

# SKMA – A Key Management Architecture for SCADA Systems

Robert Dawson    Colin Boyd    Ed Dawson    Juan Manuel González Nieto

Information Security Institute  
Queensland University of Technology,  
GPO Box 2434, Brisbane, QLD 4001, Australia  
Email: {re.dawson,c.boyd,e.dawson,j.gonzaleznieto}@qut.edu.au

## Abstract

Supervisory Control And Data Acquisition (SCADA) systems are widely used in the management of critical infrastructure such as electricity and water distribution systems. Currently there is little understanding of how to best protect SCADA systems from malicious attacks. We review the constraints and requirements for SCADA security and propose a suitable architecture (SKMA) for secure SCADA communications. The architecture includes a proposed key management protocol (SKMP). We compare the architecture with a previous proposal from Sandia Labs.

*Keywords:* SCADA Security, Key management, Secure protocol, Key Distribution Center (KDC), Key establishment protocols

## 1 Introduction

Nations are becoming increasingly dependent on automated Supervisory Control And Data Acquisition (SCADA) systems to help deliver critical services such as water, sewerage and electricity distribution. SCADA systems, which once used proprietary communication mechanisms, are increasingly using standard protocols, such as DNP3 (Curtis 2005).

The use of standard protocols, combined with increased interconnectivity with other networks, has changed the threat environment. In 2001 the British Columbia Institute of Technology (BCIT) began recording information about world-wide industrial security incidents (Byres & Lowe 2004), storing this information in a database, similar to the CERT computer security incident database. CERT began capturing computer security incident data in 1988, and has seen the number of incidents rise from six in 1988 to 137,529 in 2003 (*CERT/CC Statistics 1988-2005* 2005). While the current rate of incidents being added to the BCIT database is currently low, it is also increasing. The increase in incidents reported, and the changing nature of the sources, indicate that the risk of SCADA incidents occurring is increasing.

The need to secure SCADA systems has therefore been identified as an important field of research. One critical security requirement for SCADA systems is that communication channels need to be secured. Secure keys need to be established before cryptographic techniques can be used to secure communications.

---

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at the Fourth Australasian Information Security Workshop (AISW-NetSec 2006), Hobart, Australia. Conferences in Research and Practice in Information Technology, Vol. 54. Rajkumar Buyya, Tianchi Ma, Rei Safavi-Naini, Chris Steketee and Willy Susilo, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

## 1.1 Relationship to Existing Work

Communications security for SCADA is a topic that is being addressed in both the academic community and in industry. Wang & Chu (2004) have developed broadcast and point-to-point protocols, based on earlier work in Sensor Networks (Perrig, Szewczyk, Tygar, Wen & Culler 2002). In industry, the American Gas Association is developing a standard for secure communication (AGA 12-1 Working Group 2005) that is based on link-level encryption. Although these protocols use cryptographic techniques to protect the confidentiality and integrity of data, they do not directly address key establishment.

Cryptographic protocols depend on having secure keys distributed to the parties participating in the protocol. A cryptographic key needs to be established before messages can be encrypted and sent between parties.

Researchers at Sandia have produced a paper on key establishment for SCADA (SKE) (Beaver, Gallup, Neumann & Torgerson 2002). Their paper firstly outlines SCADA security systems architecture, and then discusses a key management solution. However the key management design that Beaver et al have proposed, has the following limitations:

1. Both symmetric and public key cryptography techniques are used.
2. Long term keys are shared between nodes via manual installation. If a Remote Telemetry Unit (RTU) has multiple master stations, its key will need to be installed on each master station. Also, if a master station is compromised, long term keys are also compromised.

The SCADA Key Management Architecture (SKMA) proposed in section 5 of this paper has the following advantages over SKE:

1. SKMA only uses symmetric techniques; thus simplifying implementation, and minimising overheads.
2. SKMA only requires that long term keys to be stored on the node to which the key belongs, and one other party, the Key Distribution Center (KDC). This decreases the number of copies of each long term key, minimising the risk of exposure, and simplifying recovery from the compromise of a master station.

## 1.2 Contribution

This paper provides a concise description of the constraints of a SCADA system with respect to secure communication (Table 1). Key management requirements for SCADA systems are also outlined (Table 2).

The most important contribution of this paper is the key management mechanism (tailored specifically for SCADA systems) proposed (Section 5). This contribution consists of the SCADA Key Management Architecture (SKMA), and the SCADA Key Management Protocols (SKMP).

SKMA is an architecture that provides security meeting the constraints and requirements in Sections 2 and 3. SKMA specifies the keys and mechanisms required to secure SCADA communications.

SKMP uses a series of existing security techniques to provide secure key management. ISO 11770-2 mechanism 9 (ISO 1996) is used to establish a long term key shared between the nodes. An approach for deriving session keys is suggested, and a technique for key revocation is described.

## 2 SCADA Architecture

A SCADA system consists of a number of different entities communicating with each other. These entities are diverse in purpose and design, varying from a Remote Telemetry Unit (RTU) that interacts with the physical environment, to the Human Machine Interface (HMI) that operators interact with. In this paper, the term *node* will be used to refer to any entity in the system.

