DESIGN AND DEVELOPMENT OF AN ON-LINE VENDING SYSTEM FOR SELLING PREPAID ELECTRICITY VIA THE INTERNET

by

Gareth Brett Hearn

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Electronic)

in the

Faculty of Engineering, the Built Environment and Information Technology

UNIVERSITY OF PRETORIA

March 2006

SUMMARY

Design and development of an on-line vending system for selling prepaid electricity via the Internet

by

Gareth Brett Hearn

Supervisor: Prof. W T Penzhorn

Department of Electrical, Electronic and Computer Engineering

University of Pretoria

Master of Engineering (Electronic)

Keywords: Prepaid electricity, STS, CVS, Security, On-line, Cryptography, XML, SOAP, Public Key Infrastructure, X.509.

The sale of prepaid electricity is prevalent in South Africa due to the current economic, social, and political conditions. The system currently used for the distribution of tokens for prepaid electricity, CVS, has a design flaw that leads to many security vulnerabilities. The design flaw is that the security devices that generate the tokens are distributed in the field and operate independently of centralised control. This was done because of the limited communication infrastructure in South Africa 10 years ago, but is no longer necessary.

An improvement to the system is suggested that removes the security vulnerabilities by making the system on-line. By employing the communication infrastructure that is available today to provide access to the security devices, the security devices can be located in a secure environment. Changing the mode of operation to on-line also has other advantages such as simplifying auditing and removing synchronisation problems.

This improved system works by communicating on-line with a centralised server and database for every transaction that a customer makes. By doing this, all of the parties involved are kept up to date with the most recent transactions. There can no longer be financial discrepancies and the risk of all parties involved is thus reduced. It is no longer

meaningful to steal the vending machines because they no longer have the ability to generate tokens independently.

In order to implement such a system, however, there are many security aspects that need to be addressed, such as the confidentiality of the information within the system and proving that a transaction did occur between two specific parties. To this end, cryptographic functions and protocols are selected that meet the requirements of the system. Public key cryptography was found to be a necessary ingredient in making the system work effectively and efficiently.

In order to use public key cryptography in the new system, Public Key Infrastructure is required to manage public keys and provide authentication services. A suitable system is developed and described that employs certificate authorities and X.509 certificates. The procedures that are required from each party are listed.

A set of messages that is required for the functions of the system is given. For each message, the contents of the message is given, the parts of the message that must be encrypted are defined and the parts of the message that must be digitally signed are given.

Finally, the security of the individual parts of the system is critically analysed to show that all of the design goals have been achieved. Particular attention is given to the authentication of parties involved in the communication. The security of the system as a whole is also evaluated with respect to the X.810 security framework and it is shown that the system is robust from a security perspective.

The result of the research is a system that meets the required functionality to replace the existing system, and at the same time meets all of the security requirements. It is shown that the proposed system does not have the security flaws of the existing system and thus is more effective in its purpose of vending prepaid electricity.

SAMEVATTING

Ontwerp en ontwikkeling van ’n aanlyn verkoopstelsel vir die verkoop van vooruitbetaalde elektrisiteit deur gebruik te maak van die Internet

deur

Gareth Brett Hearn

Toesighouer: Prof. W T Penzhorn

Departement van Elektriese, Elektroniese en Rekenaar Ingenieurswese

Universiteit van Pretoria

Meester van Ingenieurswese (Elektronies)

Sleutelwoorde: Vooruitbetaalde elektrisiteit, STS, CVS, Sekuriteit, Aanlyn, Kriptografie, XML, SOAP, Publieke Sleutel Infrastruktuur, X.509.

As gevolg van die huidige sosio-ekonomiese toestande in Suid Afrika is die gebruik van vooruitbetaalde elektrisiteit baie gewild en kom dus algemeen voor. Die huidige stelsel wat gebruik word vir die verkoop van sleutels vir vooruitbetaalde elektrisiteit, CVS (“Common Vending System”), het ’n ontwerpersfout wat dit kwesbaar maak vir bedrog. Die fout is dat die apperaat wat die sleutels genereer versprei is in die veld en onafhanklik van ’n sentrale beheerder die sleutels kan genereer. Dit was gedoen omdat die kommunikasie infrastruktuur 10 jaar gelede nie voldoende was sodat die apperaat in die veld ’n sentrale databasis kon skakel nie. Die beperking is natuurlik nie meer vandag van krag nie.

In orde om die stelsel dus te verbeter en die sekuriteits risiko’s uit te skakel stel die verhandeling voor dat die sekuriteits apperatuur aan die Internet gekoppel word. Deur gebruik te maak van die hededaagse kommunikasieinfrastruktuur kan die hart van die stelsel veilig sentraal gehou word. Die aanlyn model het ook ander voordele soos die vereenvoudiging van oudit prosedures en die vermindering van sinkronisasie probleme.

Die verbeterde stelse funksioneer sodanig dat elke transaksie wat ’n gebruiker maak onmiddelik na die sentrale bediener gestuur word vir verwerking. Die sentrale bediener

gebruik dan sy eie databasis om die transaksie te verwerk. Sodanig bly al die partye wat betrokke is by die transaksie in sinkronisasie en kan daar dus nie meer enige finansiele verskille ontstaan nie en die risiko van die model word aansienlik verlaag. Dit is nie meer sinvol om die verspreide veld apperatuur te steel nie aangesien hulle nie meer die vermoë het om selfstandig sleutels te genereer nie.

Die kern aspek van die verspreide stelsel is natuurlik die beiliging van die data wat oor die Internet gestuur word. Daarsonder kom die privaatheid van die gebruiker se persoonlike inligting in gedrang en is dit ook onmoontlik om te bewys dat ’n transaksie wel tussen die twee partye plaas gevind het. Om die probleem op te los is besluit op kriptografie algoritmes en protokolle te gebruik om die data wat gestuur word te beskerm. Daar is besluit op publieke sleutel kriptografie om die stelsel effektief en beskik te maak.

Om publieke sleutel kriptografie te gebruik en om bevestigings dienste te verskaf moet ’n publieke sleutel infrastruktuur geskep word. Die ontwerp en ontwikkeling van ’n toepaslike stelsel word beskryf wat van sertifikaat outoriteite en X.509 sertifikate gebruik maak. Die prosedures wat benodig word vir elke party word ook ontwerp en beskryf.

Verder word die boodskapstelle wat nodig is vir die funksies van die stelsel ook gegee. Vir elke boodskap word die inhoud beskryf, aangedui watter gedeeltes gekodeer word en uitgewys watter dele digitaal geteken moet word.

Laastens word bewys dat die ontwerp’s doelwitte van die stelsel bereik is deur die veiligheid van die individuele dele van die stelsel krities te toets. Daar word veral gekonsentreer op die bevestiging van die identiteit van die partye betrokke in die transaksie. Die sekuriteit van die volledige stelsel word ook ondersoek teen die X.810 sekuriteit raamwerk en dit word bewys dat die stelsel ondeurdingbaar is.

Die eindproduk van die navorsing is ’n stelsel wat al die huidige stelsel se funksionaliteit het met die bykomende voordeel dat dit al die beveiligings meganismes byvoeg. Dit word gewys dat die voorgestelde stelsel nie die sekuriteit tekortkominge van die huidige stelsel het nie en is dus ’n meer effektief in die doel van die verkoop van vooruitbetaalde elektrisiteit.

