Chapter 5

IMPLEMENTATION

This chapter describes the implementation of a security policy that will addresss the threats identified by the risk analysis in section 3.2. It describes in detail how the security services identified in chapter 4 can be implemented using smart card technologies. It further elaborates on the technical requirements and specific implementation for each system component. A policy overview is given that describes the security protocols used to provide the security services. Finally possible implementations of mechanisms are given that could be used to implement the protocols.

5.1 SECURITY SERVICES

5.1.1 Availability

In this system the greatest risk is denial of service attacks. These attacks could occur frequently and require little technical expertise or knowledge of the particular network system e.g. flooding of network with useless messages, physical attack on infrastructure. These attack are easily detected but difficult to prevent. Preventing attacks from the public network could be accomplished by a firewall implementation at the gateway. Owners should protect their systems in a similar way. Public traffic will only be allowed once a secure association has been made between the owner and the gateway or node. The possibility of a denial of service attack originating from the private network is low but is still a possibility. Nodes must only accept connections from authenticated entities, a message received from an unauthorized entity should be disregarded and trigger an alarm. The network management must be alerted immediately if any suspect behaviour is observed by a node or a gateway.

5.1.2 Authentication

Authentication is the second largest concern in the system. Users, hosts or service providers and even server nodes might pretend to be a legal entity in order to gain access to the system with malicious intent. The first step is to determine which entities must be authenticated, and by whom. In order to meet the system's functional requirements the following need to take place:

- user authentication by the node
- node authentication by other nodes, the gateway and its owner
- gateway authentication by the owner and the gateway

- owner authentication by the gateway and its nodes

The authentication protocols can be based on public-key or private key cryptography and are shown in section 5.4. The necessary key pairs are generated when a new user, node, gateway or owner is registered. Adequate security checks (e.g. management authorization, background check) must be performed to ensure that these entities are valid. These key data must be stored in a physically secure location and kept for a mandatory length of time in order to provide information for the settling of disputes. Each server, host and gateway also maintain a revocation list of key pairs that have been compromised before their expiry time. This prevents compromised keys being used to gain access to the network.

In the network any user can gain access to services from a node. Taken that a node only has one user it is possible for that node to store unique information that can identify the user. Depending on hardware requirements the user can be authenticated using a PIN or biometric information, e.g. fingerprint profile can be stored on the node's smart card [20]. This legally binds the user to the node.

To prevent illegal entities from accessing the network all nodes on the private network must authenticate themselves before communication between the two entities commences. The gateway keeps record of all nodes housed on the private network. The broadcast nature of the network protocols ensures that the gateway can monitor all network traffic. If a unregistered node transmits on the network the gateway will attempt to authenticate this node. If authentication fails an alarm will be raised. If a node sees that it receives network traffic from a node using its identity it must also raise an alarm. Symmetric or asymmetric authentication mechanisms may be used.

5.1.3 Access Control

Users or nodes gaining access to resources that they are not entitled to are mostly due to problems with authentication. User A might masquerade as User B in order to get B's privileges. A secure authentication protocol should prevent this. The second problem is to allow an authenticated entity access to only specific services, and to regulated what that entity is entitled to do. The simpler the access control conditions the easier it is to implement. An owner can only be accessed by a node it owns while a node can only be accessed by its owner or another node sharing the private network. Nodes and

information on these nodes must be given security labels which indicate clearance levels similar to multi-level security systems like La Padula and Biba [52],[53]. Node-to-node access control can then be done on a node basis (each node has a clearance level, to access a node you need equal or higher clearance) or on a data basis (data has a clearance level, to access data you need equal or higher clearance). Some other examples of access control rules are given below:

- Write/Read: Can the user only read data or also modify data ?
- Owners: Specific assets might only be available to certain owners.
- Clearance: Specific information can only be viewed by users with adequate clearance.

When a node is registered it is given a fixed clearance which cannot be altered by anyone except the owner. Registration is the process of generating a smart card (e.g. adding keys, biometric information and access control information), inserting it into the node (physically sealing it) and connecting the node to the network for the first time (node registers with gateway and owner online before starting to operate).

5.1.4 Integrity and Non-repudiation

Digital signatures provide both integrity and non-repudiation when used in conjunction with a hash function. The hash function which is recommended is SHA-1 although smart cards also provide the MD5, RIPEMD and DES based algorithms. Smart cards provide DSA and RSA PKI signature techniques but private key MACs can also be implemented. A MAC does not offer the same level as DSA or RSA signatures because either of the recipient or sender could have created the signature. RSA has the added benefit of facilitating key exchange. DSA is slightly faster but due to the nature of the algorithm secret information cannot be transferred. It does however prove that a transaction took place between two entities. After the authentication protocol is completed the two communicating entities should have enough information to verify each other's signatures. Two verification processes must be completed:

- The plaintext user information is signed by the providing entity. This provides non-repudiation and ensures integrity of information. This also pads the message sufficiently to ensure higher security for the next signature.

- The encrypted data transmitted on the communication line is signed by the sender. This protects against modification during transmission. Integrity can therefore be checked without decrypting the information.

All entities are required to keep the necessary information to settle disputes. A record of transactions, with signature and required key pair must be kept. This record must be backed up and stored for an indefinite period. The laws governing the specific provision of service might specify such a mandatory time period.

5.1.5 Confidentiality

The authentication protocol implemented must allow for the exchange of a session key. This session key along with a suitable encryption algorithm should prevent leakage of content. Due to the processing and storage constraints it would be more efficient and even more secure to use symmetric key algorithms. It is recommended that the DES algorithm be implemented, although smart cards also provide 3-DES and IDEA, while AES should be provided in the near future. In the distributed system given in section 3.1 there is limited value to traffic flow confidentiality on the private network. In this case it is more important to ensure that the user or service data is not leaked. Traffic confidentiality is still a concern on the public network. Traffic padding is to be used to prevent traffic analysis on the network. Due to the fact that traffic padding might consume unnecessary resources it can be applied only to critical applications. Due to the end-to-end nature of the system only the payload and communication fields not used to route messages (e.g. sequence numbers) may be encrypted.

5.1.6 Key Management

Key management is a process that continues through a key's entire life cycle to prevent disclosure, modifications, substitutions, reuse of expired or revoked keys and unauthorized utilizations. The secure management of cryptographic keys relates to key production, utilization, withdrawal, deletion and archiving. Before the implementation of the above services can be implemented in detail at the system component level it is crucial that it is determined what roles public and private key structure will play in the security policy.

The risk that a key is compromised increases with time and usage. Keys have to be replaced regularly without causing service interruption. The most difficult aspect of this system is the distribution of shared keys between entities to facilitate data confidentiality.

The most efficient method is to have entities share a public-private key pair. The PKI keys are assigned to the node with its implementation and never need to be changed. If a secret key is compromised the node gets assigned a new identity. Session keys can then be exchanged at will if needed for symmetric cryptography techniques using the X.511 authentication sequences. The PKI system also allows for easier authentication, integrity and non-repudiation mechanisms. The system's storage constraints make it impractical for each entity to store another entity's public key as inexpensive smart cards usually have only 4K memory and cheap microcontrollers have even less. Taking into account that these keys vary from 512 - 1024 bits it is feasible that a smart card can store its owner's public key and its own public-private key pair. The X.511 authentication methods will also take considerable time and processing effort. Symmetric secret keys are only 64-bits long for most implemented algorithms, so more of them can be stored. Therefore the node could store the most recent keys used in a similar way to a cache system in a PC. A node can also store a master symmetric key. This key can be used to derive different session and authentication keys. The problem is that the entity it communicates with must also have this master key. Therefore another technique is still needed to distribute master keys.

