers and theorem provers [24] conduct an exhaustive attack analysis and provide valuable insights into potential attacks and weaknesses and have been used extensively in protocol analysis and verification.

In this work we focused on protocol development that may be standardized for broad adoption. Just like the design principles and tools mentioned in this work can help improve the security of the protocol, similar effort is needed to ensure that the software and hardware implementations do not introduce vulnerabilities. An analysis of such vulnerabilities for the power grid environment would be an interesting topic for research.

Acknowledgements

We would like to thank Madhava Sushilendra and Grant Gilchrist for discussions on DNP3 and DNP3 Secure Authentication Supplement. This material is based upon work supported by the National Science Foundation under Grant No. CNS-0524695.

References

[1] M. Abadi. Security protocols and specifications. In *FoSSaCS '99: Proceedings of the Second International Conference on Foundations of Software Science and Computation Structure, Held as Part of the European Joint Conferences on the Theory and Practice of Software, ETAPS'99*, pages 1–13, London, UK, 1999. Springer-Verlag.

[2] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, 1996.

[3] R. J. Anderson and R. M. Needham. Robustness principles for public key protocols. In *CRYPTO '95: Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, pages 236–247, London, UK, 1995. Springer-Verlag.

[4] S. M. Bellovin and M. Merritt. Limitations of the kerberos authentication system. *SIGCOMM Comput. Commun. Rev.*, 20(5):119–132, 1990.

[5] A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir. Key recovery attacks of practical complexity on aes variants with up to 10 rounds. Cryptology ePrint Archive, Report 2009/374, 2009. `http://eprint.iacr.org/`.

[6] C. Boyd and W. Mao. On a limitation of ban logic. In *EUROCRYPT*, pages 240–247, 1993.

[7] R. Canetti. Security and composition of multi-party cryptographic protocols. *Journal of Cryptology*, 13:2000, 1999.

[8] R. Canetti, C. Meadows, and P. Syverson. Environmental requirements for authentication protocols. In M. Okada, B. Pierce, A. Scedrov, H. Tokuda, and A. Yonezawa, editors, *LECTURE NOTES IN COMPUTER SCIENCE*, volume 2609, pages 339–355, 2003.

[9] C. D. Cannière and C. Rechberger. Finding sha-1 characteristics: General results and applications. In *ASIACRYPT*, pages 1–20, 2006.

[10] L. Chen and C. J. Mitchell. Parsing ambiguities in authentication and key establishment protocols. Cryptology ePrint Archive, Report 2008/419, 2008. `http://eprint.iacr.org/`.

[11] Q. Dang. Recommendation for Applications Using Approved Hash Algorithms. National Institute of Standards and Technology. Special Publication.

[12] W. Diffie, P. C. van Oorschot, and M. J. Wiener. Authentication and authenticated key exchanges. *Des. Codes Cryptography*, 2(2):107–125, 1992.

[13] DNP3 Users Group Technical Committee. DNP3 Secure Authentication Specification Version 2.0, DNP Users Group Documentation as a supplement to Volume 2 of DNP3. Technical report, DNP Users Group, 2008.

[14] C. Ellison, B. Frantz, B. Lapson, R. Rivest, B. Thomas, and T. Ylonen. SPKI Certificate Theory. Technical report, September 1999.

[15] T. Fleury, H. Khurana, and V. Welch. Towards A Taxonomy Of Attacks Against Energy Control Systems. *Critical Infrastructure Protection II, The International Federation for Information Processing*, 290, 2009.

[16] D. Gollmann. What do we mean by entity authentication? In *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, page 46, Washington, DC, USA, 1996. IEEE Computer Society.

[17] L. Gong. A security risk of depending on synchronized clocks. *Operating Systems Review*, 26(1):49–53, 1992.

[18] L. Gong. Fail-stop protocols: An approach to designing secure protocols. *Dependable Computing for Critical Applications 5*, 1995.

[19] International Electrotechnical Commission. IEC 62351-5 Power systems management and associated information exchange - Data and Communication Security - Part 5: Security for IEC 60870-5 and Derivatives. Draft, International Electrotechnical Commission.

[20] International Standards Organization and International Electrotechnical Commission. ISO/IEC 9798-1:1997 Information technology – Security techniques – Entity authentication – Part 1: General. Standard, International Organization for Standardization and International Electrotechnical Commission, 1997.

[21] International Standards Organization and International Electrotechnical Commission. ISO/IEC 9798-4:1999 Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function. Standard, International Organization for Standardization and International Electrotechnical Commission, 1999.

[22] G. Lowe. Some new attacks upon security protocols. In *CSFW*, pages 162–169. IEEE Computer Society, 1996.

[23] W. Mao and C. Boyd. Development of authentication protocols: some misconceptions and a new approach. pages 178–186, Jun 1994.

[24] C. Meadows. Formal methods for cryptographic protocol analysis: emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, 21:44– 54, January 2003.

[25] NIST. NIST 800-57: Recommendation for key management - Part 1: General. National Institute of Standards and Technology, 2007. Available at `http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part1.pdf`.

[26] D. J. Otway and O. Rees. Efficient and timely mutual authentication. *Operating Systems Review*, 21(1):8–10, 1987.

[27] V. Shoup. On formal models for secure key exchange. Technical Report RZ 3120 (#93166), 1999.

[28] G. J. Simmons. Cryptoanalysis and protocol failures. *Commun. ACM*, 37(11):56–65, 1994.

[29] K. Stouffer, J. Falco, and K. Kent. Guide to SupervisoryControl and Data Acquisition (SCADA) and Industrial Control Systems Security. *Recommendations of the National Institute of Standards and Technology. Special Publication*, pages 800–82.

[30] P. Syverson. Limitations on design principles for public key protocols. In *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, page 62, Washington, DC, USA, 1996. IEEE Computer Society.

[31] M. Tatebayashi, N. Matsuzaki, and D. N. Jr. Key distribution protocol for digital mobile communication systems. In *CRYPTO '89: Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, pages 324–334, London, UK, 1990. Springer-Verlag.

[32] T. Y. C. Woo and S. S. Lam. Authentication for distributed systems. *Computer*, 25(1):39–52, 1992.

[33] T. Y. C. Woo and S. S. Lam. A lesson on authentication protocol design. *Operating Systems Review*, 28(3):24–37, 1994.