Cryptography Concepts and Effects on Control System Communications

Rhett Smith
Schweitzer Engineering Laboratories, Inc.

Published in
Sensible Cybersecurity for Power Systems: A Collection of Technical Papers Representing Modern Solutions, 2018

Previously presented at
Saudi Arabia Smart Grid 2013, November 2013

Originally presented at the
11th Annual Western Power Delivery Automation Conference, April 2009

Cryptography Concepts and Effects on Control System Communications

Rhett Smith, GSEC, CISSP, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—This paper provides a high-level explanation of the cryptographic components that can be used in control system communications and the risks these components mitigate. This paper makes the vocabulary and technology of cryptography understandable so that engineers know what tradeoffs must be considered to select the correct tool for the job. Discussions and examples cover authentication versus encryption, confidentiality, data integrity, key management, and operational concerns.**

I. Introduction

Cryptography provides a variety of powerful cybersecurity tools. It enables more interproduct communications, centralized data collection, and remote access, which result in increased workforce efficiency. Cryptography began as the science of hiding information. This science has expanded far beyond the goal of keeping the data confidential; it now includes integrity checking and authentication.

Control systems have operational priorities where safety and availability come before security. This means that the selection of a cryptographic solution to secure the communications channel must not impact the safety and availability of that communication. As with all engineering tasks, if the wrong tool is selected for the job, disaster can result. But when selected correctly, cryptography can enable an organization to do many tasks more efficiently and effectively. This is accomplished by allowing the information to be stored in an easily accessible location while reducing the risk of unauthorized access or providing the infrastructure to allow secure remote access.

II. Control System Communication

Many types of communications systems are used in control systems, ranging from public to private serial or routable networks. Examples of electric utility communications links include:

- Radio frequency (RF)
- Dedicated fiber
- Ethernet
- Public telephone
- Telecommunications network

These communications links are used to perform three primary functions:

- Real-time protection
- Supervisory control and data acquisition (SCADA)
- Engineering access

Real-time protection communication is a high-priority communication that measures and controls the electrical power system. The inclusive data are only important for a short amount of time. Data payloads range from a few bytes for pilot protection to a hundred bytes for line current differential protection. This type of communication between devices needs to be timely (within milliseconds) and follows an unsolicited communications structure. Two examples are IEC 61850 GOOSE and Mirrored Bits® communications.

Based on the nature of this type of data, the security focus should be on protecting the data in transit and authenticating the sender, instead of protecting the data after they are received and processed.

SCADA communications are a lower priority than real-time protection but are given high priority in operations. Like real-time protection, the data communicated on SCADA also have a finite life span, but the time between data updates is usually farther apart than with real-time protection. SCADA communication consists of a poll/response format. Examples include DNP3, Modbus®, or IEC 61850, with update rates ranging from hundreds of milliseconds to minutes.

Similar to real-time protection, SCADA communications security is focused on protecting the data in transit and authenticating the sender, instead of protecting the data after they are received and processed.

Engineering access communication is used to configure and monitor the electronic devices that make up a control system. These communications are often set up to allow remote access so that the devices can be centrally managed. Unlike real-time protection or SCADA communication, the timing is not critical. These communications require seconds and not microseconds. Retrieving event reports, accessing fault locations, and changing configuration settings are examples of engineering access communications data. The life span of the data is longer than that of SCADA.

Engineering access communications security is much different than either real-time protection or SCADA communication. Engineering access data need to be protected in transit. After the data are received and processed, authentication of the sender and the data must be protected. This is because the data may have passwords or other sensitive material that have long life spans.