The entities that make up a SCADA system are shown in Figure 1. The boxes with dashed lines in Figure 1 indicate parts of the system that should be physically secured. The box at the top left indicates the main network. The entities in the system, and the communication channels between entities are described in more detail below.

### 2.1 Remote Telemetry Unit

RTUs are devices composed of a microprocessor that controls sensors and actuators that interact with the physical environment.

For example, in a water control system a typical RTU would consist of:

1. one or more water pumps (actuators);
2. sensors that measure the water level; and
3. a microprocessor that takes input from the water level sensors, and sends commands to control the pumps.

RTUs are able to communicate with other entities in the network. This communication is two-way, with the RTUs typically allowing settings to be changed and commands to be sent to the sensors or actuators of an RTU.

RTUs have limited memory and processing power. There are RTUs running industry standard protocols on 16 bit Microprocessors with 8 kilobytes of RAM (working memory), and 64 kilobytes of EPROM (persistent memory).

An RTU can often be located remotely to the main corporate offices. The location of RTUs may make physically securing the units difficult. For example, sewage pumps need to be located throughout residential areas, in locations where extensive physical security is not practical.

### 2.2 Master Stations

The master station is a node which provides supervisory control of an RTU. The master station is the superior in a communication hierarchy (IEEE Standards Board 1994).

The structure of a SCADA system will normally include one central master station, which communicates with a hierarchy of other nodes, including sub-master stations, and RTUs.

Master stations and sub-master stations, are computers with resources at least as plentiful as a modern desktop computer. These machines typically run on commodity (standard) hardware and operating systems.

### 2.3 Human Machine Interface (HMI)

The HMI is the device that people use to interact with a SCADA system. HMIs for SCADA systems have been developed utilising a wide range of client technologies, including PDAs, web browsers, and Desktop PCs (Iconics 2005).

### 2.4 Historian

The historian is a database of the historical data from the SCADA system. It is updated by the master station, and can be accessed from the HMI. The Historian runs on similar hardware to the master station.

### 2.5 Communication Channels

The network topology of a SCADA system is highly structured. The available communication paths between nodes are known in advance. In a SCADA system there is no need to support ad hoc communication between nodes. Nodes are added in a managed fashion. A detailed description of the communication paths in a SCADA system is outlined below.

#### 2.5.1 Master-RTU Communication

Figure 1 shows many of the diverse options used for the master-RTU channel. The communication can take place using diverse mechanisms such as:

1. the Internet
2. satellite
3. radio
4. physical cables
5. WiFi
6. standard modem/ethernet

As malicious messages on the master-RTU channel could lead to physical damage, it is critical that this channel is secured. The physical remoteness of the RTUs limits the possibility of physical security. A technical solution, including the use of cryptographic mechanisms is required. This technical solution will include a system for managing cryptographic keys.

The mechanisms listed above include a number of channels where the message travels via wireless signals. In SCADA systems that use these channels, messages can be marked for a single node, or can be sent to all nodes.

#### 2.5.2 RTU-RTU Communication

RTU-RTU communication is possible, and occurs in a controlled manner. However, not all RTUs will communicate with other RTUs. There are a number of scenarios where RTU-RTU communication is required. These scenarios can all be planned for in advance. Any security solution designed for master-RTU communication should also support RTU-RTU communication. In situations where an RTU is able to control another RTU, the RTU that acts as a master should be physically secured.