ACKNOWLEDGEMENTS

I would like to sincerely thank everyone who has contributed in any way to the completion of this dissertation. It has been a journey that was longer than envisioned, had more detours than planned, but has now come to completion.

Firstly, thank you to God, Who is my source of strength, determination, and perseverance. The completion of this work has been more of a spiritual battle than anything else, but has now been won.

Secondly, thank you to my study leader, Prof. Penzhorn. The going has not always been easy, and there have been many differences of opinion, but the result has been significantly better than it would otherwise have been.

Thirdly, thank you to the National Research Foundation for the financial support that was provided in the form of a scholarship (reference GUN 2044153).

Finally, I'd to thank those who have contributed in some other way, whether it be encouragement, actual assistance (such as proof reading or helping to making time available), or just making my life easier so that the work could be finished. These are: my sister Karyn, my mother Averil, my father Graham, my aunt Valerie, Denize, Karlien, Jéan, Nardus, my cell group, the secretaries at Tukkies (Elmien and Cornel), all of the people who prayed, and the various people at SAUPEC 2004 and Africon '04 that gave me words of appreciation, admiration, and encouragement. I have probably omitted some names, but you know who you are, and I thank you for your contribution to the team effort.

Gareth Hearn

March 2006

ABBREVIATIONS

AES	Advanced Encryption Standard
ATM	Automatic Teller Machine
CA	Certificate Authority
CDU	Credit Dispensing Unit
CIS	Customer Information System
CRL	Certificate Revocation List
CVS	Common Vending System
DB	DataBase
DES	Data Encryption Standard
DK	Data Key
EBSST	Electricity Basic Support Service Tariffs
ED	Electricity Dispenser
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IDEA	International Data Encryption Algorithm
IP	Internet Protocol
ISO	International Organisation for Standardisation
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardisation Sector
KDK	Key Distribution Key
KMC	Key Management Centre
MAC	Message Authentication Code
MRK	Master pRivate Key
MUK	Master pUblic Key
MIS	Mainframe Information System
OSI	Open Systems Interconnection
PKI	Public Key Infrastructure
POS	Point Of Sale
RC	Rivest Cipher
RSA	Rivest Shamir Adleman

SGML	Standard Generalized Markup Language
SHA	Secure Hash Algorithm
SK	Signature Key
SMS	System Master Station
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
STS	Standard Transfer Specification
STT	Standard Token Translator
TLS	Transport Layer Security
TM	Transaction Manager
TCP	Transmission Control Protocol
VK	Verification Key
XML	eXtensible Markup Language

TABLE OF CONTENTS

One

INTRODUCTION

1.1 INTRODUCTION

This first chapter describes the motivation, objectives, and research of this dissertation. The subject of the dissertation is the vending system that is used for the sale of prepaid electricity tokens.

Prepaid electricity is being more and more widely used in South Africa. Prepaid electricity systems are favourable when compared to conventional postpaid billing systems for a number of reasons:

- no labour intensive and costly billing has to be done,
- there cannot be overspending or bad debts, and
- fewer administration staff are required during normal operation.

An Electricity Dispenser (ED) is placed at a customer's premises and releases the appropriate amount of electricity. Approximately three million EDs have been deployed in South Africa to date.

1.2 MOTIVATION

The effective operation of the system that is used for the sale of tokens in the prepaid electricity market is essential to the success of the prepaid system. The system that is currently used is called the Common Vending System (CVS). This system has a number of shortcomings due mostly to the manner in which the architecture was originally designed in 1991.

The problems with the existing CVS are mainly due to the off-line nature of the system – there are devices present in the field that are independently capable of producing tokens that represent electricity credit. Because of this, the system has the following weaknesses:

- illegitimate tokens can be produced, and
- tokens that are legitimately sold are not always paid for.

In order to remove these problems from the system, the structure of the system has to be changed from off-line to on-line. In an on-line system, devices that are capable of creating tokens are not required to be present in the field.

On-line communication means that requests are sent to a centralised server that processes the request and sends a response back. To do this, communication infrastructure is required that allows the request for a token to be responded to in a timely manner. Due to its ubiquity and falling cost, the Internet is an ideal means of communication for this purpose.

However, the Internet provides no security directly, and thus the relevant security services must be provided in order for a system that utilises the Internet to be secure and operate effectively.

1.3 OBJECTIVES

The primary objective of this dissertation is to design and develop an on-line vending system that makes use of the Internet for communication, and at the same time overcomes the shortcomings in the existing system. This implies that the necessary security features must be implemented so that the system is not vulnerable to security attacks.

The objectives are thus:

- to design a replacement system that overcomes the existing shortcomings, and
- to ensure that the replacement system is not vulnerable to security attacks.

1.4 RESEARCH

The research that must be done in order to achieve the above objectives is:

- analysis and documentation of shortcomings in the existing systems,
- design of a replacement system that resolves these shortcomings,
- design of the system security and supporting services so that the required security services can be provided,
- working out the details of the replacement system (actual messages used for communication),
- checking that the requirements are met, and
- implementing the system.

The manner in which the existing token vending system is compared to the proposed replacement system is that each of these systems is analysed using the security requirements of X.810. Using this international standard, the security aspects of the two systems can be compared.

1.5 CONTRIBUTION

The result of this work is the analysis of the existing security problems in the existing system, and a replacement for the existing system. The replacement system provides the required security functionality, uses the available technology to the best advantage, and is compatible with as much of the already deployed infrastructure as possible.

The contribution to the field of knowledge is the analysis of the shortcomings in the existing system, and the creation of a system that overcomes the shortcomings.

1.6 CONCLUSION

This chapter has discussed the motivation, objectives, proposed research, and the contribution to the field of knowledge of this dissertation. The result is a system that provides the necessary prepaid electricity vending functionality in a secure manner that is not vulnerable to security attacks.

The next chapter gives an overview of the systems currently used to implement prepaid electricity and to vend tokens for prepaid electricity. The following chapter proposes improvements to the CVS. Then the security building blocks and relevant security protocols are examined. In the following two chapters, the system security of the proposed system is examined and the PKI (Public Key Infrastructure) of the proposed system is given with motivation for the choice. The last chapters give the messages that are used in the proposed system, a critical analysis of the security in the proposed system, and an evaluation of the security of the system as a whole.

Two

INTRODUCTION TO PREPAID ELECTRICITY

2.1 PREPAID ELECTRICITY IN SOUTH AFRICA

The sale of prepaid electricity is becoming increasingly prevalent in South Africa due to the current economic, social, and political conditions. In 1988, Eskom developed the "Electricity for All" concept that was intended to supply electricity directly to a large proportion of the population. One of the methods for implementing this concept is through the sale of prepaid electricity.

Prepaid electricity presently operates in the following manner. A unit, the ED, is installed at the customer's premises to control the amount of electricity provided to the customer. The customer then purchases a token from a supplier. The token provided to the customer represents the amount of electricity that has been paid for. This token is input into the unit that provides electricity to the customer. This unit then makes the corresponding amount of electricity available for use.

Prepaid electricity has become the preferred method of selling electricity in parts of South Africa for a number of practical reasons. Because of the nature of prepaid services, there are no accounts to be paid. There cannot be overspending or bad debts. When the purchased credit has expired, there is no longer service without authorities having to physically disconnect the electricity supply as is required for normal account-purchased (postpaid) electricity.