The TTP CA model of public key distribution is also impractical because the node cannot afford to store another public key, even if it is temporary. If the gateway is a trusted entity it could distribute keys between the nodes. If the gateway cannot be trusted the key distribution might have to occur between the owners. In some cases nodes can be preconfigured with shared master keys if it is known that they will communicate with one another on the private network. The master key never leaves its secure storage and if a key is compromised another can be derived. Although this is feasible if both nodes belong to the same vendors or if the system implementer has the rights to program the security features it does not allow for ad hoc connections between nodes or later network changes. The system requires both asymmetric and symmetric key management. Owners and gateways have greater resources and therefore they can use the protocols described in X.509 and X.511 standards. These entities are also connected over a public network. This allows for authentication using public key certificates provided by a trusted CA. How the keys are managed by the different components will be described further in section 5.2.

5.2 SYSTEM COMPONENTS

In order for a security policy to be successfully implemented each component of the system must address certain security aspects. In section 3.2.1 the system's main components were identified. Threats to each of these components, with regards the functional requirements, were listed in section 3.2.3. This section explains how security services are implemented in order to mitigate threats to the system assets and functional requirements.

5.2.1 Node

The smart card at the node supplies random number generation, symmetric encryption/decryption, hash functions and PKI. This allows the node to secure its data and implement the necessary security service. The nodes provides the following:

1. Time.
2. Sequence numbers of message communicated with different entities.
3. Support for operational commands. The six basic operations are:
   - The gateway must send out broadcast commands regularly to determine whether all nodes are still functioning. Node must respond with a signed identity. This also provides traffic padding on the private network.
   - Node must send data when needed.
   - Node must update its data when needed. This includes receiving and executing instructions, or updating security information.
   - Node must be able to initiate or respond to an authentication request.
   - Node must implement the communication protocol used by the private network.
   - Node must raise an alarm if it detects security violations or suspicious behavious, e.g. receiving a message using this node's ID as the source ID.
4. Data about the service it is monitoring or providing:
   - Data identifier.
   - Data descriptor.
   - Data format, e.g. ASCII, INTEGER, FLOAT.
   - Read/Write access.

- Length.
- Security level of data. Any entity that wished to access this data must have adequate clearance.

5. Information about transactions and actions. Owner will retrieve and store this records to perform a security audit.

The smart card stores data about a number of entities:

1. User, this information authenticates the user:
   - PIN number used to identify the user.
   - Biometric information used to identify the user. This requires newer smart cards with 16-32K of memory.
2. Node
   - The identity of the node.
   - Security level of the node. Used by other nodes to implement access control.
   - The public-private key pair and/or master key used for authentication, non-repudiation and integrity.
3. Owner
   - The public key of the owner. Used in authentication between node and owner. When using a master key with symmetric cryptography this is not needed.
   - The owner's identity.
   - The symmetric key used to communicate with its owner. Assures the confidentiality of transmitted data. This key is timestamped when it is exchanged and expires after a set time.
4. Other nodes
   - The symmetric keys used to communicate with the other nodes. Assures the confidentiality of transmitted data. This key is timestamped when it is exchanged and expires after a set time.
   - The identities of the nodes.

5. Gateway

   - The public key of the gateway. Used in authentication between node and gateway. When using a master key with symmetric cryptography this is not needed.
   - The gateway's identity.
   - The symmetric keys used to communicate with the gateway. Assures the confidentiality of transmitted data. This key is timestamped when it is exchanged and expires after a set time.

One concept needs to be emphasized: Each node has one smart card acting as a security application module. The node and the user inherit the security attributes of that smart card. Therefore the smart card determines the user and the node's access control clearance. It also provides all the mechanisms and information for authentication, integrity, confidentiality and non-repudiation. If a message from a non-authenticated entity is received or the integrity of the message fails the message is immediately discarded. This should prevent unauthorized entities from tying up the node's resources and provide availability.

5.2.2 Private Network

The network itself does not provide much security services but a protocol must be specified to provide functionality for the network entities. Most fieldbus systems only implement the physical (1st) and data link (2nd) layers of the OSI model. Therefore the system designer must implement the network (3rd) and transport (4th) layers. In each case it must be stated whether data can be protected (P) (confidentiality and integrity) or whether it is send unprotected (U). The following must be provided for each data frame:

1. Multicast Flag (U): Used to indicate a broadcast message.
2. Destination identifier (U): Identifies the intended recipient of the message.
3. Acknowledge flag (U): Used to indicate an acknowledge message.
4. Error Flag (U): Indicates that an error has occurred. Used in an acknowledge message.
5. Last message flag (U): Indicates the end of a data sequence.

6. Source identifier (U): Identifies the sender of the message.

7. Sequence number (P): Used to prevent replay attacks.

8. Instruction identifier (P) : Instruction identifier.

9. Data (P): The payload.

The destination and source identifiers are also appended to the payload (once per sequence) and compared to the advertised values once the message is recovered and verified. The sequence number, the acknowledge flag and the error flag provide connection-oriented communication. The acknowledge message will contain the sequence numbers of all the messages it is acknowledging. The gateway must send out broadcast commands regularly to determine whether all nodes are still functioning. Nodes must respond with a signed identity to prove they are still functional. This also provides traffic padding on the private network. All the security services needed are provided by the network entities (the gateway and nodes) and not by the actual infrastructure.

5.2.3 Public Network

The protocols in the public network are well known (TCP/IP) and provide services for the 3rd and 4th layer of the OSI model already. Therefore existing functionality can be used. To provide confidentiality traffic padding can be implemented. The owner can regularly poll the gateway or some of its nodes to see if they are still active. These entities can respond with a signed identity. The TTP CA is also housed on the public network although the security concerns for a CA is beyond the scope of this dissertation.

5.2.4 Gateway

The gateway has sufficient resources to implement random number generation, symmetric encryption/decryption, hash functions and PKI algorithms. It also has unlimited storage and processing resources compared to the node. This allows the gateway to secure its data and implement the necessary security service. The gateway provides the following:

1. Time.

2. Sequence numbers of messages communicated with different entities.

3. Support for operational commands. The five basic operations are:

- Gateway must relay node-to-node and node-to-owner communication.
- Gateway must be able to initiate or respond to an authentication request.
- Gateway must implement the communication protocol used by the private network.
- Gateway must implement the communication protocol used by the public network.
- Gateway must raise an alarm if it detects security violations or suspicious behaviour, e.g. receiving an alarm from a node.

4. Information about the nodes on the private network:
   - Node identifier.
   - Node descriptor.
   - Information on node's owner.
   - Node's public key or master key.
   - Symmetric key used to transmit secure data to that node. This key is timestamped when it is exchanged and expires after a set time.
5. Information about the owners of nodes on the private network:
   - Owner identifier.
   - Owner descriptor.
   - List of nodes owned by the owner.
   - Owner's public key and certificate.
   - Symmetric key used to transmit secure data to the owner. This key is timestamped when it is exchanged and expires after a set time.
6. The public-private key pair used for authentication, non-repudiation and integrity.
7. Information about the operation of the private network. This entails sufficient information for a security audit.

The gateway can either be a trusted entity or it can be assumed that it is untrusted. An untrusted gateway is still authenticated by the nodes and the owners so it not an unauthorized entity. A gateway is seen as untrusted if it can be physically attacked or

the possibility exists that malicious entities can gain access to the communications on the gateway. A gateway secures its communications with both the nodes and the owners. All message are signed to ensure integrity and non-repudiation. A firewall might be implemented at the gateway to provide additional availability and access control services to the private network and the gateway. The gateway is the main access control point in the system. The owners are authenticated and prevented from accessing nodes they do not own. Nodes are authenticated to ensure that only valid nodes reside on the private network.