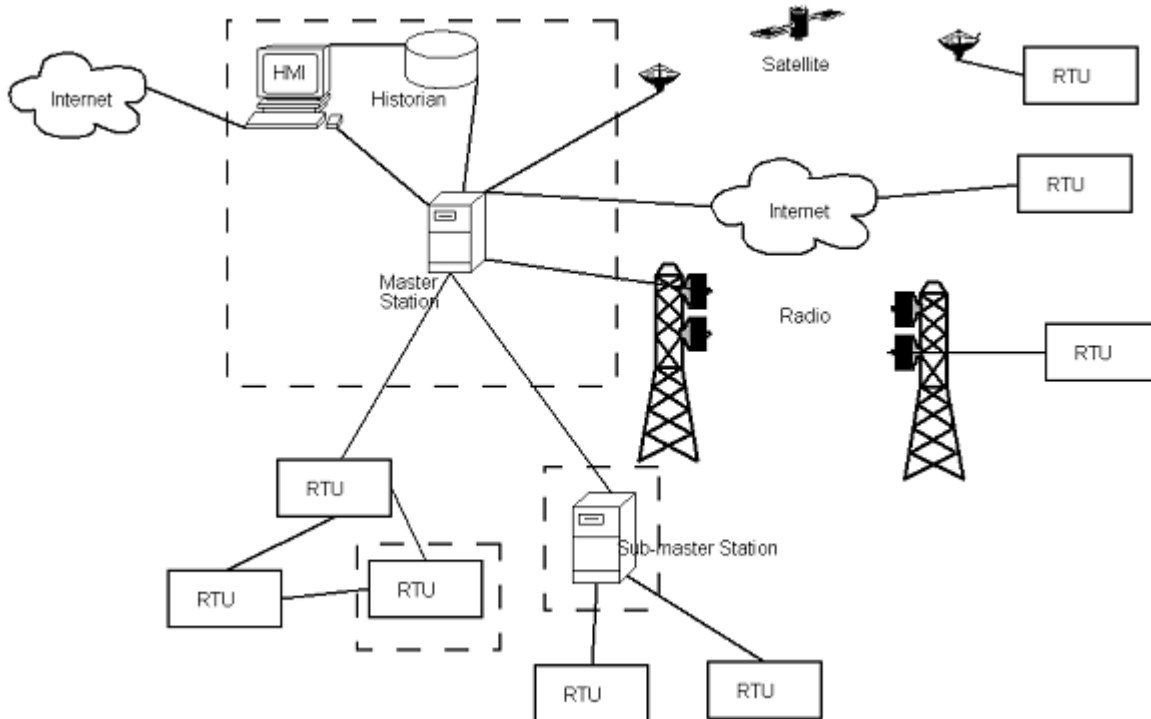


Figure 1: SCADA Architecture

### 2.5.3 Other Communication

Other communication channels include:

**HMI-Master Communication:** The HMI is able to communicate with the master station. This communication is typically run using TCP/IP based protocols, and utilises a client server architecture.

**HMI-Historian Communication:** The HMI-Historian communication is similar to that of the master-HMI channel.

## 2.6 Key SCADA Architectural Constraints and Requirements

Table 1 outlines the key elements of the SCADA architecture that impact the design of a security architecture for SCADA systems. These elements are referred to as (C1) to (C10) in the remainder of the paper. The SCADA architecture outlined above does not match that of popular computer networks. The differences have an impact on the security requirements, therefore a specific solution for SCADA security is needed. As the HMI-Master architecture utilises a standard client-server architecture, standard security solutions can be applied to this part of the SCADA system.

Of particular note are the requirements relating to performance. The RTU has low resources, and many of the communication mechanisms used have low bandwidth. In addition many of the processes that are controlled by SCADA systems need to be monitored and controlled in real time. Most nodes in the system will only communicate with a small number of other nodes. Many SCADA systems are always on, and have been designed to be failproof. The physical location of a RTU is dictated by the physical environment the RTU needs to interact with. This makes it difficult to apply physical security mechanisms to the device.

RTUs have a long life time and are designed to last for at least ten years. Many RTUs have been deployed

for up to twenty-five years. This in combination with the dispersed physical structure of SCADA systems means that rolling out changes to the RTUs will take a long time.

When an RTU is initially added to the network, its clock should not be trusted, as there will have been no way of synchronising it with the master station clock. After the RTU has been installed, it will have its clock synchronised, in order to support timestamping of messages.

## 3 Security Requirements

When looking at the security of a system, the requirements can be classified in terms of:

**Confidentiality:** limiting access to information or resources to those people.

**Integrity:** ensuring that the data has not been changed (data integrity), and the origin has not been changed (origin integrity or authentication). This also includes user authentication.

**Availability:** the ability to use the information or resource desired. (Bishop 2002)

In a SCADA system, these can be prioritised, with integrity of messages being the highest concern, followed by availability, and then confidentiality. The rationale for this prioritisation is seen below.

### 3.1 Integrity

It is critical that messages between nodes are not tampered with, and that no new messages are inserted (data integrity). A malicious attacker could cause physical damage if they have the ability to alter or create messages. It is also important that the messages are authenticated, allowing confidence in the source of messages, and also preventing attackers from inserting messages.

User authentication should be performed using techniques familiar to standard client-server applications.

<b>ID</b>	<b>Constraint/Feature</b>	<b>Description</b>
C1	Resource Constrained RTU	RTUs have low processing power as well as limited persistent and working memory.
C2	High Resiliency	Due to their interaction with the physical world, SCADA systems have been designed to be always on, without any downtime. Any change to this should be minimal.
C3	Low Bandwidth and Low Latency Communications	Bandwidth is limited to 9600 baud on many systems, such as those that use satellites to communicate with remote devices
C4	Long Node Life	Nodes will typically last for up to 25 years, much longer than the life spans of typical computer hardware components.
C5	Real Time	The physical processes controlled by a SCADA system often need to be interacted with in a real time manner. This constraint is not constant across all SCADA systems.
C6	Structured Network	The structure of the network and its communication channels will be well defined. Ad hoc communication between nodes are not required.
C7	Phased Delivery	Due to the size of the systems, the real time properties, and long life span of RTUs, a phased rollout of communication security is required, perhaps running over a number of years, while legacy hardware that cannot support the security is upgraded.
C8	RTUs Physically Insecure	As RTUs are deployed to remote locations, they cannot always be physically secured.
C9	RTU Clocks Initially Unsyncronised	When initially installed the clock of an RTU cannot be relied on.
C10	RTU Clocks Synchronised After Initialisation	Once the system has been initialised, the SCADA system will ensure clocks are synchronised.

Table 1: SCADA system constraints.

### 3.2 Availability

Many SCADA systems need to be available for use at all times, as outages can cause physical damage or threaten human life. Countermeasures designed to provide improved integrity or confidentiality need to be implemented in such a way that the availability of the system is not decreased. The proposed cryptographic solution should not require messages outside of the initialisation of the system, and dependence on new messages should be kept to a minimum.