However, despite the advantages of prepaid electricity, there are a number of difficulties to be overcome. These problems lie in the administration and sale of tokens that are purchased by the consumer to provide electricity at the consumer's premises.

Two complementary systems are currently used in South Africa for the vending and use of prepaid electricity: the STS (Standard Transfer Specification) and the CVS. Both of these are described in this chapter, and the problems that exist are highlighted. Similar systems

are briefly described at the end of the chapter, and the chapter concludes with the problem statement.

2.2 STANDARD TRANSFER SPECIFICATION

In order to facilitate the administration of prepaid electricity, the STS was developed. The STS is a specification that defines how electricity is represented by a token. The token can be interpreted by an ED at the customer's premises, and the corresponding amount of electricity is made available. Systems based on this specification have been widely deployed both within South Africa and internationally, and have enjoyed substantial success. Approximately three million prepayment EDs have been deployed by Eskom to date [1].

Historically, in the field of prepaid electricity, the focus of specification and standardisation by distributors was on the ED, rather than on the vending system and infrastructure required to support the ED. Typically the specification and development of the vending systems were left to the various ED manufacturers. As a result, different vending systems were developed and these were usually incompatible with one another [2].

The most significant consequence of this incompatibility is the inability of the vending system of one manufacturer to vend to the ED of another. Consequently, a distributor purchasing EDs from different manufacturers had to purchase separate vending systems to support the sale of prepaid tokens to each manufacturer's EDs. This proved to be expensive, inefficient, operationally inconvenient, and unnecessarily complex.

The STS was developed to provide an "open system" standard in the electricity dispensing industry. This allows compatible electricity dispensing equipment to work with vending equipment from different manufacturers to the benefit of the customer, distributor, and agent.

Essentially, the STS allows information that is provided by a CDU (Credit Dispensing Unit) to be transported to an ED and be meaningfully interpreted by the ED. This is indicated in Figure 2.1. It is important to note that information transfer is in one direction only – from CDU to ED. This is because of historical reasons where communication in one direction was difficult enough and the cost of the EDs needed to be kept as low as possible. The STS is based on a number of concepts. These are briefly explained below.

Figure 2.1: Information movement in the STS

The system uses a standard *token* [3] that provides the data transport mechanism for the transfer of management and credit information between the CDU and the ED. STS standardises the representation of the following items on a token: the set of token functions that can be transferred via the token, the set of token data fields required by the CDU to support the various token functions, the token formats corresponding to the various token functions, the way in which data is encrypted on the token, the technologies that can be used for the transfer of a token from CDU to ED, and the encoding of token data onto each of the token technologies.

The STS defines the set of *token functions* that can be selected at the CDU and transferred via a token to be executed at the ED. The token functions are classified into three categories: credit transfer token functions (which can only be used by a specific ED), dispenser specific management token functions (which support the transfer of management information, such as key changes, from a CDU to a specific ED or group of EDs) and non-dispenser specific management token functions (which can be used by any ED; normally for testing purposes).

The STS defines the *format* of each token. All tokens are 66 bits long. Two bits of the token indicate which of the three categories is represented by the token. Four bits define

the function within the category that is represented by the token. 44 bits are data, and 16 bits are a checksum so that the integrity of the token can be verified.

For credit transfer function tokens and dispenser specific management function tokens, the rightmost 64 bits are *encrypted* [4]. Before this data leaves the CDU, it is encrypted. Before the data can be meaningfully interpreted by the ED, it is decrypted. The encryption is to ensure that the transfer of data between CDU and ED is secure for data that is specific to an ED.

There are two types of *token technologies*: disposable magnetic card [5] and numeric [6]. The disposable magnetic card technology adheres to the ISO (International Organisation for Standardisation) 7810 series of standards. It is made of paper and is intended to be used once only. A magnetic card reader is required at the ED. The numeric token technology, on the other hand, encodes the data as a string of 20 digits. Because of this, the physical transport medium can vary, and could even be communicated via audio or e-mail. The corresponding input mechanism at the ED is a numeric keypad. This is the more versatile of the token technologies.

For the purpose of installation, operation and management, each ED is allocated a *supply group* by the distributor. This is used to facilitate the management and control of EDs. Three types of supply group are defined: unique supply groups (geographical area), common supply groups (geographical area) and default supply groups (at the time of manufacture).

There are four different STS key types, as indicated in Table 2.1. The type of key that is used at a specific instant in time depends on the environment that the ED is in and thus the purpose for which the key is used [7].

Customers purchasing electricity for an ED at a CDU must be able to identify the ED. This is required because each ED is unique for the purpose of credit transfer token functions. Because of this, each ED must be uniquely identified within the vending system by an *ED number* that is the ED's serial number.

Table 2.1: STS key types

Key type	Abbreviation	Name	Purpose
0	DITK	Dispenser Initialisation STS Key	Used during production or repair. This type of key never leaves the factory.
1	DDTK	Dispenser Default STS Key	Temporary key type used in an ED that has not yet been allocated to a unique supply group or a common supply group.
2	DUTK	Dispenser Unique STS Key	Key type for normal use for EDs that have been allocated to a unique supply group.
3	DCTK	Dispenser Common STS Key	Key type for normal use for EDs that have been allocated to a common supply group and that use magnetic token technology only.

When a customer wishes to increase the amount of electricity that an ED will make available, the customer must purchase electricity from a CDU. The customer provides the CDU with the unique ED identification number and the amount of electricity that is desired. The CDU then produces a token that represents the required amount of electricity. This token is then provided to the ED, either by swiping it through a magnetic card reader (if a magnetic card token) or by typing the 20 digit number into a keypad (if a numeric token). The ED then makes the appropriate amount of electricity available for use.

2.3 COMMON VENDING SYSTEM

The required vending infrastructure for the sale of tokens is called the CVS. The CVS includes CDUs, SMSs and possibly a centralised database, that is not strictly required for normal operation.

CDUs are used to provide tokens directly to customers. They are capable of creating the appropriate tokens given information from the customer, such as meter number and the amount of electricity credit required. In order to do this, the CDU contains a Security Module (SM) that can encrypt the generated token appropriately. The CDUs must also have the appropriate keys for encryption of the token for the specific ED.

SMSs are used to concentrate transaction data collected from CDUs. The centralised database is part of a high-level management system that records purchases and monitors corresponding electricity usage, and obtains this information from the SMSs. The CVS is a system that is implemented according to a set of standards.

The STS provides the means of conveying to an ED the amount of electricity that has been purchased from a CDU. The CDUs are part of the CVS. The CVS is the system that is used to vend the prepaid electricity tokens and close the loop in the system from supplier to customer. Eskom defined and developed the CVS to provide a total electricity dispensing system capable of supporting the widespread deployment of prepayment EDs sourced from a number of different manufacturers.

The CVS consists of multiple groups of CDUs. These are located at various Point Of Sale (POS) locations, with the data from each CDU group concentrated by a System Master Station (SMS). The SMSs are in turn concentrated by a Transaction Manager (TM) on Eskom's Mainframe Information System (MIS). This forms a hierarchical network architecture, as shown in Figure 2.2.