5.2.5 Owner

The owner has sufficient resources to implement random number generation, symmetric encryption/decryption, hash functions and PKI algorithms. It also has unlimited storage and processing resources compared to the nodes. This allows the owner to secure its data and implement the necessary security service. The owner provides the following:

1. Time
2. Sequence numbers of message communicated with different entities.
3. Support for operational commands. The four basic operations are:
   - Owner must be able to communicate with its nodes, gateways and other owners.
   - Owner must be able to initiate or respond to an authentication request.
   - Owner must implement the communication protocol used by the public network.
   - Owner must raise an alarm if it detects security violations or suspicious behaviour, e.g. receiving an alarm from a node.
4. Information about the nodes on the private network:
   - Node identifier.
   - Node descriptor.
   - Node's public key or master keys for itself and the gateway.
   - Symmetric key used to transmit secure data to that node. This key is timestamped when it is exchanged and expires after a set time.
   - Transaction data from the node used for security audit.

5. Information about other owners:
    - Owner identifier.
    - Owner descriptor.
    - List of nodes owned by that owner.
    - That owner's public key and certificate.
    - Symmetric key used to transmit secure data to that owner. This key is timestamped when it is exchanged and expires after a set time.
6. Information about gateways:
    - Gateway identifier.
    - Gateway descriptor.
    - List of nodes owned by the owner on that private network.
    - Gateway's public key and certificate.
    - Symmetric key used to transmit secure data to the owner. This key is timestamped when it is exchanged and expires after a set time.
7. The public-private key pair used for authentication, non-repudiation and integrity.

The owner needs to communicate with its nodes to provide services and obtain information. Sometimes other owners will need their nodes to speak to a node it owns and therefore ask for permission. All gateways, owners and nodes are authenticated before transactions can take place. An owner secures its communications with the nodes, gateways and other owners. All message are signed to ensure integrity and non-repudiation. A firewall might be implemented at the gateway to provide additional availability and access control services to the owner's IT infrastructure.

5.3 SECURITY POLICY OVERVIEW

The overall system security policy comprises of different protocols that are used under different circumstances.

5.3.1 Node Registration

This protocol describes the steps taken when a node is placed on a private network for the first time. These steps provide the basis for later security operations. Please refer to figure 5.1. Two different protocols are proposed: Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

5.3.1.1 Methods

Scheme 1:

1. User A supplies some secret information, e.g. password or biometric data. User A is authenticated by Node A

2. Node A send its ID, its access rights and its owner's ID to the gateway in plaintext. The message is signed by Node A.

3. Gateway authenticates the owner:
   a) Gateway requests owner's certificate from CA. If gateway still holds a valid certificate for the owner this step is skipped.
   b) Owner requests gateway's certificate from CA. If owner still holds a valid certificate for the gateway this step is skipped.
   c) Gateway authenticates owner and exchanges session key K_{GO}.

4. Gateway sends the node ID to owner. Communication is encrypted using K_{GO} and signed by the gateway.

5. Owner records Node A's location and the gateway's information (e.g. address, public key and certificate).

6. Owner checks its records and sends the gateway Node A's public key. Communication is encrypted using K_{GO} and signed by the owner.

7. Using Node A's public key the gateway verifies Node A's signature of the information send in step 2.

8. Gateway send its public key to Node A.

9. Gateway authenticates Node A. Session key K_{GA} is exchanged.

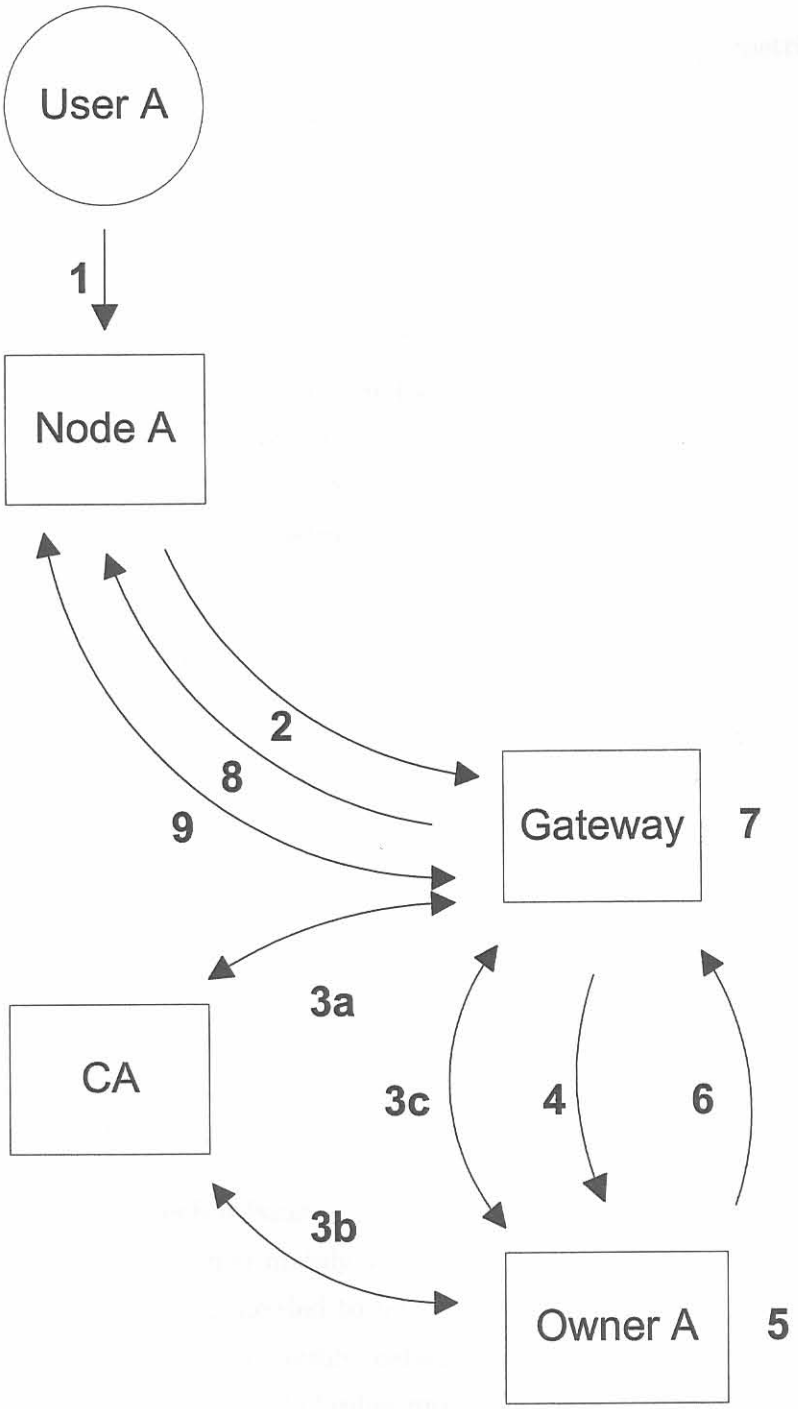

Figure 5.1:
Security protocol for node registration

Scheme 2:

1. User A supplies some secret information, e.g. password or biometric data. User A is authenticated by Node A

2. Node A send its ID, its access rights and its owner's ID to the gateway in plaintext. The message is signed by Node A using master key K_{MGA} or a derived key,.

3. Gateway authenticates the owner:
   a) Gateway requests owner's certificate from CA. If gateway still holds a valid certificate for the owner this step is skipped.
   b) Owner requests gateway's certificate from CA. If owner still holds a valid certificate for the gateway this step is skipped.
   c) Gateway authenticates owner and exchanges session key K_{GO}.

4. Gateway sends the node ID to owner. Communication is encrypted using K_{GO} and signed by the gateway.

5. Owner records Node A's location and the gateway's information (e.g. address, public key and certificate).

6. Owner checks its records and sends the gateway Node A's master key, K_{MGA}. Communication is encrypted using K_{GO} and signed by the owner.

7. Using Node A's master key, K_{MGA} or a derived key, the gateway verifies Node A's signature of the information send in step 2.

8. Gateway authenticates Node A using an exchange protected by the master key or by a derived key. Session key K_{GA} is exchanged or is derived from the master key.