### 3.3 Confidentiality

Confidentiality is a much lower priority for most SCADA systems. Support for confidentiality is important, but will not be used in some environments. Systems that need to respond instantaneously (C5) and those that contain resource constrained RTUs (C1) may not be able to afford the extra processing overheads associated with providing confidentiality services.

### 3.4 Key Establishment Requirements

Having a set of well defined requirements is critically important for a key establishment system. This facilitates thorough analysis of the system, including the use of provable security techniques.

The key management requirements are outlined in Table 2. In this table, A and B refer to the entities exchanging keys (master stations and RTUs in SCADA). The definitions of goals are based on those given by Boyd & Mathuria (2003).

In listing these requirements, there is some redundancy. Any protocol that provides mutual key authentication will also provide mutual entity authentication. Similarly, a protocol that provides key confirmation will also ensure key integrity.

## 4 Proposed Architecture for Secure Communication

The architecture outlined below focuses on key management for point-to-point communication. The keys being managed are ones that would be suitable for the AGA 12-1 standard (AGA 12-1 Working Group 2005). Many of the SCADA systems require broadcast communication. Secure broadcast (as opposed to point-to-point) message transfer is not within the scope of this paper. Sending messages over insecure channels (such as the Internet, and radio) to a specific node is supported. The  $\mu$ TESLA protocol, tailored for sensor networks may be a suitable broadcast option (Perrig et al. 2002).

Figure 2 depicts a SCADA system, including an additional entity, the Key Distribution Center (KDC). Nodes that do not form a part of the new security architecture are excluded.

### 4.1 Design Goals

The goals of the security architecture are driven by the combination of security requirements and the constraints inherent in SCADA systems. These factors combine to form a unique environment, leading to the design goals outlined below.

As the system has strict performance constraints (C1, C3 and C5), only symmetric cryptographic techniques will be used. Due to the network structure, which does not require support for ad-hoc node-to-node communication (C6), this solution will not produce the problem of key explosion that is usually found with symmetric systems.

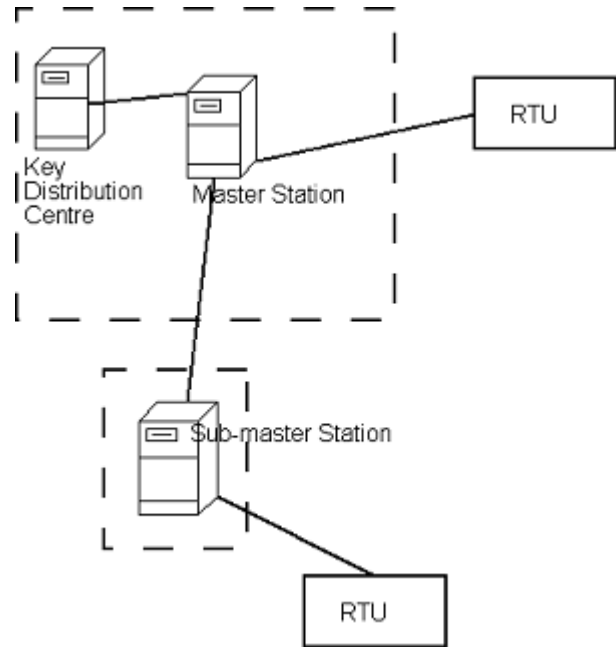


Figure 2: Key Management Architecture

It is critical that the dependence on nodes such as the KDC is kept to minimum after the successful deployment of a node in order to maintain the resiliency of the system (C2). This requirement is met by limiting the dependence on the KDC to the establishment of a node-node key (see section 5.2), and deriving session keys, rather than communicating them.

As the nodes are physically insecure (C8), the security system needs to ensure that the compromise of a node has limited impact. Compromising a node should not compromise all communication.

As a security system cannot be instigated across all nodes simultaneously (C4 and C7), a node needs to be able to communicate with other nodes that are known to be insecure, without using the security system.

### 4.2 Key Distribution Center

The KDC will be used to maintain a long term key for each node in the system. The KDC will also contain information regarding the system structure, and will be responsible for allowing and denying key establishment requests. In performing this role, it will be facilitating the distribution of keys, and the initialisation of trust relationships between nodes. In addition, key revocation messages will be issued by the KDC.

The KDC should be co-located with the master station of the SCADA system, which means that messages between the KDC and master station will be efficient. In addition physical security will be high, as the master station has a requirement of being physically secured. If possible, the KDC should be implemented using a secure hardware device.

### 4.3 Trust Relationships

In general RTUs will be deployed into untrusted locations, which leads to the assumption that the RTUs are untrusted. The proposed key management system allows for a node to be compromised without compromising the entire system.