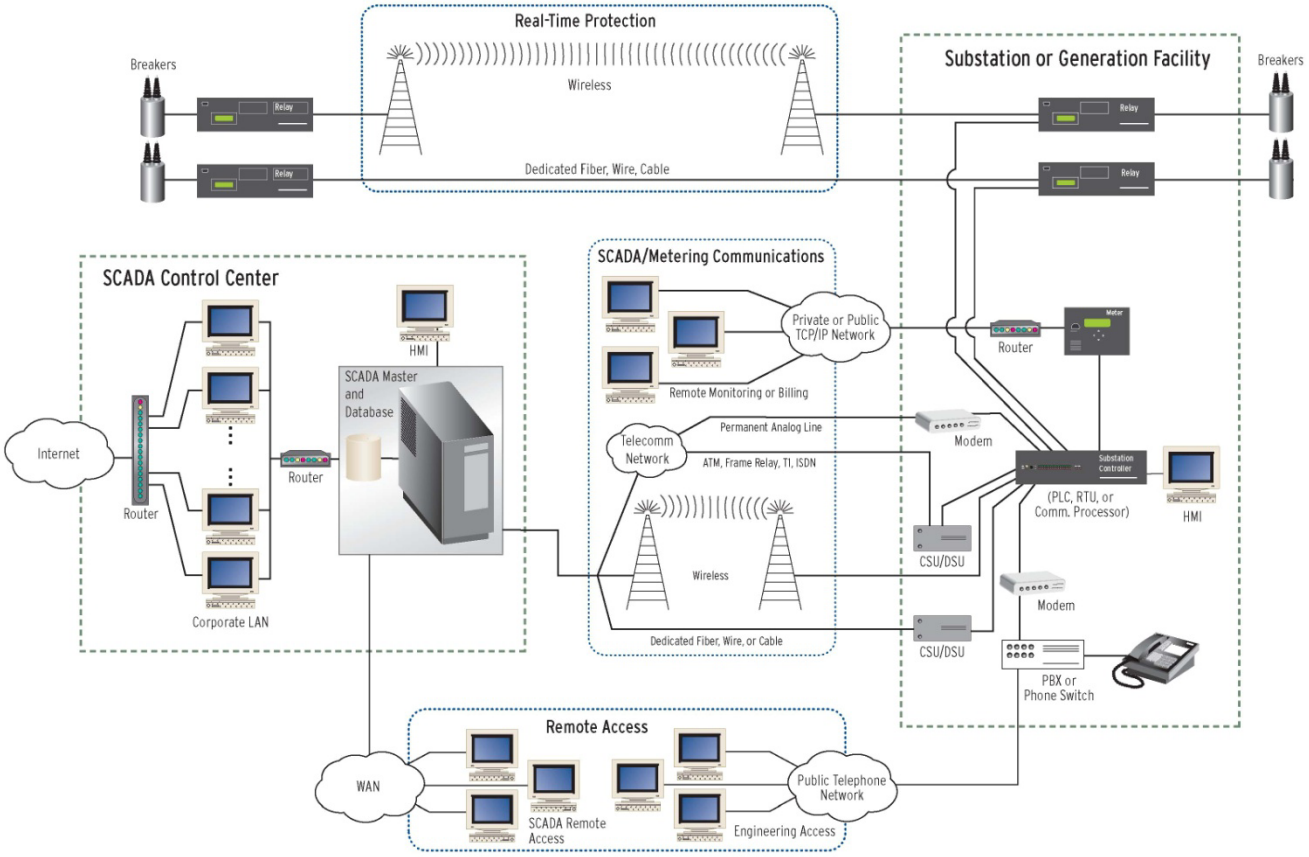

Fig. 1. Electric Utilities Communications Links

Fig. 1 illustrates how one control system integrates a wide variety of communications and combines the three types of communications just discussed.

Ethernet or synchronous optical network (SONET) communications systems are becoming more popular because of their diversity and economical advantage. However, serial communications are still the bulk of the installed base and the focus of this paper. Serial networks are primarily defined as data rates from 300 bits per second (bps) to 115200 bps.

Serial protocols and architectures were designed many years ago with little to no thought about security. At the time of their development, most serial protocols used in the United States were assumed to be integrated into trusted private networks, and data security was never a concern. Recently, there are efforts to redesign serial protocols with cryptographic cybersecurity. A good example of this is DNP3, which has added cryptographic authentication and integrity checking to the protocol.

By adopting more public communications infrastructures and automation, companies tend to increase productivity and decrease operational expenses. In doing this, it is imperative to secure these communications links and select the appropriate cryptography for each. However, adding cryptographic security to existing communications links is a challenge. The largest challenge is finding the additional bandwidth or operational latency the cryptographic functionality requires. The impact of bandwidth consumption or data latency will vary, depending on the communications link and the type of cryptography selected.

Engineering access communications links are not time-sensitive, so adding cryptographic mechanisms does not adversely impact the communications operations. However, when adding cryptography to time-sensitive real-time protection or SCADA communication, bandwidth and data latency are major concerns. Cryptographic algorithms and schemes require additional bytes of data to be added to the frame in order to keep the communication synchronized and provide the basis for frame security. These additional data could be headers, trailers, initial values, known values, or counters. It is important to calculate the timing these additions will create. For example, in a SCADA system, assume there is a master that polls outstation devices. The SCADA master requires data to update every two seconds for proper system operation. The total number of outstations that can be polled will depend on the poll requirement divided by the maximum polling cycle time plus the cryptographic overhead plus the processing time:

maximum update time / (maximum outstation response time + cryptographic overhead + process timing)
= outstations that can be polled

III. Cryptography

Cryptography has tools to provide information across a public communications system that can only be understood by authorized users, with confidence that it is accurate, has not been altered in transit, and authenticates the identity of the sender. Selecting the correct cryptographic tool for a system depends on the business objectives of the organization. Each tool comes with its own set of advantages and disadvantages, along with operational and technical costs. Cryptography has three main functionalities—confidentiality, integrity, and authentication. Looking closer at each of the functionalities cryptography provides, we see:

1. Confidentiality conceals information from any unintended viewers. This is done using encryption, which scrambles information only authorized users can unscramble.
2. Integrity ensures information has not been tampered with or altered from its last known good state.
3. Authentication provides proof that a message was sent by a specific entity. Nonrepudiation ensures senders cannot claim they did not send it.