5.3.1.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.

2. Public key cryptography must provide digital signature and key exchange.

   - RSA only available algorithm to provide both.

- X.511 authentication procedure recommended.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication and key exchange sequence on the private network.
4. Public key cryptography must provide digital signature and key exchange on the public network.
   - RSA only available algorithm to provide both.
   - X.511 authentication procedure recommended.

5.3.2 Owner ⇒ Node

This protocol describes the steps taken when an owner initiates communication with a node. Please refer to figure 5.2. All communications between an owner and a node must be secured in such a way that end-to-end security is provided. The gateway must not be able to decipher the messages even if it is trusted. Two different protocols are proposed: Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

5.3.2.1 Methods

Scheme 1:

1. Node A authenticates the gateway and exchanges a session key K_{GA}. If a previous session key is still valid this step is skipped.
2. Node A exchanges a session key K_{OA} with its owner. If a previous session key is still valid this step is skipped.
   a) Node and owner know each other's public keys.
   b) Node requests an owner authentication from the gateway.
   c) Gateway and owner mutually authenticate and exchange session key K_{GO}. If a previous session key is still valid this step is skipped.
   d) Gateway forwards the request to the owner.
   e) Gateway relays authentication sequence between owner and node.

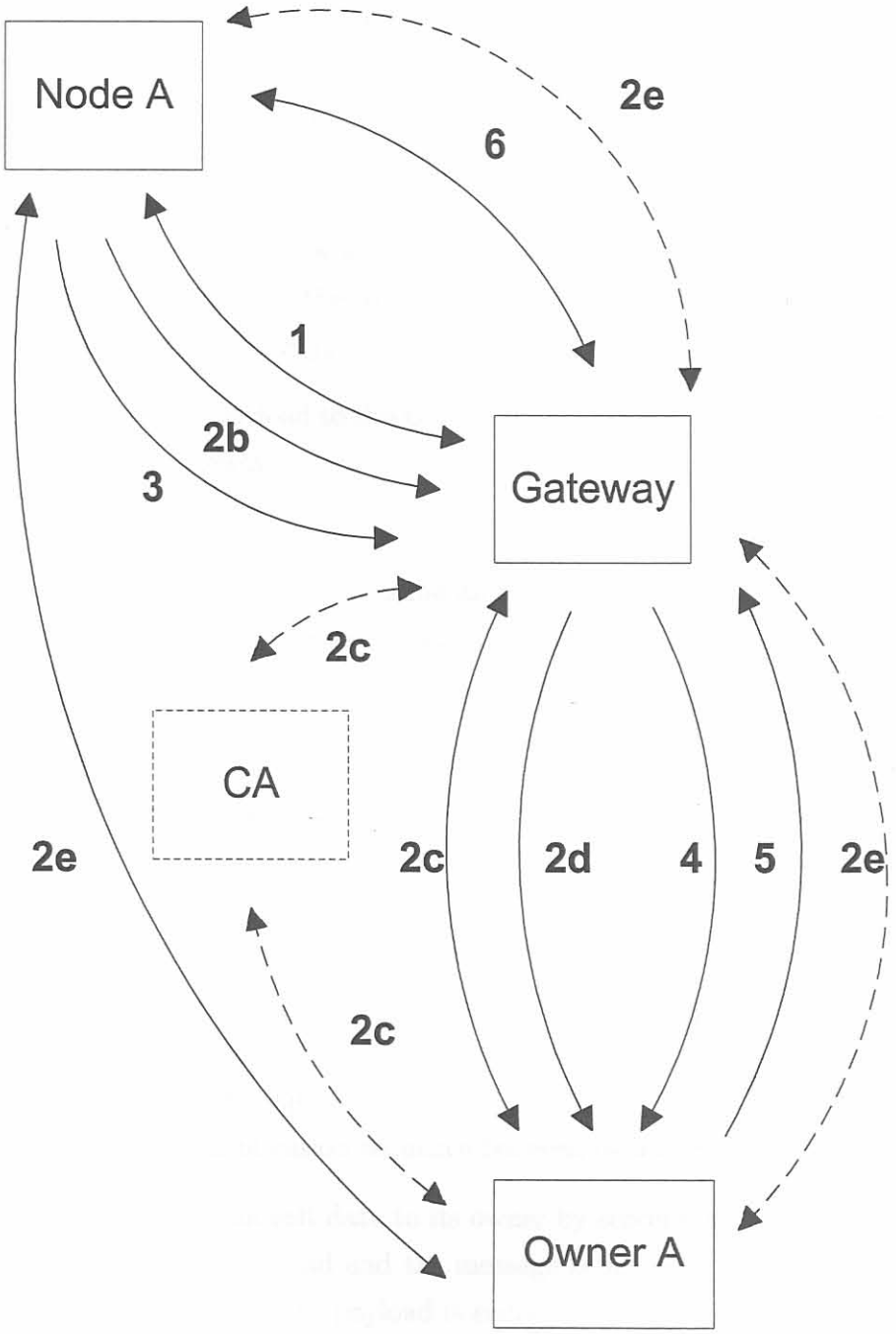

Figure 5.2:
Security protocol for owner ⇒ node communication

3. The node requests to transmit data to its owner by sending the relevant command and payload. Both the payload and the message is signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA}.

4. Gateway forwards the payload to the owner. Communication is encrypted using K_{GO} and signed by the gateway.

5. Response is generated by the owner and forwarded to the gateway. Both the payload and the message is signed by the owner. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA}.

6. Gateway forwards the payload to the node. Communication is encrypted using K_{GA} and signed by the gateway.

Scheme 2:

1. Node A authenticates the gateway using an exchange protected by the master key, K_{MGA}, or by a derived key. Session key K_{GA} is exchanged or is derived from the master key, K_{MGA}. If a previous session key is still valid this step is skipped.

2. Node A authenticates owner using an exchange protected by the master key, K_{MOA}, or by a derived key. Session key K_{OA} is exchanged or is derived from the master key, K_{MOA}. If a previous session key is still valid this step is skipped.

   a) Node and owner know the master key K_{MOA}.
   b) Node requests an owner authentication from the gateway.
   c) Gateway and owner mutually authenticate and exchange session key K_{GO}. If a previous session key is still valid this step is skipped.
   d) Gateway forwards the request to the owner.
   e) Gateway relays authentication sequence between owner and node.

3. The node requests to transmit data to its owner by sending the relevant command and payload. Both the payload and the message is signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA}.

4. Gateway forwards the payload to the owner. Communication is encrypted using K_{GO} and signed by the gateway.

5. Response is generated by the owner and forwarded to the gateway. Both the payload and the message is signed by the owner. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA}.

6. Gateway forwards the payload to the node. Communication is encrypted using K_{GA} and signed by the gateway.

5.3.2.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
   - RSA only available algorithm to provide both.
   - X.511 authentication procedure recommended.
3. Symmetric key cryptography to provide digital signature.
   - Non-repudiation is not needed between the gateway on the node.
   - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.3 Node ⇒ Owner

This protocol describes the steps taken when a node initiates communication with its owner. Please refer to figure 5.3. Two different protocols are proposed. Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