The master station will be trusted by RTUs. In the case of an environment that includes sub-master stations, each master station (or sub-master station) will be trusted by the nodes that are subordinate to

Requirement	Description	Importance
Mutual Entity Authentication	Both entities (A and B) involved in the protocol will have a fresh assurance that the other entity participated in the protocol.	It is critical in establishing the key that both the master station and RTU are assured of the existence of each other.
Key Freshness	Both entities are assured that the key is fresh (i.e. it has just been created).	Key freshness is required to prevent adversaries from reusing a revoked key.
Key Authentication	The key is only known by the parties involved in the protocol.	This requirement means that the key is not known to an adversary.
Mutual Key Confirmation	A and B have assurance that the key has successfully been established, and is ready for use.	It is critical that both nodes are assured that the key has been successfully communicated, and is available for use.

Table 2: Proposed Key Establishment Requirements

it. Each master station should be physically secured, because an attacker would be able to control any of its children if the master station is compromised.

All nodes will trust the KDC. Each node will have a key that it shares with the KDC. The trust relationships between a master station and its child RTUs will be initiated via the KDC.

An RTU may be configured to act as a backup substation to other RTUs. In this case it will be treated as a sub-master station, with the same requirements for physical security. The trust relationship between the backup master RTU, and other RTUs will be initiated through the KDC.

As communication is performed using wireless signals (such as radio, WiFi and satellite), the channels need to be treated as insecure. It is trivial for an adversary to insert, modify or delete messages from these channels. The only guarantee the system has is that messages are delivered some of the time.

#### 4.4 Secure Communication Channels

The architecture proposed will make use of a series of keys, with different uses. The keys are outlined below:

**long term node-KDC key:** This key will be shared between a node and the KDC, and will be used when establishing keys used for communication.

**long term node-node key:** Nodes that need to communicate with each other will share a key that is established using the mechanism in section 5.2.

**session key:** Underlying encryption mechanisms may or may not recommend the use of a session key, used for encrypting messages.

**broadcast keys:** The broadcast mechanism (not specified in this document), will require independent keys.

#### 4.5 Use of Keys

The keys that are generated will be used to communicate messages. As mentioned in section 4.1, these messages need to be transmitted efficiently. As discussed in section 3, all communication requires message and source integrity, and where possible confidentiality should be provided.

In environments where confidentiality produces excessive overheads, a Message Authentication Code (MAC), using the node-node (or session) key should be used. Where possible, confidentiality should be provided. The confidentiality and integrity services can then be provided using the encrypt and MAC approach (i.e. encrypt the message and then produce a MAC), or by using efficient authenticated encryption modes of operation such as those proposed by NIST (2003).

### 5 Proposed Key Management Mechanism (SKMP)

Each of the keys outlined in section 4.4 will need to be managed. In order to implement SKMP, processes for each of the following need to be implemented:

1. Installing the node-KDC key on a node and the KDC before a node is deployed to the system;
2. Exchanging the node-node key when installing a new node;
3. Generating a session key that will be used for more direct communication;
4. As the long term keys will not expire, there needs to be a mechanism for revoking these keys; and
5. The KDC needs to be able to notify nodes that a node does not have security deployed, and requires unsecure messages to be sent and received to it.

#### 5.1 Node-KDC Key

Each node in the proposed system will have a key which it shares with a key distribution center (KDC). Both the KDC and the node will need to keep this key secret.

When adding a new node to the system, a node-KDC key will be configured and securely stored on the node and KDC. The key will be installed on the machines using a manual process.

The node-KDC key will be used to send node-node key establishment messages between nodes and the KDC, as described in section 5.2. As this key is only known by the KDC and the node which the key protects, the risk of exposure is limited.

## 5.2 Node-Node Key Establishment

The node-node key establishment protocol will be a three party key establishment protocol that uses the KDC as a server. The node-KDC keys will be used to communicate with the server when running this protocol. The first documented three party key establishment protocol was developed in 1978 (Needham & Schroeder 1978). The Needham-Schroeder protocol establishes a key in a series of four messages with the use of a trusted third party. Unfortunately this protocol does not meet the requirements specified in Table 2. The primary failing of the Needham-Schroeder protocol is that it does not provide key freshness, allowing adversaries to reuse earlier keys, using variants of an attack first proposed by Denning & Sacco (1981). Newer protocols based on the Needham-Schroeder protocol use one of three strategies for ensuring that the keys are fresh.

### 5.2.1 Time Based Freshness

Protocols have been proposed that include a time-stamp in the messages sent. An example of a key establishment protocol that provides these properties is the Kerberos Authentication Mechanism (Neuman & Ts'o 1994). It is possible to meet the security requirements of Table 2 using this approach with only four messages. However synchronised clocks are required for this to happen.

It is not possible to have synchronised clocks when the Node-node keys are being established. Since the key initialisation is performed at the installation of the RTU, there is no way to ensure that the clocks are synchronised without sending messages through the network to confirm. While efficient techniques for this exist, time synchronisation needs to be performed securely and requires additional communication, which means that in real terms, more than four messages need to be sent.

### 5.2.2 State Based Freshness

Another approach that is not dependent on time depends on the use of a time variant parameter (TVP), the state of which is maintained by all nodes in the system. This approach can be implemented without any dependence on clock synchronisation. Numerous different techniques using this approach have been proposed. One approach that minimises the storage requirements on clients and servers is to use the clock of a trusted server to generate the sequence number (Mitchell 2000). ISO 11770-2 Mechanism 8 (ISO 1996) depends on a TVP.