The CVS provides for the vending of an STS token. This token allows a customer to activate the corresponding amount of electricity from an ED when the token is entered into the specific ED for which it was purchased.

The CDU is able to vend STS tokens to customers for EDs designed and manufactured according to the STS, i.e. EDs that support token media and formats defined by the STS. CDUs, by means of STTs (Standard Token Translators), can also support proprietary

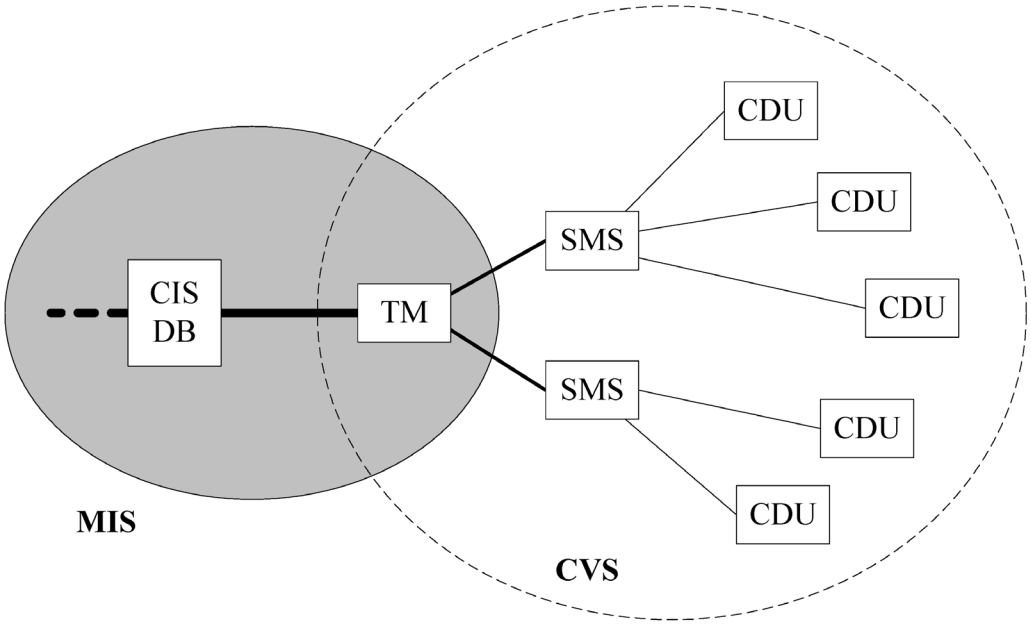

Figure 2.2: Components of the CVS

tokens utilising proprietary algorithms, thereby ensuring backward compatibility for proprietary EDs that have already been installed.

There are three different vending key types that are used to generate the STS keys that were described in Table 2.1. The vending key types are given in Table 2.2 [7]. These types of keys are only used with the CVS; the STS is never aware of them. At any given moment, a unique VDDK exists for each default group defined in the CVS. Similarly, a unique VUDK exists for each unique group, and a unique VCDK exists for each common group.

The directions of information transfer in the CVS are as shown in Figure 2.3. Communication between the centralised database and the SMSs is bi-directional. CDUs only send information to SMSs (for the purpose of transactions), and CDUs only send information to EDs via tokens. There is no reverse path for communication between CDU and SMS, and between CDU and ED.

Table 2.2: Vending key types

Abbreviation	Name	Purpose
VDDK	Vending Default DES Key	Seed key for the generation of type 1 (DDTK) key values only.
VUDK	Vending Unique DES Key	Seed key for the generation of type 2 (DUTK) key values only.
VCDK	Vending Common DES Key	Seed key for the generation of type 3 (DCTK) key values only.

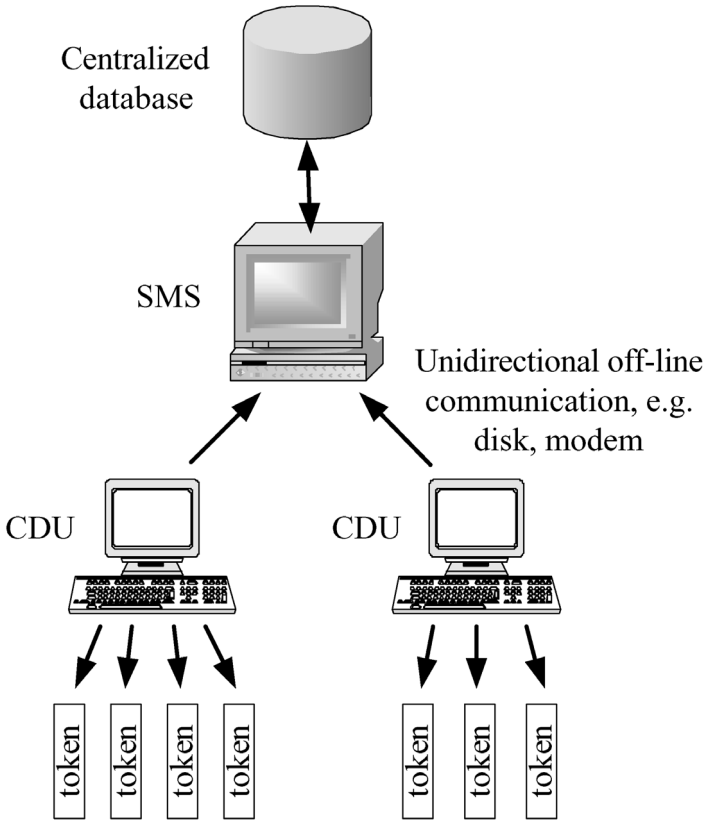

Figure 2.3: Information movement in the current CVS

The communication link between CDU and SMS is typically off-line by means of diskette transfer or modem communication. This is due to historical reasons because communication was not readily available in the rural areas in which such systems were typically deployed. When a transaction occurs, a customer interacts with a CDU and receives a token. The CDU records the transaction information at the time of the transaction. At a later time, several such transactions are sent to the SMS as part of a batch.

This order of events necessitates that the CDU contains an SM that is capable of generating the secure token without any interaction with a higher-level management system, such as SMSs and centralised databases. In other words, transactions occur off-line with respect to the management and monitoring system. The details of every completed transaction are later sent to a centralised database to provide data consistency.

2.4 COMBINATION OF THE STS AND THE CVS

The STS and the CVS work in tandem to achieve the goal of selling and dispensing prepaid electricity. The CVS is the system that is used to provide these tokens to the end customer. The STS is the standard according to which tokens are encoded. Presently, the STS works satisfactorily as has been shown during the 12 years that it has been deployed in the field. The steps in the process of the sale and use of prepaid electricity are shown in Figure 2.4 and are briefly described below.

In *step 1*, the consumer takes money to a vending station and requests a certain value of electricity. The vending station provides a token that represents this value, and that can only be used on the consumer's ED (to assist in fraud prevention).

In *step 2*, the consumer enters the details of the token into the ED. The ED then, by interpreting the token, obtains the amount of electricity that was purchased, and makes that number of units available.