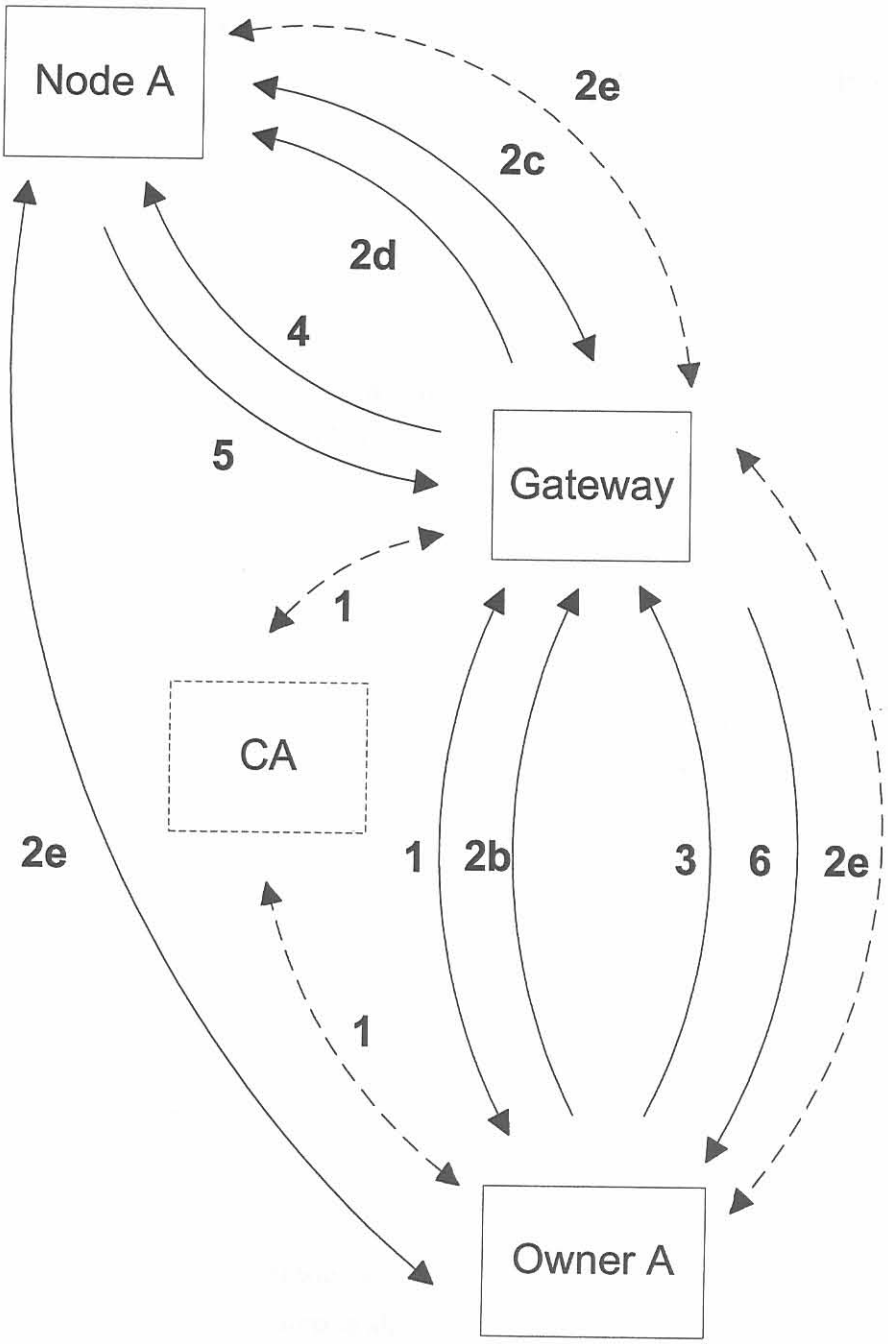

Figure 5.3:
Security protocol for node ⇒ owner communication

5.3.3.1 Methods

Scheme 1:

1. Owner authenticates the gateway and exchanges a session key K_{OA}. If a previous session key is still valid this step is skipped.

2. Owner exchanges a session key K_{OA} with its node. If a previous session key is still valid this step is skipped.
   a) Node and owner know each other's public keys.
   b) Owner requests a node authentication from the gateway.
   c) Gateway and Node A mutually authenticate and exchange session key K_{GA}. If a previous session key is still valid this step is skipped.
   d) Gateway forwards the request to the node.
   e) Gateway relays authentication sequence between owner and node.

3. The owner requests to transmit data to its node by sending the relevant command and payload. Both the payload and the message are signed by the owner. Message is encrypted using K_{GO} while the payload is encrypted using K_{OA}.

4. Gateway forwards the payload to the node. Communication is encrypted using K_{GA} and signed by the gateway.

5. Response is generated by the node and forwarded to the gateway. Both the payload and the message are signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA}.

6. Gateway forwards the payload to the owner. Communication is encrypted using K_{GO} and signed by the gateway.

Scheme 2:

1. Owner authenticates the gateway and exchanges a session key K_{GA}. If a previous session key is still valid this step is skipped.

2. Owner authenticates Node A using an exchange protected by the master key K_{MOA} or by a derived key. Session key K_{OA} is exchanged or is derived from the master key K_{MOA}. If a previous session key is still valid this step is skipped.

a) Node and owner know the master key K_{MOA}.

b) Owner requests a Node A authentication from the gateway.

c) Gateway and owner mutually authenticate and exchange session key K_{GO}. If session key still valid this step is skipped.

d) Gateway forwards the request to the owner.

e) Gateway relays authentication sequence between owner and node.

3. The owner requests to transmit data to its owner by sending the relevant command and payload. Both the payload and the message is signed by the owner. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA}.

4. Gateway forwards the payload to Node A. Communication is encrypted using K_{GO} and signed by the gateway.

5. Response is generated by Node A and forwarded to the gateway. Both the payload and the message is signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA}.

6. Gateway forwards the payload to the owner. Communication is encrypted using K_{GA} and signed by the gateway.

5.3.3.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.

2. Public key cryptography must provide digital signature and key exchange.

   - RSA only available algorithm to provide both.
   - X.511 authentication procedure recommended.

3. Symmetric key cryptography to provide digital signature.

   - Non-repudiation is not needed between the gateway on the node.
   - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.4 Node ⇔ Node: Trusted Gateway

This protocol describes the steps taken when a node needs to communicate with another node in a system where the gateway is trusted. This protocol must be followed each time node-to-node communication takes place so that the gateway can provide proof in case of disputes. Please refer to figure 5.4. Two different protocols are proposed. Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

5.3.4.1 Methods

Scheme 1:

1. Node A authenticates the gateway and session key K_{GA} is exchanged. If a previous session key is still valid this step is skipped
2. Node A requests to communicate with Node B.
3. Gateway authenticates Node B and session key K_{GB} is exchanged. If a previous session key is still valid this step is skipped.
4. Gateway generates session key K_{AB}.
5. Gateway sends this session key to Node A and Node B encrypting with K_{GA} and K_{GB} respectively. Both messages are signed by the gateway.
6. Node A and Node B authenticate each other and share information using K_{AB}. Messages are signed by the sender.

Scheme 2:

1. Node A authenticates the gateway using an exchange protected by the master key K_{MGA} or by a derived key. Session key K_{GA} is exchanged or is derived from the master key, K_{MGA}. If a previous session key is still valid this step is skipped.