In the state based options, nodes are required to maintain state so that old messages are identified. This means that nodes will need to maintain two variables, one which is their long term key, the second being the state. This is not suitable for the extremely resource limited SCADA systems.

### 5.2.3 Nonce Freshness

Instead of depending on time, nonces<sup>1</sup> are generated by the nodes establishing the key. Messages containing the key material also include the nonce which the user maintains. The user then confirms the message. ISO 11770-2 Mechanism 9 (ISO 1996) uses nonces, as well as other independently developed protocols (Carlsen 1994, Bauer, Berson & Feiertag 1983, Bellare & Rogaway 1995). These protocols all run in a series of five messages, and meet the requirements outlined in Table 2.

<sup>1</sup>A nonce is a number used once.

1.  $B \rightarrow A : N_B$
2.  $A \rightarrow S : N_A, N_B, B$
3.  $S \rightarrow A : \{N_A, K_{AB}, B, Text1\}_{K_{AS}}$   
 $\{N_B, K_{AB}, A, Text2\}_{K_{BS}}$
4.  $A \rightarrow B : \{N_B, K_{AB}, A, Text2\}_{K_{BS}}$   
 $\{N'_A, N_B, B, Text3\}_{K_{AB}}$
5.  $B \rightarrow A : \{N_B, N'_A, Text4\}_{K_{AB}}$

Figure 3: ISO 11770-2 Mechanism 9

Figure 3 describes the ISO 11770-2 mechanism 9 protocol.

This figure uses the following conventions:

- $N_A$  is a nonce generated by node A.
- $N'_A$  is a second nonce generated by node A.
- $S$  is the server (representing the KDC)
- $A$  and  $B$  are nodes that need to establish a key
- $A \rightarrow B$  represents a message sent from  $A$  to  $B$
- $K_{AB}$  is a key shared by nodes A and B
- $\{text\}_{K_{AB}}$  is the encryption of the message  $text$ , using the key  $K_{AB}$ .

In a SCADA system, nonces are the best way to ensure freshness. ISO 11770-2 Mechanism 9 is a protocol that uses nonces to provide freshness. SKMP utilises this standard protocol, with a minor enhancement.

### 5.2.4 Modification to 11770-2 Mechanism 9

In the ISO 11770-2 protocol, there are two messages that are sent between the server and node A. In the SCADA environment, the master station and KDC will be co-located, making the master station the most efficient choice for node A. This means that the RTU will not need to directly communicate with the KDC.

In most situations, the master station and KDC are the parties responsible for initiating the key initialisation process. In order to do this, an additional message is sent from the master station or KDC to the RTU, requesting an initialisation of the protocol. This would be viewed as message 0, the structure of which is shown in Figure 4. The node  $S$  is the initiator of the message, and may be either the master station, or the KDC. Node  $A$  is the master station, and node  $B$  is the RTU. After receiving this message,  $B$  will initiate the protocol of Figure 3

0.  $S' \rightarrow B : A, B$

Figure 4: Message to initiate 11770-2 Mechanism 9

## 5.3 Session Key Derivation

In some protocols there is an important requirement of supporting session-keys. In modern stream ciphers, this requirement is often avoided through the use of nonces.

In order to improve the security against cryptanalytic attacks, and minimise the consequences of exposure of keys, session keys should be used. In order to minimise the communication overheads, these

keys will be generated using a pseudorandom function, keyed by the node-node key, and a timestamp that is based on the duration of the session.

In order to keep the code footprint of this additional operation low, the MAC function that is used in the secure communication (or authentication code when an Authenticated Encryption mode is used) can be used to create the session key.

As the volume of messages between nodes in a SCADA system is low (typically less than 1000 a day), sessions could be set to be a day, keeping the clock-synchronisation requirements to a minimum. The session key derivation will only need to be performed after the RTU has been deployed to the network. As clock synchronisation is already performed on most RTUs to facilitate timestamping of messages, this is not a new requirement.

#### 5.4 Key Revocation

Key Revocation messages will be initiated by the KDC, and should be manually initiated whenever a node or key compromise is detected. There are no known ways of automatically discovering a key compromise. The system should be monitored, and if there is suspicious behavior, a compromise may have occurred. The revocation messages should be made either with a broadcast message or sending specific messages to nodes. The key being revoked is the node-node key, which should also cause session keys to be revoked.

When sending the revocation message to specific nodes, the message should identify the specific key being revoked. This prevents adversaries from initiating denial of service attacks against the system by resending key revocation messages.

The message will be sent encrypted with the node-KDC key, using a message like that in Figure 5.

In this figure the entities will be:

- A is the entity the key is being revoked from
- S is the KDC
- $K_{AB}$  is the key being revoked.
- $K_{AS}$  is the node-KDC key shared by A and the KDC.

The message is sent from the KDC to party A, revoking the key for use between nodes A and B.

1.  $S \rightarrow A : \{K_{AB}, Text1\}_{K_{AS}}$

Figure 5: Key Revocation Message

The structure of a key revocation message sent using a broadcast message will depend on the security services offered by the broadcast mechanism used.