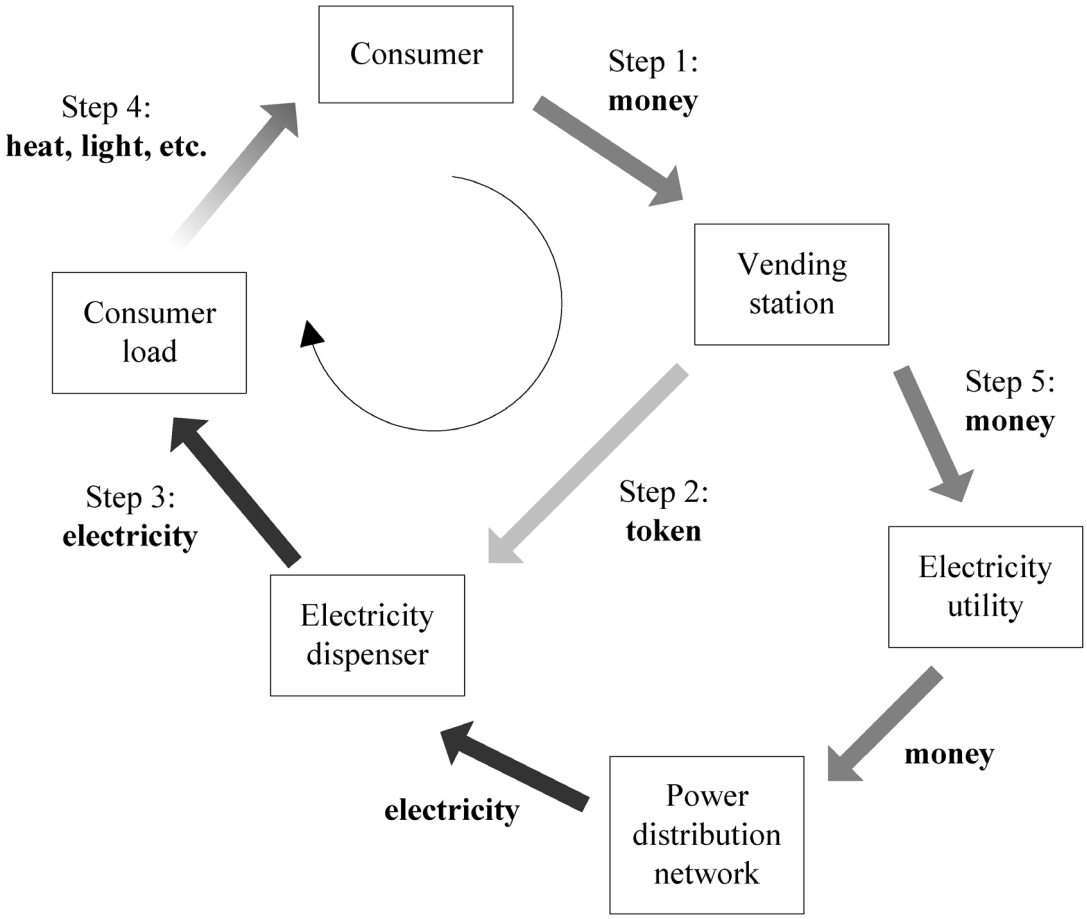

Figure 2.4: The steps in purchasing and using a token

The ED provides the electricity to the customer in *step 3* until no more credit remains. The customer may add more tokens while there is credit remaining to ensure an uninterrupted supply, or when the credit has been depleted and the electricity supply has been cut off.

The customer receives the result of the electricity in terms of heat, light, entertainment, etc. in *step 4*.

In *step 5*, the money is transferred from the vending station to the electricity utility. It is important to note that this step may occur at any time after the token has been given to the customer. Thus it is *not* a requirement for the customer to receive a token.

2.5 PROBLEMS WITH THE CURRENT SYSTEM

An electricity token represents a monetary value of electricity. Because of this, the creation of such a token by illegitimate means is fraud, similar to the manufacture of counterfeit money. The physical entity of a token does not represent the electricity, but the sequence of numbers that the token contains does. The provision of this sequence of digits can also be done in illegal ways.

The following problems exist in the distribution of prepaid electricity tokens:

1. tokens are produced illegally (by using stolen equipment, for example), and
2. tokens that are legitimately sold are not always paid for.

As already explained, the CVS is an off-line vending system. This means that details of transactions that occur at a CDU are not immediately available to the SMSs or the centralised database. The fact that this happens means the CDU is capable of producing tokens with no interaction with a higher level system. The SM that every CDU is required to contain is a security risk because it is capable of producing tokens in isolation. If such a module should be stolen, there is no longer control or accountability for the tokens that the module produces. Similarly, should the CDU lose the record of transactions, or should data be corrupted before communication with the SMS has been successfully completed, the electricity authority has no knowledge or evidence that those specific tokens were sold.

Because a CDU is capable of producing tokens independently, it has all of the knowledge necessary to do so. Should the rules for this change, such as with the recent introduction of the EBSST (Electricity Basic Support Service Tariffs), all of the CDUs have to be updated. This can be an extensive undertaking.

Because of the off-line nature, the problems with the CVS at present, are:

1. the security risk of having many security modules in the field,
2. data synchronisation problems with uploads of transaction records,

3. financial risk exposure of first providing the electricity and then potentially receiving payment for it, and
4. changes to the way in which electricity is billed has to be promulgated to many CDUs before it can be made effective.

2.6 RELATED VENDING SYSTEMS

Systems with similar requirements to the prepaid electricity system were examined to see if there were any related systems in use in South Africa. The prepaid telecommunication services were examined to determine how they operate. In addition, two systems were found that focus on the sale of prepaid electricity in South Africa: Synapse and EasyPay. These are described below.

2.6.1 Prepaid telecommunication services

All of the telecommunication service providers in South Africa (Vodacom, MTN, Cell C, and Telkom) provide a means of purchasing prepaid services. These services typically operate as follows: a customer first purchases an instrument (telephone) and a starter pack. The instrument is somehow connected to the telecommunication network. The customer then purchases a pin number that is entered on the instrument, and the corresponding credit is made available for that instrument to use on the associated network. The customer uses the instrument as and when needed, but the network prevents further service when the credit has been depleted. At this point the customer has to purchase additional pins to reactivate the service.

Various methods are used for the sale of the pin numbers. They can be printed on branded cards that are sold as commodities in shops, purchased at points-of-sale, or purchased via electronic means such as ATMs (Automatic Teller Machines) or the Internet. The two latter methods can be facilitated by EasyPay (see below).

It is important to note that in the telecommunications environment, the instrument itself does not regulate the usage of the service, but rather the network to which the instrument is connected. Thus, regardless of what is done to the instrument, it is not possible to get more

usage from the instrument than that which has been purchased. Thus the sale of prepaid telecommunication services differs fundamentally from the sale of prepaid electricity because the reverse communication channel, where the network communicates with the instrument, exists. With prepaid electricity, however, there is no such reverse channel and the ED makes available the appropriate amount of electricity and then stops. The vending system has no way of knowing how much credit an ED currently has available.

2.6.2 Synapse

Syntell developed an on-line prepaid electricity and water vending system named Synapse [8]. This system provides various channels for the vending of prepaid tokens, such as from a point-of-sale client via radio modem, analogue modem, and Short Message Service. It is very similar to the original CVS that has been widely implemented in South Africa, but also has on-line transaction processing abilities in addition to the standard off-line transaction support.