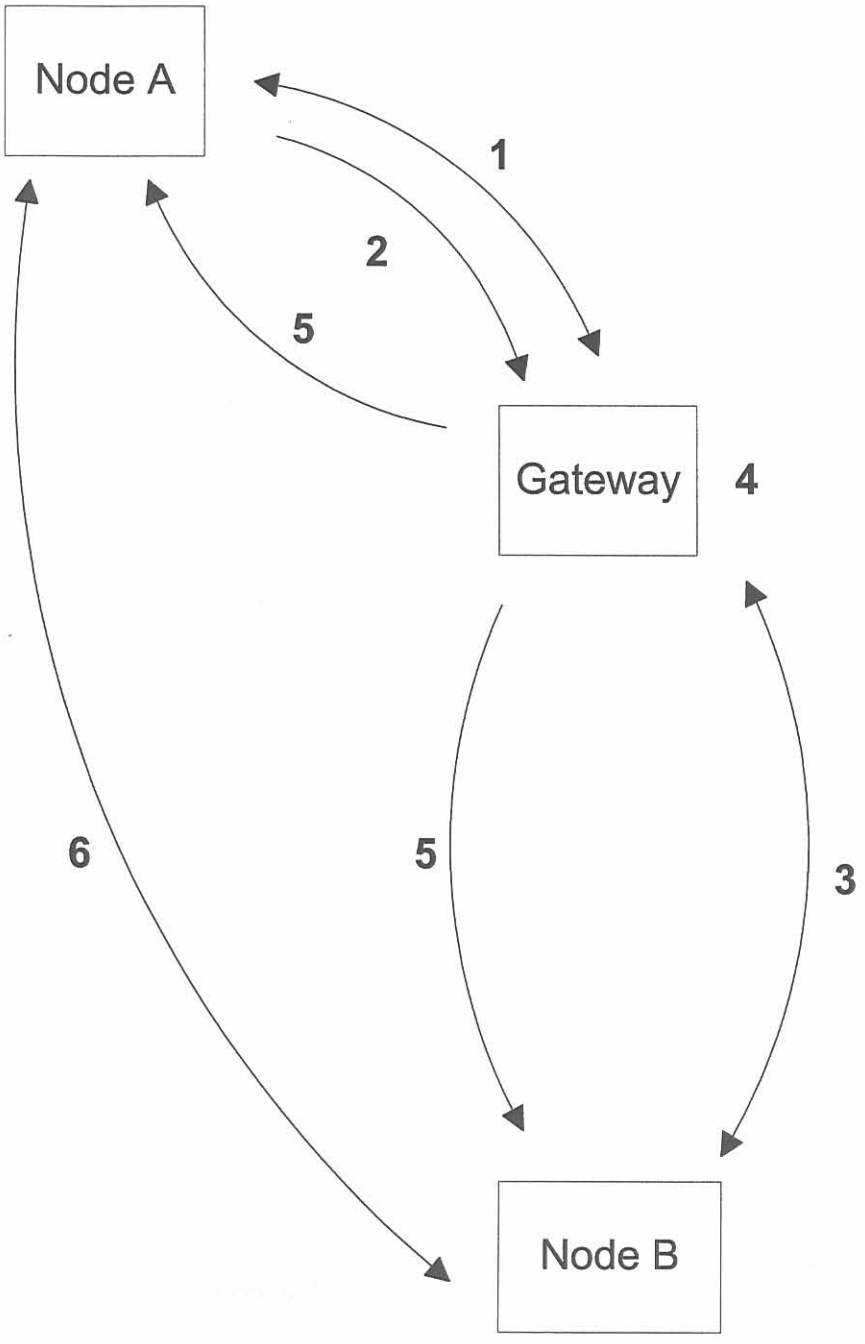

Figure 5.4:
Security protocol for node ⇔ node communication with a trusted gateway

2. Node A requests to communicate with Node B.

3. Gateway authenticates the gateway using an exchange protected by the master key, K_{MGB}, or by a derived key. Session key K_{GB} is exchanged or is derived from the master key K_{MGB}. If a previous session key is still valid this step is skipped.

4. Gateway generates session key K_{AB}.

5. Gateway sends this session key to Node A and Node B encrypting with K_{GA} and K_{GB} respectively. Node B also receives Node A's access control attributes. Both messages are signed by the gateway.

6. Node A and Node B authenticate each other and share information using K_{AB}. Messages are signed by the sender.

5.3.4.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.

2. Public key cryptography must provide digital signature and key exchange.

   - RSA only available algorithm to provide both.
   - X.511 authentication procedure recommended.

3. Symmetric key cryptography to provide digital signature.

   - Non-repudiation is not needed between the nodes as the gateway records the transaction.
   - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.

2. Digital signature using symmetric cryptography, i.e. MAC.

3. Symmetric authentication sequence.

5.3.5 Node ⇔ Node: Untrusted Gateway

This protocol describes the steps taken when a node needs to communicate with another node in a system where the gateway is not trusted. This protocol must be followed each time node-to-node communication takes place so that gateway can provide proof in case of disputes. Please refer to figure 5.5.

5.3.5.1 Methods

Scheme 1:

1. Node A send a request to its owner to communicate with Node B. If needed owner is authenticated and session key K_{OA} exchanged. See scheme 1 in section 5.3.3.
2. Owner A authenticates Owner B and session key K_{OAOB} is exchanged. If a previous session key is still valid this step is skipped. Authentication is done using X.511 and CA certificates. See gateway-owner authentication in section 5.3.1.
3. Owner A requests communication possibilities with Node B.
4. Owner B authenticates Node B and exchanges key K_{OB}.
5. Owners decide on session key K_{AB} and Node A's access control attributes are send to Owner B.
6. Owner B sends K_{AB} and Node A's access control attributes to Node B using K_{OB} and signs the message.
7. Owner A sends K_{AB} to Node A using K_{OA} and signs the message.
8. Node A and Node B authenticate each other and share information using K_{AB}. Messages are signed by the sender.

Scheme 2:

1. Node A send a request to its owner to communicate with Node B. If needed owner is authenticated and session key K_{OA} exchanged. See scheme 2 in section 5.3.3.
2. Owner A authenticates Owner B and session key K_{OAOB} is exchanged. If a previous session key is still valid this step is skipped. Authentication is done using X.511 and CA certificates. See gateway-owner authentication in section 5.3.1.