#### 5.5 Unsecured Nodes

With the addition of security, master stations will not allow messages from any RTU, they will only communicate with nodes that they share a key with. As it will not be possible for security to be deployed to all nodes in the system at the same time, the system will need to support unsecured communications.

In order to do this, the KDC will maintain a list of nodes which do not have security deployed. As KDC will also contain details of the network structure, it will also be able to communicate this information to the appropriate master stations.

The KDC will tell a secured master station which unsecured nodes it is able to communicate with. This

list of unsecured nodes needs to be communicated securely, sent in a message from the KDC (S) to Master station (A) using a message like that in Figure 6. The messages in Figure 6 will inform A that nodes B and C are insecure. Included in this message is a TVP to ensure the freshness of the message.

1.  $S \rightarrow A : \{B, C, TVP\}_{K_{AS}}$

Figure 6: Unsecured Node List Message

### 5.6 Summary of Proposed Mechanism

There are a number of keys that are used in the architecture, all of which are managed, without violating the security constraints of Table 1. The features of the proposed mechanism are outlined here.

**Node-KDC Key:** This key is manually installed, and is used to establish node-node keys.

**Node-Node Key:** A key establishment protocol is used for this which uses nonces as the clock of the node will be unsynchronised (C9). The protocol used is based on ISO 11770-2 Mechanism 9. The structure of the network makes the use of this protocol feasible without swamping the system with key establishment messages (C6).

**Session Key:** A technique for deriving a session key from the node key is given. This technique avoids sending messages (C2, C3) and uses the clocks which will be synchronised after the system is initialised (C10).

**Key Revocation:** A technique for revoking keys that have been compromised is given. This caters for the physically insecure nodes which make up the system (C8).

**Unsecured Nodes:** The system provides support for nodes which do not yet have security installed, in order to support legacy hardware (C4) and phased delivery of the system (C7).

The security mechanism is implemented without the use of public key cryptographic techniques due to the performance constraints of the system (C1, C3, C5).

## 6 Sandia Key Management (SKE)

The SKE is described fully in Beaver et al. (2002). The features of SKE are summarised below, and a comparison of it with SKMA is made.

### 6.1 Summary of SKE

SKE sorts communications into different classes.

- Controller-Subordinate communications
- Peer-to-Peer communications

The prime communications strategy proposed in SKE is for controller-to-subordinate messages. SKE uses symmetric key techniques for Controller-to-subordinate communication. SKE uses the following set of keys:

**Long Term Key (LTK):** There is a long term key shared between each controller and subordinate. This key is manually distributed.

**General Seed Key (GSK):** This key is stored by the controller and is used to generate a General Key. This key is generated by the Cryptographic Authority (CA).

**General Key (GK):** The GK is shared by the controller and subordinate. It is generated by the controller, using the GSK and LTK. It is transmitted from the controller encrypted by the LTK.

**Session Key:** The session key is generated using the GK, the senders id, and a TVP.

Peer-to-peer communications are used for communication between substations (sub-master stations). The peer-to-peer channels use public key cryptography for key exchange messages. The keys required for peer-to-peer communications are:

**Cryptographic Authority Public Key (CAPK):** The CAPK is shared with each substation.

**Public Key Signature Key (PKSK):** This is a key shared with all substations, the master station and the CA.

**Public Private Key Pair:** This key pair is created by the CA, and assigned to each substation.

**Common Key:** The CK is generated through a key exchange algorithm.

**Session Key:** The session key, generated in a similar way as for controller-subordinate communications.

The CK will also have a procedure for revoking certificates.

SKE forces communications between RTUs to be performed through a Substation. The substation will act as a controller, and will receive an encrypted message, decrypt it, and then re-encrypt it for the recipient (Beaver et al. 2002).

## 6.2 Comparison with SKMA

SKMA does not differentiate between master-controller and peer-to-peer communications in the same way that SKE does. A consistent approach is used for all communications, which simplifies the process. RTU communications are allowed to be directly made using the same key management process as for the rest of the system. Also, direct substation communications are possible without the use of public key cryptography techniques. SKE requires more work in installing long-term keys to nodes. In the SKMA system, long term keys are only shared by the node and the KDC. The GSK is not used in SKMA. SKMA performs a key exchange, but only when adding a new node to the system. This overhead of five messages will not impact the performance of the system. SKMA includes a definition of the key management requirements which will facilitate a formal security proof. SKE does not specify the requirements in this manner.

Table 3 outlines the main differences between the two systems. The primary differences are:

- SKMA uses a consistent approach for all communications between nodes, simplifying the implementation of the protocol;
- the management of long-term keys in SKMA simplifies the process for updating the structure of the system; and
- constraints on communications allowed are modified, relaxing limitations on node-node communication and simplifying the ways in which nodes are specified.

## 7 Conclusion

This paper has outlined the requirements for secure communication for SCADA systems. These requirements have been used to develop SKMA, a key management architecture for SCADA. SKMP, a key management protocol, was developed to implement the architecture. It was shown that the ISO 11770-2 Mechanism 9 protocol is suitable for the distribution of keys, and developed efficient mechanisms for session key generation, and key revocation. SKMP was compared with SKE (from Sandia), and the advantages of SKMP over SKE were identified.