Because the system provides both on-line and off-line means of vending tokens, the devices in the field have the ability to independently generate tokens. This means that SMs are present in the field and are thus susceptible to the security attacks that have already been explained in the currently deployed CVS. However, it is also possible to centralise the SMs, and so provide the additional security of not having the devices in the field.

The Synapse system is a propriety system that is used for the vending of tokens. Because of this, the internal operation of the system is not publicly known, and customers that purchase the system are locked in to this single system provider. A public system based on an open standard will be a significant advantage to ensure continued support and competitive pricing. For this reason, the development of a system that provides the required functionality will be very beneficial to the industry and the end customers.

2.6.3 EasyPay

EasyPay [9] was developed in South Africa by Prism TranSwitch Services [10]. EasyPay is an interface to Prism's backbone, the TranSwitch, that can be used to purchase prepaid

electricity, pay telephone accounts, pay television licenses, pay for mail order, purchase cosmetics, pay traffic fines, etc.

Many different channels can be used to access the system, including ATMs, the Internet, pay-points of various chain stores, such as Pick 'n Pay, and Shoprite Checkers. EasyPay is a versatile system for the payment of many different types of products and services via various means. It primarily relies on having the infrastructure of a large store available at the point of sail. As such, it is not focussed on the needs of one specific system but provides an interface between the end customer and the service provider. In the case of prepaid electricity, it provides an interface between the various utilities and the end customer for the purchase of prepaid tokens according to the CVS standards.

EasyPay merely provides a means of accessing the system that generates the tokens in the case of prepaid electricity. EasyPay is a system that is designed to generate money, and so is not necessarily the best way to promote the sale of prepaid electricity as there will be significant mark up on the costs of the items that are sold via it.

2.6.4 Differences to the required on-line CVS

The functionality required from the prepaid telecommunication services differs fundamentally from that required by the prepaid electricity system in that there is no return communication channel from the EDs back to the vending system. As such, the systems that are used for prepaid telecommunication services have different requirements from that of prepaid electricity.

Although Synapse provides the required services to a large extent, it is a proprietary system that was privately developed and so does not have industry support. For this reason, customers that purchase this system do not have the assurance that it is backed by the industry and will be supported in the future.

EasyPay is a system that merely provides an interface to a CVS TM and as such its primary purpose is not the vending of prepaid electricity tokens. It relies on all of the other

infrastructure that large chain stores already have, and so is not appropriate for the vending of prepaid tokens where this is the core focus.

Because none of the systems that were described above provide all of the required functionality, there is indeed a need to develop a system that is focussed on securely selling prepaid electricity tokens in a secure and cost effective manner.

2.7 PROBLEM STATEMENT

This chapter has briefly described the existing STS and the existing CVS – two systems used in conjunction with each other to provide a prepaid electricity service and the necessary administration that supports the prepaid electricity service. However, there are serious flaws in the current CVS.

The purpose of this study is to design and implement a system that can be used to replace the current CVS and rectify the existing shortcomings, as no suitable system exists for this task.

The basic characteristics that the new system needs to have are:

- prevention of fraud,
- reliable operation,
- high availability,
- low security risk,
- high level of automation during normal operation, and
- high agility so that changes to billing requirements can be easily implemented.

Specifically, these characteristics will be present in a system that is designed according to the following goals:

1. The sensitive parts of the system must be stored in a highly secure environment, i.e. SMs must not be present in the field where they are susceptible to being stolen or tampered with.
2. The design of the system must ensure that data is consistent between the various components of the system so that administration can be easily handled.
3. The money that is paid for prepaid electricity must reach the electricity utility with a high degree of certainty and reliability so that the utility's financial exposure is limited.
4. The logic that determines how money is translated to electrical units must be centralised so that changes to this can be implemented in a cost effective, efficient manner.

The most important aspect of the new system is that the security used to implement the system must be robust, reliable, and provide all of the necessary security features, as, without this, the system cannot be successful. For this reason, the main focus of this research is on the security aspects of the system – determining exactly what the security requirements are and then ensuring that the system meets the requirements.

The new system that is designed must be cost effective. Thus, due to the large installed base of EDs in South Africa, it is highly desirable to change as few components as possible in the existing STS and CVS in order to limit expenditure.

Three

AN IMPROVED COMMON VENDING SYSTEM

3.1 INTRODUCTION

In this chapter, the basic design of an improved CVS that addresses the problems described in the previous chapter is developed. The requirements of a prepaid electricity system, as well as the shortcomings of the currently deployed system, are described in more detail. The services that are required from a security point of view are listed and defined in the context of a prepaid electricity system. Examples of how each of these required services can be attacked in the currently deployed CVS are given.

The core problem in the existing CVS, that a CDU can independently generate a token, is described and illustrated in the context of the currently used CVS. This is used to develop the proposed system in such a way that the shortcomings no longer exist.

The architecture of the proposed CVS is explained, as well as how the system is going to work by comparison with the existing system. Examples of transactions are used to do this.

Finally, the problem statement is given, i.e. the security design goals of the proposed CVS. The advantages, as well as the disadvantages, that the proposed architecture has over the existing one, are explained.

3.2 REQUIRED SERVICES

It is expected that the system provide various functions. In order to design the system, the functions that the system must provide must first be described. The services that the combination of CVS and STS should provide are:

Confidentiality: The contents of tokens must be kept confidential in order to maintain the security of the system. Confidentiality is needed to a high level of certainty for the internal workings of the system (the security keys) so that the system can be robust against security

attacks. This is similar to storing documents in a safe so that unauthorised parties do not have access to the information.

Authentication: Needed to a high level of certainty so that the relevant parties can be held responsible for payment. Without a strong financial backbone, the system will collapse. Authentication can be compared to a unique signature on a document that only the indicated party can produce.

Integrity: A request that is made for a token and the token itself that is provided must correspond to a high level of certainty. If this is not the case, users will lose faith in the system should the token provided not be what was requested and paid for. This is the same as a written contract that is initialled on every page so that all involved parties agree that the complete document has not been changed.

Non-repudiation: Within STS, the various entities must not be able to deny requests made to each other. This is required to a high level of certainty as the various entities in the system may be operated by different organisations and thus each want to maximise profit. The requesting party must not be able to deny making a request; the receiving party must not be able to deny replying to the request. Non-repudiation is similar to having invoices and receipts as proof that specific services were requested and provided, indicating the parties that did the requesting and providing.

Access control: Only authorised users and entities must be able to access the system. This is required to prevent unauthorised requests for tokens being granted. Access control can be implemented in a physical sense by security personnel preventing unauthorised access to premises or preventing access to safes.

Availability: The system must be available when required so that customers will receive an acceptable level of service and be able to purchase tokens at their convenience. Unavailability is not acceptable from a customer's point of view. Availability can be guaranteed by having sufficient resources available, similar to having enough tellers in a bank so that no client waits more than an acceptable amount of time before being served.

Table 3.1 shows possible attacks that prevent the desired services from being provided in the existing system. The focus of the targets of the attacks are the CVS. Experience has shown that STS is not the main source of security problems in the prepaid electricity industry in practice.

As can be seen from Table 3.1, the weaknesses in the CVS are primarily caused by the following facts:

1) The CDU is capable of independently producing tokens.
2) In normal operation, the CDU operates using credit and payment is made after the transactions have been completed, i.e. the transactions and flow of money are not tightly coupled.