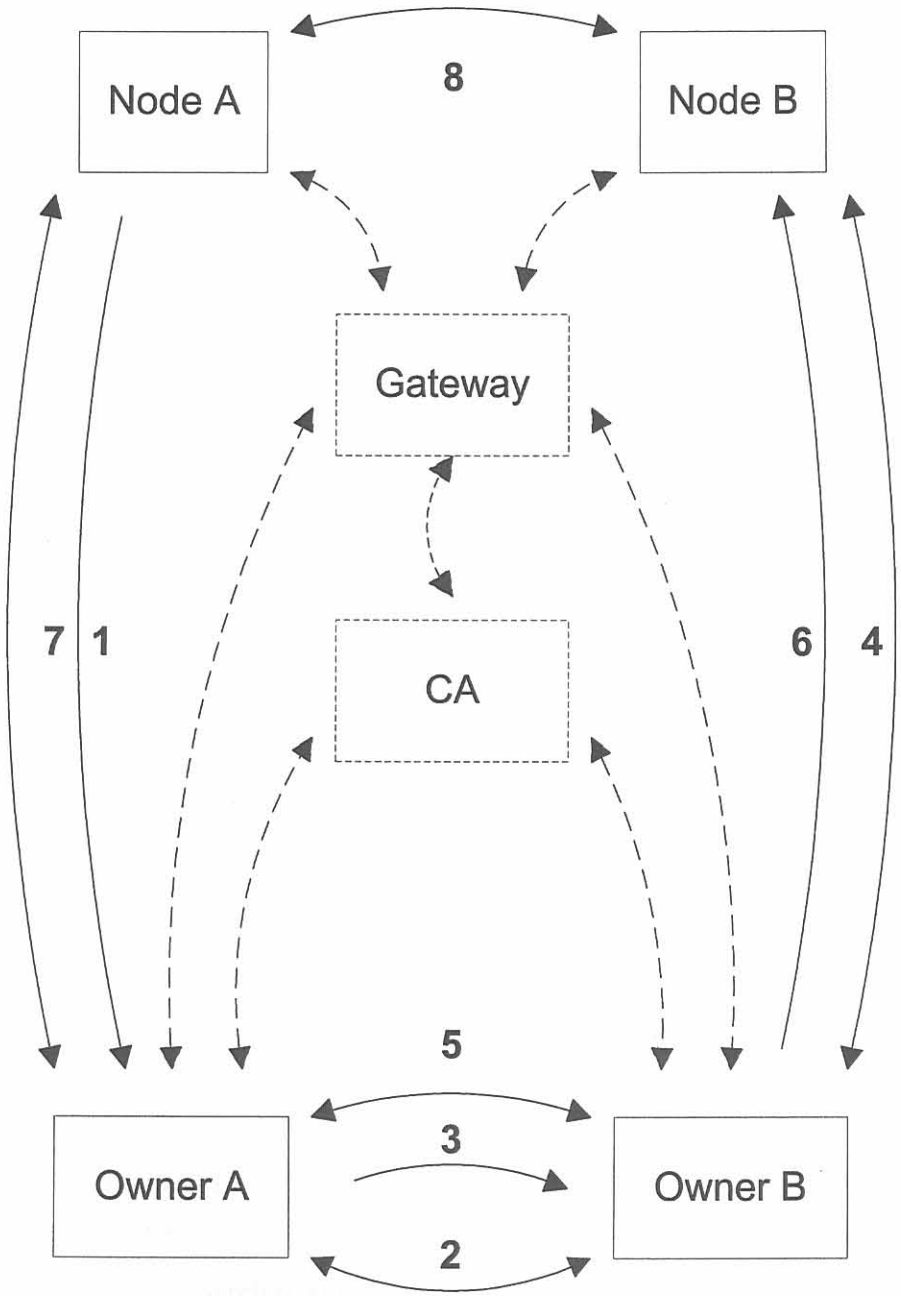

Figure 5.5:
Security protocol for node ⇔ node communication with an untrusted gateway

3. Owner A requests communication possibilities with Node B.
4. Owner B authenticates Node B and exchanges key K_{OB} by using K_{MOB} or a derivation.
5. Owners decide on session key K_{AB} and Node A's access control attributes are send to Owner B.
6. Owner B sends K_{AB} and Node A's access control attributes to Node B using K_{OB} and signs the message.
7. Owner A sends K_{AB} to Node A using K_{OA} and signs the message.
8. Node A and Node B authenticate each other and share information using K_{AB}. Messages are signed by the sender.

5.3.5.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
   - RSA only available algorithm to provide both.
   - X.511 authentication procedure recommended.
3. Symmetric key cryptography to provide digital signature.
   - Non-repudiation is not needed between the nodes as the owners record the transactions.
   - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.6 Node ⇔ Node: Direct

This protocol describes the steps taken when a node needs to communicate with another node and the nodes have sufficient means to negotiate the conditions of communication.

5.3.6.1 Methods

Scheme 1:

1. The other node's public key is obtained using either the gateway's (section 5.3.4) or owners (section 5.3.5) as means of distribution.
2. Node A and Node B then authenticate each other and exchange session key K_{AB}.
3. Messages are encrypted using K_{AB} and signed by the sender.
4. Node A and Node B record transactions for non-repudiation and settling of disputes.

Scheme 2:

1. The other node's master key, K_{MAB}, is obtained using either the gateway's (section 5.3.4) or owners (section 5.3.5) as means of distribution.
2. Node A and Node B then authenticate each other and exchange or derive a session key K_{AB} using K_{MAB}.
3. Messages are encrypted using K_{AB} and signed by the sender using K_{MAB} or a derived signing key.
4. Node A and Node B record transactions.

5.3.6.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
   - RSA only available algorithm to provide both.

- X.511 authentication procedure recommended.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.7 Key Management

Two different schemes are proposed for each scenario. One allows the nodes to use public key encryption while the other scheme only specifies symmetric mechanisms. Figure 5.6 shows the overall key management for scheme 1 and figure 5.7 shows the overall key management of scheme 2.

5.3.8 Additional Considerations

Some additional features must be taken into consideration after implementation. Each entity is responsible for access control to its features. If an entity requests data the entity supplying the service must ensure that access control rules are enforced. An adequate security audit structure must be put into place to detect security breaches. This could be automated by using commercial solutions but a management structure must also be implemented to ensure that user concerns are addressed. In the event of a security breach there must be a set of predetermined action plans to ensure that the situation is rectified and losses minimized. The actions should at least specify that affected key-pairs be revoked and the overall policy reviewed. The policy must be reviewed regularly and information from the audit and the users taken into consideration.

5.4 MECHANISM IMPLEMENTATION

This section shows possible ways to implement mechanisms mentioned in section 5.3.1 to section 5.3.6. Mechanism implementations are not novel and were obtained from literature [14],[27],[54],[55]. The following mathematical conventions are used:

- $K_{AB}[M]$ Message M is encrypted using the key shared between entities A and B.
- $A(M)$ Message M is signed by entity A using a secret key.