All of these functions require the endpoints be able to keep a secret. This secret is the key value. There are two types of cryptographic key systems, symmetric and asymmetric. Symmetric key systems use the same key to apply cryptography as they do to reverse the process. Symmetric cryptography is computationally efficient but requires that each sender/receiver pair have an identical secret key. How the sender and receiver initially agree on a secret key can cause an operational challenge. If the communications channel is untrusted, passing a symmetric key needs an out-of-band transport solution, such as a telephone or a secure courier.

Asymmetric key systems use a key pair. One key is private and is not shared, and one is public and is shared broadly. In an asymmetric system, the public key is used to apply cryptography, whereas the private key is used to reverse the process. Asymmetric cryptography is much more computationally burdensome but does not have the initial key distribution problem discussed with symmetric keys because the receiver can send a public key to everyone without care of disclosure. In modern cryptographic systems, hybrid systems use asymmetric key cryptography to communicate symmetric keys. This solves the initial symmetric key exchange problem and leverages the speed of symmetric key cryptography.

Key space is defined as all the possible key combinations the system can have. Some alphabetic encryption systems use a code word as a key. The key space would then be all possible letter combinations for that code word. A four-letter code word is weaker than a ten-letter code word. With binary data, the key space is 2^n possibilities, where n is how many bits long the key is. For example, with a 128-bit key, there are 340,282,366,920,938,463,463,374,607,431,768,211,456 possibilities for that key. It would take a computer with a 2.5 GHz processor $4.3 \cdot 10^{21}$ years to guess the correct key if the processor had a new guess on every clock cycle and guessed the correct key on the last guess.

Control systems are designed to last 20 years or more, so it is important to select a key that will be strong enough to withstand the test of time. Computer processing power doubles every two years. Applying this fact to control system equipment that may be installed for 20 years shows that the key space must be strong enough to protect against computers running 2.560 THz processors. Looking again at the 128-bit key example, we see that it would take $4.2 \cdot 10^{18}$ years to break the key under the same assumptions. This shows that selecting key space of 128 bits or greater is appropriate.

Operational considerations focus on key management. Key management includes all the information and configurations needed for all parties to successfully communicate. This includes the policies to follow and the procedures to establish communication. There are two methods to accomplish this, preshared and central server-based control. In a preshared system, all the information and configurations needed for both sides of the communication to talk to each other are preconfigured in the devices before they are deployed for service. The advantage of this is the devices need little to no direction to start communicating, and the devices do not require communication to a central PC. The disadvantage is that in a dynamic system, it is difficult to keep up-to-date because the process to load new configurations is burdensome. Centrally managed servers, on the other hand, provide one initially trusted source for all devices to communicate to. This way the amount of preshared information needed is small, and the configuration will not change as often. The system then leverages this one trusted source to acquire any additional rules needed to communicate to other devices in the same system. When a device wants to communicate to another device, it will ask the central server how to do this. This system is scalable, and one central change is propagated throughout the system. The disadvantage is that every device in the system must be able to communicate with the central server any time it needs to talk to any device in the network.

Once all the cryptographic tools are understood, two more considerations (the technical and operational costs of using cryptography) must be analyzed to select the correct tool for the job.

A. Confidentiality

Confidentiality, or encryption, hides the content of a message from unauthorized viewers. “Hiding” in encryption terms does not mean the unauthorized viewer does not see the message being transmitted but that the viewer cannot understand what the message means. For example, if a malicious person tapped the phone line, that person could see data being communicated, but encryption would prevent that person from understanding what the data mean. Encryption does not provide covert messaging but does provide confidential messaging. This keeps the data scrambled in such a way that no one except the intended receiver can unscramble the message and understand the information.

Early encryption used physical ways of concealing the message, such as the Greek scytale that used a specific-diameter rod to wrap a message around to reveal what was

written on a long, slim strip of leather. Later, stronger encryption methods brought alphabetic replacement. Alphabetic replacement took on many forms, each one more complicated than the next, to protect against cracking or unintended receivers breaking the key and understanding the transmitted information. Today, cryptography is based on mathematical permutation and substitution.

There are three things to consider when selecting the appropriate encryption system to apply to the control system—algorithm and key space, operations, and architecture. With today's mathematical cryptographic systems, it is strongly advised to use 128-bit or larger keys and algorithms that are approved by the National Institute of Standards and Technology (NIST) to ensure mathematical robustness.