In developing this protocol, the following areas of future related work have been identified:

**Secure Broadcast Messages:** As SCADA systems currently use broadcast communications, a suitable secure and efficient broadcast mechanism is required. Such a mechanism will enable nodes to verify the source of broadcast messages. A key management solution for the mechanism will also need to be specified.

**Formal Security Proof:** The authors are not aware of a security proof for the 11770-2 Mechanism 9 protocol. A security proof for this protocol should be developed, based on the Bellare & Rogaway (1995) protocol which is closely related to 11770-2 Mechanism 9.

**Efficient Algorithms:** The most appropriate secure and efficient algorithms that provide confidentiality and integrity need to be identified.

## 8 Acknowledgements

The authors would like to thank Suzanne Dawson for the many hours spent editing and reading the drafts of this paper. This work was performed at QUT and was supported by the Australian Research Council through grant LP0455608 in partnership with Multi-Trode.

## References

- AGA 12-1 Working Group (2005), Cryptographic protection of SCADA communications, Technical Report 12-1 Draft 5 Revision 3, American Gas Association. <http://www.gtiservices.org/security/>; accessed 14 May, 2005.
- Bauer, R. K., Berson, T. A. & Feiertag, R. J. (1983), 'A key distribution protocol using event markers', *ACM Transactions on Computer Systems* 1(3), 249–255.
- Beaver, C., Gallup, D., Neumann, W. & Torgerson, M. (2002), Key management for SCADA, Technical report, Sandia. <http://www.sandia.gov/scada/documents/013252.pdf>; accessed 5 May, 2005.
- Bellare, M. & Rogaway, P. (1995), Provably secure session key distribution – the three party case, in '27th ACM Symposium on Theory of Computing', ACM Press, pp. 57–66.
- Bishop, M. (2002), *Computer Security: Art and Science*, Addison-Wesley, Boston, USA.
- Boyd, C. & Mathuria, A. (2003), *Protocols for Authentication and Key Establishment*, Springer-Verlag.

SKE	SKMA
RTU-RTU communication is not directly allowed.	RTU-RTU communications are allowed.
Master-Controller and substation-substation communications are treated differently.	All communications are handled consistently.
Substation-substation key management uses <i>public key cryptography</i> .	Only symmetric cryptographic techniques are used.
Long term keys between Master and Controller are managed manually.	Long term keys between nodes are established with a KDC.
No Key Establishment protocol required before Master-Controller communications first take place	A four step key establishment protocol is required before Master-Controller communications take place.
No formal key establishment requirements.	Formal key establishment requirements are outlined.

Table 3: Differences between Sandia Key Management and SKMA

- Byres, E. & Lowe, J. (2004), The myths and facts behind cyber security risks for industrial control systems, in 'VDE Congress, VDE Association For Electrical, Electronic & Information Technologies', Berlin. <http://brief.weburb.dk/archive/00000135/>.
- Carlsen, U. (1994), 'Optimal privacy and authentication on a portable communications system', *ACM Operating Systems Review* **28**(3), 16–23.
- CERT/CC Statistics 1988-2005 (2005). [http://www.cert.org/stats/cert\\_stats.html](http://www.cert.org/stats/cert_stats.html).
- Curtis, K. (2005), A DNP3 protocol primer, Technical report, DNP Users Group.
- Denning, D. E. & Sacco, G. M. (1981), 'Timestamps in key distribution protocols', *Communications of the ACM* **24**(8), 533–536.
- Iconics (2005), 'Iconics MobileHMI'. [http://www.iconics-uk.com/products/pdf/mobilehmi\\_ds.pdf](http://www.iconics-uk.com/products/pdf/mobilehmi_ds.pdf); accessed 22 August, 2005.
- IEEE Standards Board (1994), Ieee standard definition, specification, and analysis of systems used for supervisory control, data acquisition, and automatic control, Technical report, IEEE. <http://ieeexplore.ieee.org/iel1/3389/10055/00478424.pdf>; accessed 5 May, 2005.
- ISO (1996), *Information Technology – Security Techniques – Key Management – Part 2: Mechanisms Using Symmetric Techniques ISO/IEC 11770-2*. International Standard.
- Mitchell, C. J. (2000), 'Making serial number based authentication robust against loss of state', *ACM Operating Systems Review* **34**(3), 56–59.
- Needham, R. & Schroeder, M. (1978), 'Using encryption for authentication in large networks of computers', *Communications of the ACM* **21**, 993–999.
- Neuman, B. C. & Ts'o, T. (1994), 'Kerberos: An authentication service for computer networks', *IEEE Communications Magazine* **32**(9), 33–38.
- NIST (2003), 'Modes of operation for symmetric key block ciphers'. <http://www.ecrypt.eu.org/stream/>; accessed 6 May, 2005.
- Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V. & Culler, D. E. (2002), 'SPINS: Security protocols for sensor networks', *Wireless Networks* **8**(5), 521 – 534.
- Wang, Y. & Chu, B.-T. (2004), 'sSCADA: Securing SCADA infrastructure communications', Cryptology ePrint Archive, Report 2004/265. <http://eprint.iacr.org/2004/265.pdf>.