Attacks that have occurred in South Africa include:

1) the sale of a token and not paying for it at all (payment is not required to generate the token legally),
2) the sale of the same "token" multiple times and paying for it only once, and
3) the theft of the technology that is capable of creating a token and then being able to generate tokens without the authorities being able to easily make these tokens invalid.

The attacks listed can occur because a CDU is capable of independently generating a token.

Administrative problems have been caused by the lack of control over the CDU / SMS / TM synchronisation where details of the transactions that have occurred are communicated from CDU to TM for reconciliation purposes. It would be a great advantage if the link between tokens sold and money paid to the utility could be tightly coupled.

Table 3.1: Partial list of possible attacks that could prevent the required services from being provided by the currently deployed CVS.

Service	Possible attacks
Confidentiality	The key change information transmitted in the system could be used to assist in cracking the STS encryption algorithms should it become freely available. CDUs that transmit the largest transaction logs to SMSs could be targeted for theft as they would have a correspondingly large amount of cash on the premises before it is banked.
Authentication	A CDU could send a transaction log as if it were another CDU and in so doing skew the records of the system and disrupt financial record keeping. This could also allow a CDU to be responsible for more or less credit used than would otherwise be the case.
Integrity	A message could be intercepted between TM and SMS or between SMS and CDU and the contents modified. This would allow the CDU to be responsible for smaller payment or skew the transaction details. A party could prevent complete and correct data being sent from the CDU to the TM and thus reduce the amount of money to be paid towards the TM for tokens sold. Another possibility is that the CDU could even sell the value of tokens multiple times.
Non-repudiation	A CDU could deny sending a transaction log and thus claim that it is not responsible for the corresponding debt.
Access control	The CDU could generate tokens independently and so duplicate CDUs could sell the same credit from the TM multiple times.
Availability	Communication between CDU and SMS or between SMS and TM could be interrupted resulting in transaction details not being sent from CDU to TM, even though the transaction between CDU and customer has occurred.

It is true that tokens themselves can be fabricated if the STS encryption algorithm and the applicable keys are obtained, but this has not proved to be a problem in the field. Also, the single direction communication between the CDU and the ED is not a significant problem in practice when compared to the cost advantage of having unidirectional communication.

3.3 CORE PROBLEM IN THE EXISTING CVS

The first goal of the system is that the SMs should not be present in the field where they are susceptible to being stolen, damaged, and mishandled. The reason the SMs are required in the field is obvious when Figure 3.1 is examined. Figure 3.1 shows the sequence of communication messages when a token is purchased by a customer from a CDU.

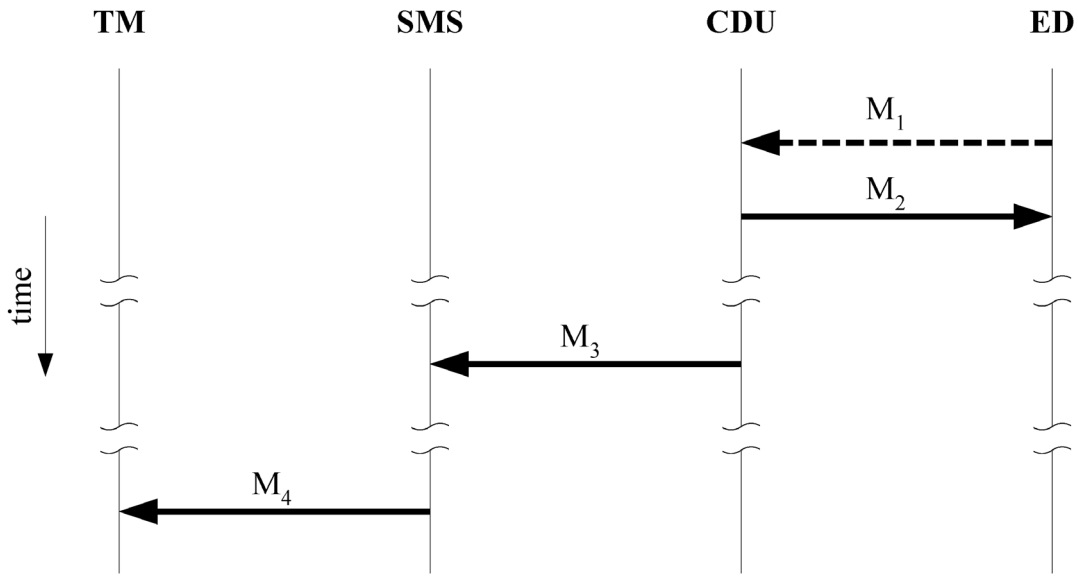

Figure 3.1: Sequence diagram for the purchase of a token in current CVS

The messages in Figure 3.1 have the following meanings:

M_1: The customer goes to the CDU and requests a token for a specific ED for a certain value. Part of this request is the payment for the token. This is an oral request.

M_2: The CDU returns the generated token once payment has been accepted and it has been determined that the ED's serial number is valid.

M_3: At some later point in time, a batch indicating all tokens sold in the last period is transmitted from the CDU to the SMS.

M_4: Again, at some later point in time, a batch indicating all tokens sold by all of the CDUs reporting to the SMS is sent to the TM.

From this sequence of messages, it can be seen that the receipt of a token by a customer is not dependent on that information reaching the SMS, nor is it dependent on that information reaching the TM. This allows the information that the SMS and the TM have regarding transactions to become outdated. The TM's knowledge of transactions will always be outdated because the transaction details are sent only in batches at some point after the last transaction in the batch has occurred. Even worse, the TM may never know about a specific transaction (or batch of transactions) should the data stored on the CDU be lost due to hardware failure or theft.

The logical structure of the existing system is shown in Figure 3.2. The transfer of information was done in this manner in the original design of CVS due to the difficulty in transferring information. Information transfer between CDUs and SMSs was typically done by some manual means, such as diskette transfer or perhaps via a modem if there was telecommunication infrastructure in the CDU's area. The cost of communication was too high and the geographical distances were too large to warrant more regular information transfer.

The TM communicates with the banks (financial institutions that are responsible for money) and also with the SMSs (responsible for co-ordinating finances and tokens within the CVS). The TM is in a secure environment and there are typically very few – in the order of two or three in South Africa. The KDC (Key Distribution Centre) is responsible for generating and distributing the various types of keys indicated in Table 2.1 and Table 2.2.