Encryption architectures have two parts. The first part is if they are stream or block ciphers. The second is their workflow structure. Stream ciphers encrypt or decrypt information one bit at a time, while block ciphers combine multiple bits to perform encryption functions on blocks of information at once. In real-time systems, it may not be acceptable to wait for larger blocks of data on a serial line to collect before cryptographic functions are performed, whereas in engineering access, this is acceptable. This makes stream ciphers attractive for real-time protection or SCADA control system applications. However, stream ciphers may be susceptible to man-in-the-middle attacks. These types of attacks occur when attackers know the original message that is being sent (such as a DNP3 trip command) and also know how they want to change the message (such as to a DNP3 close command). These attacks are simple to perform if the stream cipher does not include some sort of man-in-the-middle attack prevention.

The workflow of encryption is illustrated in two examples, electronic codebook (ECB) and counter mode (CTR). ECB (shown in Fig. 2) is very fast, and much of the encryption process can be precomputed. The drawback is that for every given block of information, there is one encrypted output. This is a disadvantage if the information sent is often repeated, as in a control system command response environment.

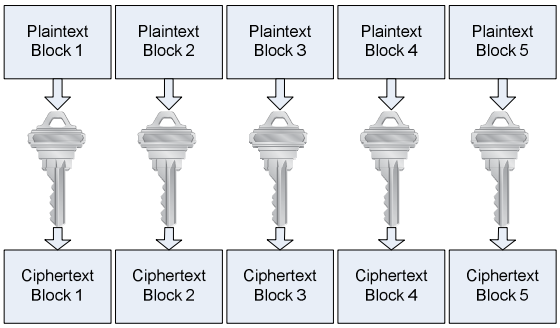

Fig. 2. ECB Example

CTR (shown in Fig. 3) combines the information with a count that changes, so that even if information is sent repeatedly, the encrypted output is different as long as the count does not repeat for the same key.

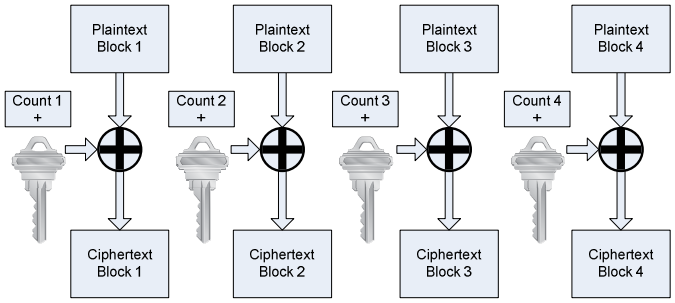

Fig. 3. CTR Example

Shown in Table I are five encryption architectures and some of their design comparisons. The stream or block cipher column denotes how much data are processed at the same time, which results in a requirement of how much data need to be received before calculations can start. The second column shows whether any calculations can be done in advance of receiving any data. The more calculations that can be done in advance, the lower the computational impact is once a message is received. The last column shows if errors cascade. This refers to whether an error occurring in one message corrupts the next message.

TABLE I
COUNTER MODE IMPLEMENTATION EXAMPLES

Modes	Stream or Block Cipher	Advanced Work	Cascading Errors
Electronic Code Book	Block	Yes	No
Output Feedback	Stream	Yes	Yes
Cipher Feedback	Stream	Partial	No
Cipher Block Chaining	Block	Partial	No
Counter	Stream	Yes	No

B. Integrity

Message integrity provides confidence that a message is received just as it was transmitted or that a file has not been altered since the last known good state. This is important when setting up automation or operational processes that are dependent on communications. Integrity is valuable to all three types of control system communications. File integrity is usually provided by running the information through a cryptographic checksum, called a hash. Hashing does not use a key and gives the user a short string of digital data that can be used to easily compare the current state of a file with the previous state. This is usually used when files are stored and the user wants a method to check if the message has changed over time. A keyed hash is used when transporting a message

across a communications channel and allows the user to provide everything hashing does, including protection against anyone (outside of those who know the key) altering the message and appending a new digest. When the receiver gets the message from the sender, the data portion of the message is rerun through the same hash algorithm. If both the calculated and received hash match, the receiver is assured the data integrity is intact. Hashing, keyed hashing, and digital signatures accomplish this process. Hashing is computing the information in a one-way process where the outcome is a fixed-length answer (called a digest) that cannot be reversed, and if any part of the original message is altered, the hash will change. This process uses the following steps:

1. The sender uses the message and the cryptographic keyed hash algorithm to compute a message digest.
2. The message digest is appended to the end of the data message and sent with the message.
3. The receiver separates the original data and the hash digest and computes a new digest using the same key and cryptographic algorithm.
4. If the received message digest and the calculated message digest match, the receiver can be assured the data were not altered (as long as the key is secret).

No matter how large the amount of information that is hashed, the answer will always be the same length. This protects the original information from inference attacks because an unintended receiver of the hash digest cannot assume anything about the original information, including its size. The cryptographic strength is a function of the algorithm and length of the message digest. Other unique qualities of hashing or a hash digest are that the process is one way, no one given the hash digest can reconstruct the original information, and any change in the original information, no matter how small, makes a large change in the hash digest.