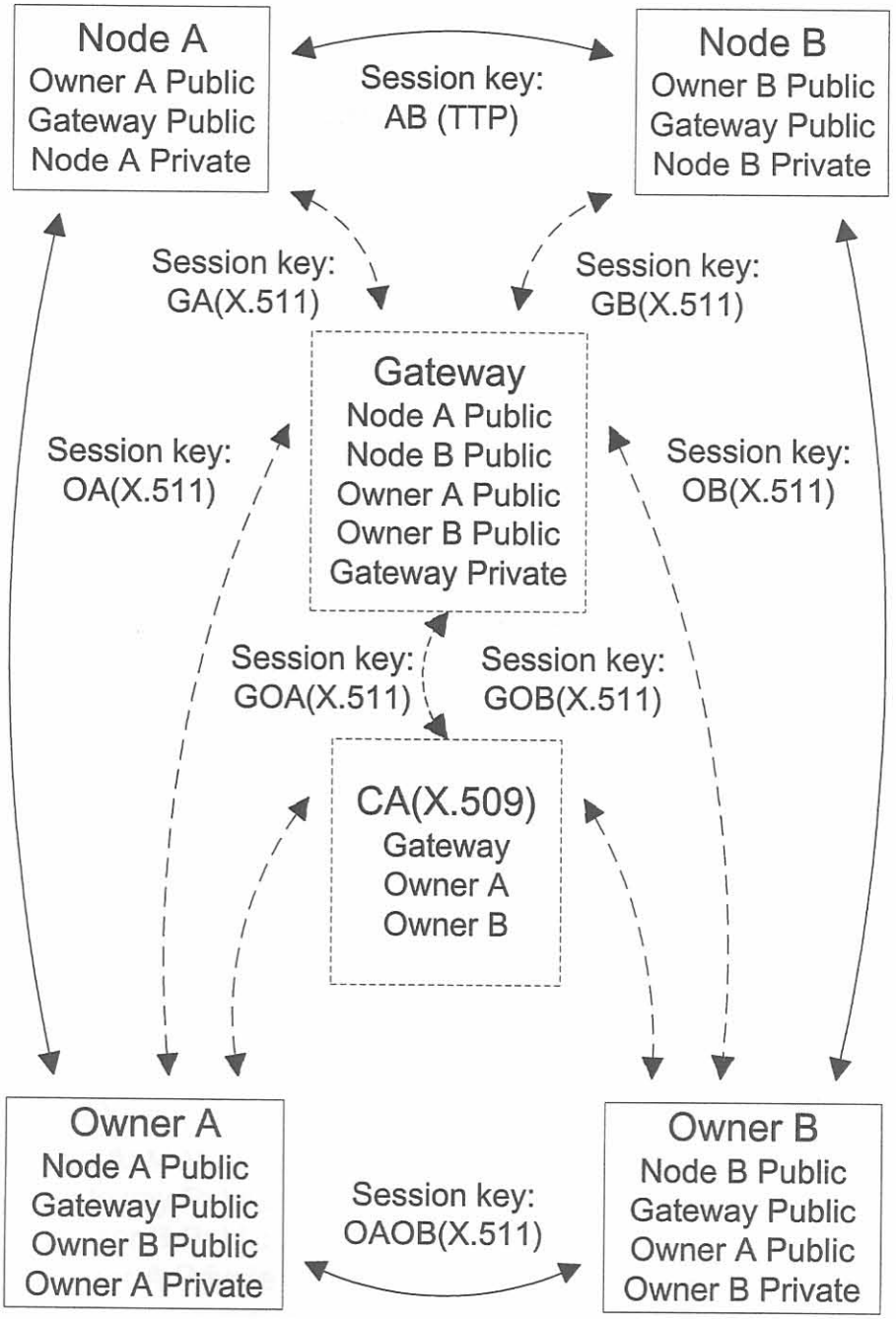

Figure 5.6:
Key management for scheme 1

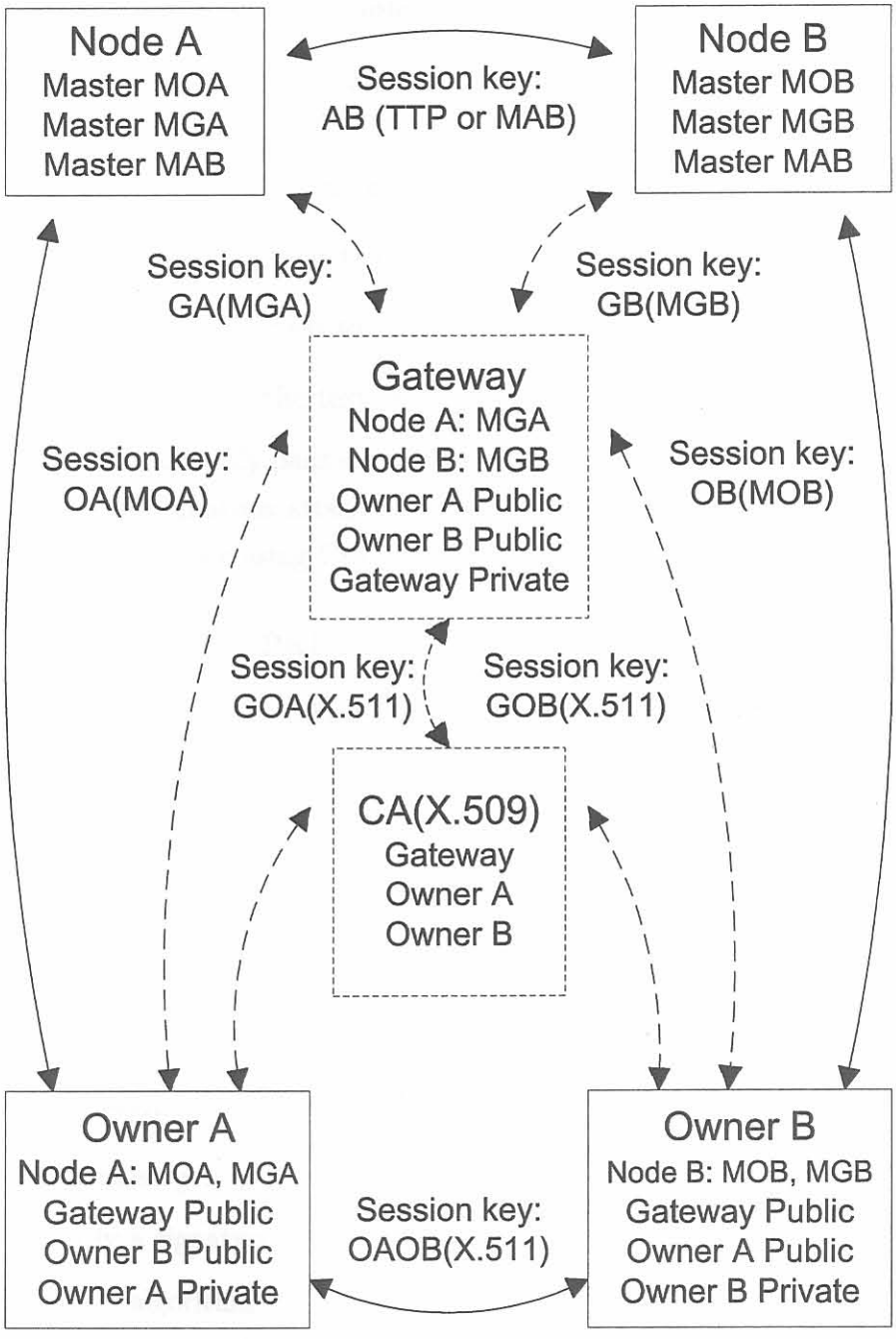

Figure 5.7:
Key management for scheme 2

5.4.1 Confidentiality

Symmetric key cryptography is the easiest mechanism to implement. The steps needed are:

- Select the relevant secret key.
- Select an encryption algorithm, e.g. DES, IDEA, 3DES.
- Select whether encryption or decryption is required.
- Transfer plaintext/ciphertext to smart card.
- Receive the plaintext/ciphertext and store.

The smart card automatically pads a plaintext block to 8 bytes. This padding is random and prevents a crypto-analysis attack that targets small plaintext messages. Successive blocks are chained together using CBC.

5.4.2 Digital Signature: PKI

A digital signature is basically a hash function encrypted with a private key. The steps to compute a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Select private key (generally there will be only one).
- Compute the signature (send Compute Signature command).
- Receive the signature and store.
- Append to message.

The steps to verify a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Select public key.
- Verify the signature (send Verify Signature command and the signature).
- Check if it successful.

5.4.3 MAC: Symmetric encryption

A digital signature is basically a hash function encrypted with a shared secret key. Smart cards also provide a retail MAC function that used 3DES. The steps to compute a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Select secret key.
- Compute the signature.
- Receive the signature and store.
- Append to message.

The steps to verify a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Receive the hash result.
- Decrypt the received signature using the shared key.
- Compare the two - if they are equal the signature is valid

To be secure the MAC works as follows:

$$MAC(M) = E_{key}[key, h(M)]$$

5.4.4 Authentication: PKI

Authentication using public key cryptography uses digital signatures and must facilitate session key exchange. RSA is the only available algorithm that provides both. The X.511 authentication procedure described in section 2.2.3 is recommended.
One Way Authentication:
$A \Rightarrow B\ A(t_A, r_A, ID_B, K_{APublic}[K_{AB}])$
Two Way Authentication:
$A \Rightarrow B\ A(t_A, r_A, ID_B, K_{BPublic}[K_{AB}])$

$B \Rightarrow A\ B(t_B, r_B, ID_A, r_A, K_{APublic}[K_{AB}])$
Three Way Authentication:
$A \Rightarrow B\ A(t_A, r_A, ID_B, K_{BPublic}[K_{AB}])$
$B \Rightarrow A\ B(t_B, r_B, ID_A, r_A, K_{APublic}[K_{AB}])$
$A \Rightarrow B\ A(r_B)$
The use of timestamps t_X are optional when using three way authentication because sufficient replay attack is provided by the nonces r_X.

5.4.5 Authentication: Symmetric

Authentication using symmetric key cryptography needs not facilitate session key exchange. Session keys are derived from a shared master key distributed by a trusted entity. The following challenge/response authentication is proposed:
$A \Rightarrow B\ A(r_A, ID_A, ID_B)$
$B \Rightarrow A\ B(K_{AB}[ID_A, ID_B, r_A, r_B, DerivedKeyInfo])$
$A \Rightarrow B\ A(K_{AB}[ID_B, r_B, DerivedKeyInfo])$

5.4.6 Messages

For the system to function correctly a number of message structures need to be defined. The required message formats are:

- Message format for Node A to gateway:
  $NodeA(K_{GA}[SeqNo, Instruction]NodeA(K_{OA}[Data]))$
- Message format for gateway to owner:
  $Gateway(K_{GO}[NodeA(K_{OA}[Data]))$
- Message format for owner to gateway:
  $Owner(K_{GO}[Owner(K_{OA}[Data]))$
- Message format for gateway to Node A:
  $Gateway(K_{GA}[SeqNo, Instruction]Owner(K_{OA}[Data]))$
- Message format for Node A to Node B:
  $NodeA(K_{AB}[SeqNo, Instruction, Data])$