C. *Authentication, Authorization, and Nonrepudiation*

Authentication establishes proof of identification and authorization eligibility to have access to specific information. Nonrepudiation provides evidence that a sender or recipient of data cannot deny their actions. To accomplish authentication and nonrepudiation, a keyed hash can be used, but this is limited to authenticating whether the sender knew the secret key or not. A stronger cryptographic method to accomplish this is to use a combination of hashing and asymmetric encryption. This is very similar to a signature on a document, where if a person's signature is on a contract, that person cannot claim they did not agree to it. In cryptography, the sender digitally signs the message, providing proof of who sent it. This process follows everything, such as the hashing discussed in the integrity section, with the addition of using asymmetric encryption. The sender uses the private key to sign the document, and the receiver uses the public key to verify the signature. Because the private key is supposed to be known only by the sender, if it does verify with the public key, the sender cannot claim they did not send it. This is illustrated in Fig. 4.

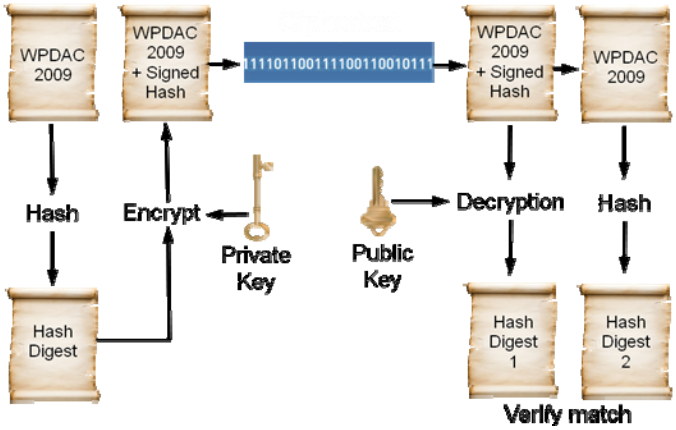

Fig. 4. Digital Signature

D. *Cost of Cryptography*

There is a technical cost for using cryptography. This cost should be carefully analyzed before applying cryptography to the control system. We can break this cost into two groups, a computational cost and a communications cost.

Computational cost is the extra burden cryptography has on the microprocessor in the device. Cryptography is based on complicated mathematics, such as factoring large prime numbers, and to do this uses a considerable amount of processor resources. One solution is to have a cryptographic card or plug-in to the product to offload the computational burden to another processor. Another important factor in a successful cryptographic implementation is to have a good source of random data. These random data are used to generate session keys or to pad data blocks. The operational burden will be how long it takes to produce a large enough block of sufficiently random data. Some PC solutions require the user to wiggle the mouse or to type random keys on the keyboard. A better solution that will not need human interaction is to have a hardware random-number generator. Another computational cost is in the architecture of the cryptography.

Communications cost is the extra bandwidth the cryptographic solution requires and the extra latency of the messages. With any cryptographic solution, more data are communicated for each message. This comes in the form of additional headers, trailers, key exchanges, synchronization frames, or any additional checks, such as known value challenges. This additional information keeps both ends of the communication synchronized on where the messages start and stop and what values should be used to understand the next message.

Encryption and authentication have different requirements. For the most part, authentication requires more communications cost than encryption does, because the message must be authenticated before being processed. Encryption, in most cases, can perform much of the computational activities before the message arrives. Then, as the message arrives, it can be decrypted while being received, adding very little latency. Contrast this with the holdback, a term used to describe the

process of needing to collect multiple bits in a serial communications link before being able to perform cryptographic functions on them. This is done if the cryptographic function needs blocks of data or if the data are being authenticated. If the data are being authenticated, all the data in a message must be collected. The cryptographic functions run on the message; only then (after authentication) can the message be passed on to be processed or used. The holdback extends the latency from the time the message is generated to the time it is acted on by many factors. For example, commonly used hash functions generate 160-bit hash. If we are protecting a DNP3 message that is 64 bits, the DNP3 and hashed message sent is 224 bits. We have added 71 percent overhead. In many existing protection or SCADA communications channels, there is not enough available channel bandwidth to accommodate this much overhead. Fortunately, hashes can be truncated, such as appending only 120 bits instead of 160 bits. This truncation reduces the bandwidth requirements, at the cost of less security for detecting a digest collision.

E. Examples of Cryptographic Impact

Knowing that a solution claims 256-bit Advanced Encryption Standard (AES) encryption does not ensure it will protect the information correctly. As discussed, there are many considerations that assist in selecting the appropriate tool.

Looking at two serial cryptographic protocols, we can see the very different impacts cryptography can have on control system communication. These two protocols bring powerful security to the communications and are fulfilling different business goals.

The design objectives for Protocol 1 are the following:

- Minimal latency
- Minimal cryptographic overhead
- Defense against injection, modification, splicing, replay, man-in-the-middle, forging, and reordering
- Confidentiality
- NIST-approved cryptography

This protocol provides data confidentiality and session authentication but not individual message authentication. A cryptographic frame consists of a header and the encrypted data. The header is 7 bytes and consists of start-of-message characters, a counter, and other data to keep the encryption/decryption process synchronized. The data length of the message, protected by the header, is variable, allowing this protocol to match many control system protocols. Matching frame structure minimizes delays that could arise between data frame size and encryption frame size that would require padding. Typically, the data length is set to the maximum message length of the protocol, for example, 100 bytes for Modbus. When the first byte of data to be protected is received by the encryption device, the frame header is sent, followed by the encrypted byte. Thereafter, each plaintext byte is encrypted and sent on a byte-per-byte basis until the frame data length is reached or the message is over. When the frame length is exceeded, a new frame header with an incremented counter is created, and the process repeats. Fig. 5 shows the structure of the session data encryption and link header.

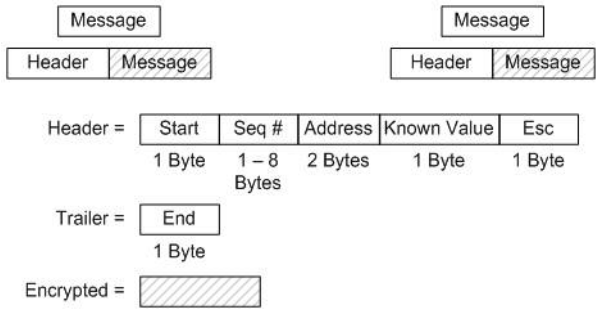

Fig. 5. Protocol Design 1 Session and Link Layer

This simple protocol is also very efficient because it does not burden the existing SCADA channel with a large amount of cryptographic communications cost. The frame header is the only additional channel burden and is relatively small, 7 bytes. After the frame header is sent, each byte received by the cryptographic protocol is encrypted and sent; thus the encryption process only incurs a 1-byte latency for the communications cost. This cryptographic protocol only costs about seven percent of overhead at 9600 bps.

The design objectives for Protocol 2 are the following:

- Message integrity protection
- Defense against injection, modification, splicing, replay, man-in-the-middle, and reordering
- Authentication
- Confidentiality
- NIST-approved cryptography

This protocol considers message integrity and authentication the most important design goals, more so than confidentiality. This means that it prioritizes authenticating every message over encrypting every message. The reason is that in real-time protection or SCADA communications, most messages are not secret; for instance, an open breaker in a SCADA system is known and expected. However, it is extremely important to ensure that the message a remote device receives instructing it to open the breaker is from an authorized and trusted source. Fig. 6 shows the session and link layer.

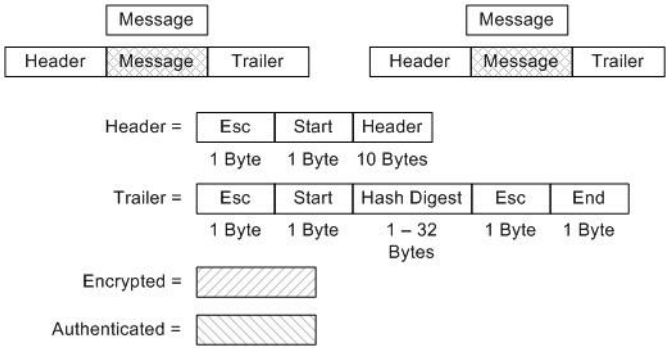

Fig. 6. Protocol Design 2 Session Transport and Link Layer

This protocol has full message holdback, where the sending and receiving sides must have an entire frame before performing cryptographic functions. The advantage to this mode is that each message is authenticated. The disadvantage is latency.

In serial communications, either the polling rate must be increased or the number of polled devices must be decreased if the cryptographic overhead is too much for the communications channel.

Note that encryption and decryption do not add any additional overhead. Removing encryption allows us to troubleshoot the communications channel, but it does not lower the additional bandwidth requirements and cryptographic overhead.

The test system shown in Fig. 7 was developed to validate the actual impact of these two cryptographic protocols on SCADA traffic when the protocols were added to the system in a bump-in-the-wire architecture. This simulates cryptographic solutions being applied to existing control systems where the primary equipment is not upgraded.

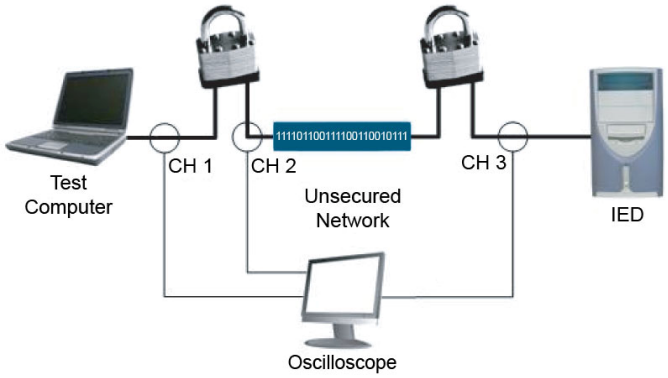

Fig. 7. Test Configuration

SCADA traffic was generated on a test computer acting as a SCADA master. A single DNP3 outstation was configured. Two serial encryption devices, in turn, were placed between the SCADA master and the SCADA outstation. These devices were configured for a data rate of 9600 bps, 8 data bits, no parity, and 1 stop bit. Protocol 1 was configured to use AES-128 encryption. Protocol 2 was configured for AES-128 encryption and Hash Message Authentication Code-Secure Hash Algorithm1 (HMAC-SHA1) with a 128-bit key.

Three points within the communications systems were tapped and routed to an oscilloscope to measure the relative timing between the transmission of frames:

1. CH1: between the SCADA master and the first cryptographic device.
2. CH2: between the cryptographic devices.
3. CH3: between the second cryptographic device and the SCADA outstation.

A DNP3 "read binary inputs" command was used to elicit a response from the outstation device. The resulting timing diagrams show the effect the two cryptographic protocols had on the communications.

F. Protocol 1 Data Latency

Fig. 8 shows the transmission latency of a "read binary inputs" request introduced by Protocol 1 on a DNP3 frame. CH1 is at the top, CH2 in the middle, and CH3 at the bottom. The additional latency added by the cryptographic protocol is 7 milliseconds.

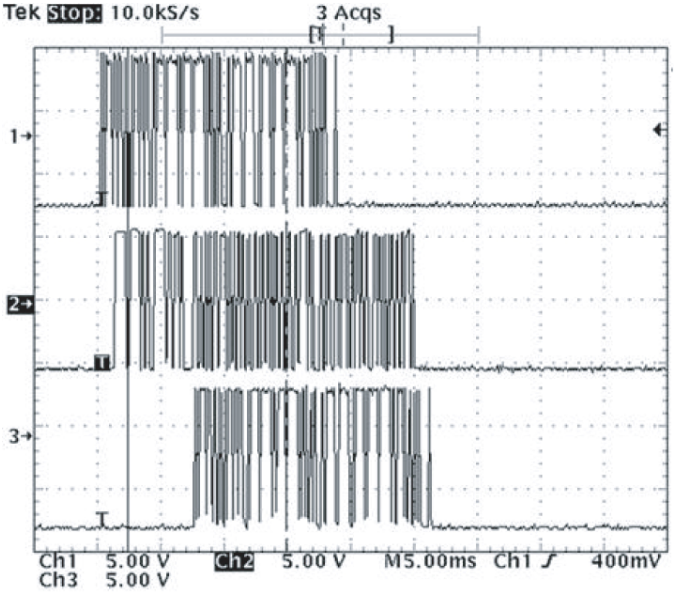

Fig. 8. Protocol 1 Mode DNP3 Latency

G. Protocol 2 Data Latency

Fig. 9 shows the transmission latency of a "read binary inputs" request introduced by Protocol 2. CH1 is at the top, CH2 in the middle, and CH3 at the bottom. The additional latency added by the cryptographic protocol is 59 milliseconds.

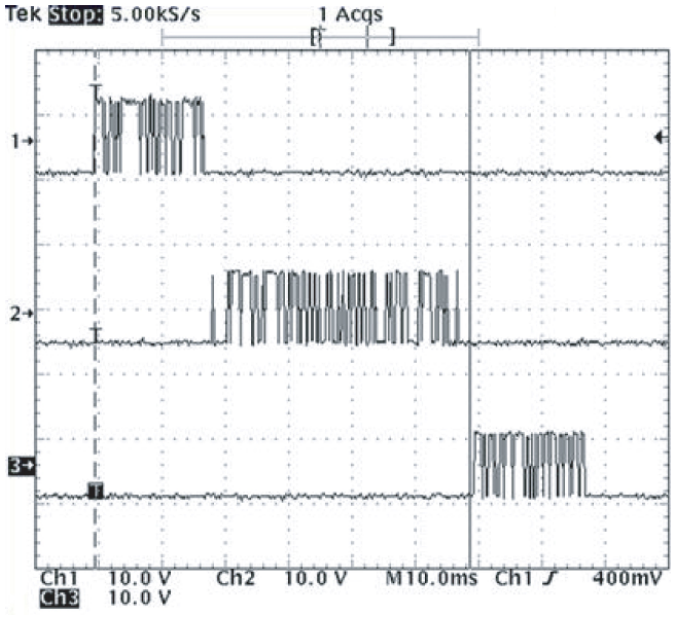

Fig. 9. Protocol 2 DNP3 Latency

Full holdback is shown in Protocol 2. The initial delay is because the protocol works on blocks of 16 bytes; thus no output is generated until at least 16 bytes of data are received. Protocol 2 performs cryptographic authentication before transmitting the results, which creates output latency from the second cryptographic device. This latency is a function of the frame size and overhead introduced by the transport and link layers.

Both cryptographic protocols achieved the goals they set out to accomplish but impacted the control system in different ways. The key is to balance the business objectives with the technical solutions.

IV. Operational Impact

We have talked in-depth about the technical cost of cryptographic tools. This is intended to provide guidance on how to select the correct tool for the job. The next consideration must be the operational costs. This encompasses what is needed for ongoing tasks to keep the cryptography running correctly. This is answered by investigating scalability and maintainability. Scalability refers to the readiness of a system to grow or shrink in an efficient manner. Maintainability focuses on what needs to be done to make sure the cryptographic technology continues to be used in compliance with policies and procedures after the initial rollout.

The first step in investigating maintainability is to understand the operational needs. In a corporate networking environment, confidentiality usually holds the highest priority. Engineering access communication in control system confidentiality is important, whereas in SCADA communication, authentication and integrity hold a higher priority. The policies and procedural requirements of the organization must be understood in order to select the appropriate cryptography. Once the requirements have been identified and the technology selected, the next step is to test, set initial configurations, and plan deployment. An understanding of cryptography is necessary for analyzing and testing proposed solutions. Accredited third-party validation processes, such as the Federal Information Processing Standards (FIPS), can be leveraged to provide a level of assurance. In testing, focus is on analyzing whether the cryptographic technology meets the policy objectives. Make sure the type of cryptography matches the type of communication it is applied to. For example, if we apply cryptography that only authenticates to an engineering access communications channel, the cryptographic solution falls short. Confidentiality should also be applied to protect passwords and settings being communicated across this channel. If using encryption on SCADA data, do not use ECB, because it is vulnerable to replay or man-in-the-middle attacks.

The second half of maintainability covers all the additional requirements of operations to update and support the technology after deployment. These requirements include account and key management controls, event and log retrievals, updates and patch management, and periodic validation testing. This cost depends on the type of features the product includes. Using a product that has central authentication or includes software to do account updates and creation from a central location to all installed units will be faster than having to physically visit each installed unit. However, this type of system requires out-of-band communications to all installed devices. In-band messaging can help, but the tradeoff is the use of bandwidth. In-band messaging uses the existing communications channel to pass management data and stops the operational data temporarily. Depending on the infrastructure capabilities, this centralized management structure can save time and money. If the technical infrastructure does not support this larger scale communication, preshared configurations will have to be accounted for.

Scalability costs occur when the control system demands change and the system grows or reduces in size. Suddenly, new trust relationships need to be established. Cryptography is about shared secrets and the trust that a secret has not been compromised. The cost of scalability is how much it will cost the organization to establish the new system trust, as well as the cost of additional deployment.

V. Recommendations

Cryptographic solutions can and should be applied to control systems to provide security, confidence in data integrity, confidentiality of sensitive information, and authentication and authorization of the operator. As has been shown, correctly chosen cryptography will secure even bandwidth-limited serial channels.

Cryptography enables secure and efficient control system communications no matter what type is used, SCADA, real-time protection, or engineering access. The solutions, on the other hand, will be different for each type. Encryption is very important for engineering access; authentication is important for real-time and SCADA. Cryptography lowers enterprise risk when selected and applied correctly. It also allows an organization to control who has access to information and how it is seen. Correct selection is only possible through clear identification of business needs, solid understanding of day-to-day procedures, and analysis of available cryptographic solutions. Cryptography enables more interproduct communications, centralized data collection, and remote access, which result in increased workforce efficiency.

There is a gradual transition that can be followed to upgrade an existing noncryptographic infrastructure to a cryptographic infrastructure:

1. Secure legacy serial lines with bump-in-the-wire cryptographic technology, allowing operations to continue to use dial-up access for engineering access, leased line or radio modems for SCADA, and fiber for real-time protection.
2. Secure legacy Ethernet with security gateways using cryptographic tunneling technology.
3. Select next generation products that integrate technology (when appropriate) that interoperates with the gateway cryptographic protocols.

VI. Further Reading

D. Whitehead and R. Smith, "Cryptography: A Tutorial for Power Engineers," proceedings of the 35th Annual Western Protective Relay Conference, Spokane, WA, October 2008.

VII. Biography

Rhett Smith is a development manager for the security solutions group at Schweitzer Engineering Laboratories, Inc.. In 2000, he received his BS degree in electronics engineering technology, graduating with honors. Rhett is working on two DOE control system security cooperative agreements. He is the project director for the Hallmark project and is one of the principal investigators on the Lemnos project. Rhett has his GSEC, GIAC Security Essentials Certification, and is a Certified Information Systems Security Professional (CISSP).

© 2009 by Schweitzer Engineering Laboratories, Inc.
All rights reserved.
20090226 • TP6366-01