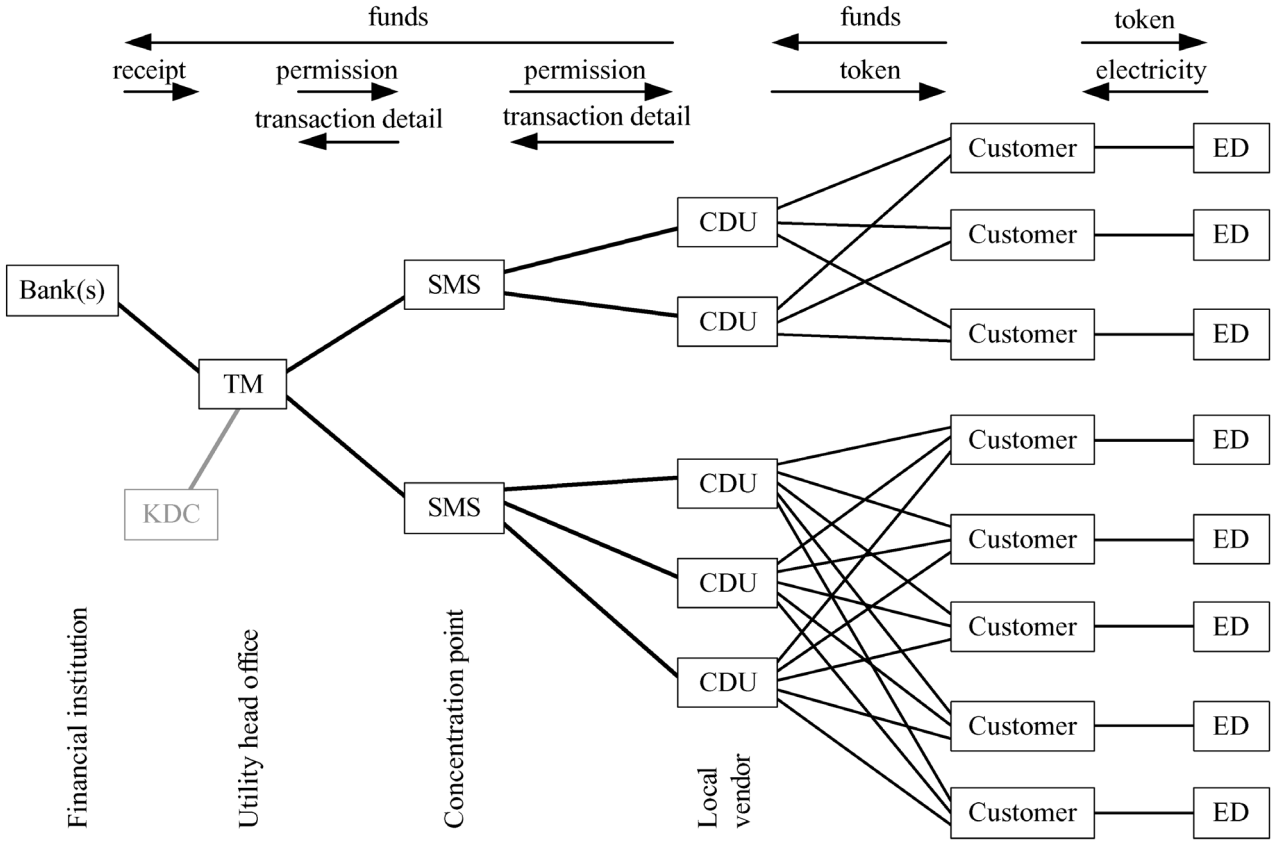

Figure 3.2: Structure of the existing system, including type of information conveyed between entities

The SMSs co-ordinate the system in a geographical area. They are responsible for granting permission to perform transactions and for recording transaction detail for reconciliation purposes. When CVS was initially developed, communication between SMS and CDU was by means of diskette or perhaps modem. As such, this communication link was seldom available and was definitely not real-time. This was done as it was the best technology available at the time.

The CDU is the interface that works directly with the customer (or at least via a shop assistant). On the order of 1000 CDUs have been deployed in South Africa. A customer presents a request to a CDU, and the CDU returns a token to the customer. This token can then be input into the customer's ED to allow the release of electricity. There are in the order of millions of EDs that have been deployed within South Africa.

The process that is followed during the sale of a token is shown in the middle of Figure 3.3 for the currently deployed system.

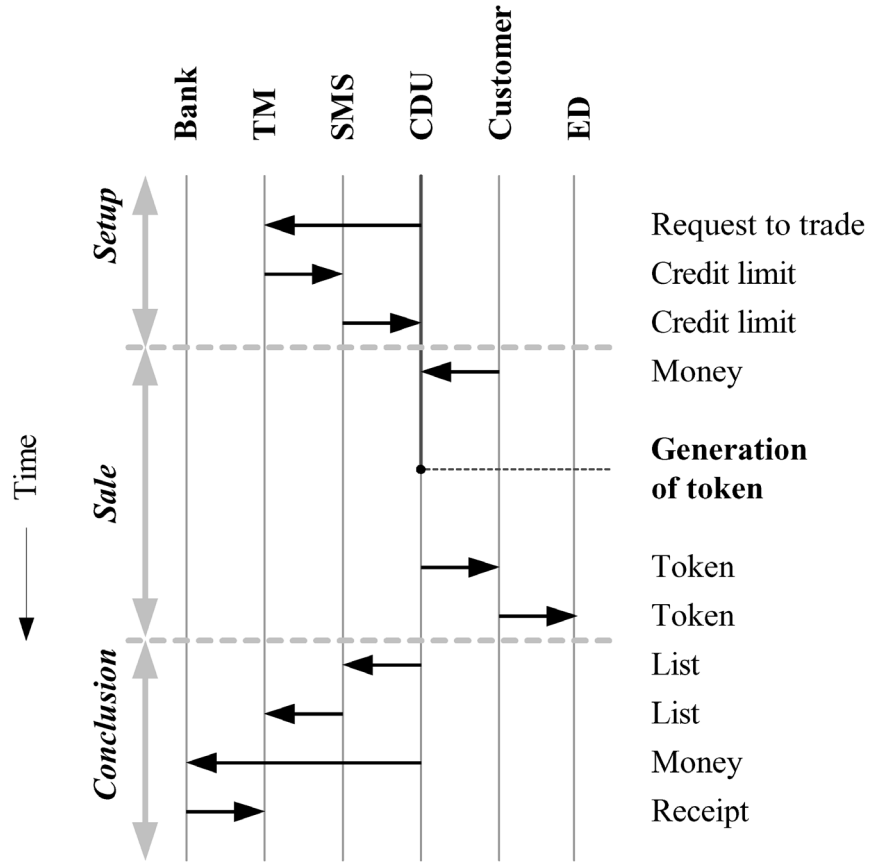

Figure 3.3: Information and money flow before, during, and after the sale of a token in the current system

As can be seen from Figure 3.3, the sale of a token consists of the following steps:

1) The customer presents money to the CDU and requests a token for a specific ED.
2) If valid information is presented to the CDU and the CDU has sufficient credit, it requests the internal SM to generate a token of the appropriate value.
3) The operator of the CDU provides the token to the customer.
4) The customer enters the information on the token into his / her ED.
5) The ED makes the corresponding amount of electricity available.

The fundamental problem that can be seen here is that the CDU is capable of generating the token independently of any other part of the CVS. Recall that there are thousands of CDUs deployed in the field that cannot realistically all be secure. Thus the ability of a CDU to generate a token independently is a weak point in the existing system.

During the setup phase, the CDU requests permission to trade and is issued with a credit limit. During the conclusion phase, reconciliation is performed and the CDU pays for the tokens that it has sold. It is thus obvious that the CDU sells tokens on credit.

3.4 CHANGING THE TRANSFER OF INFORMATION TO ON-LINE

The shortcomings of the CVS described above are due to the manner in which communication between CDU and SMS takes place. A transaction during which a prepaid token is purchased depends on the information provided by the customer, as well as the trustworthiness and reliability of the CDU.

If it becomes a requirement that a CDU *must* communicate with an SMS (or other higher entity) for every transaction, then the problems with the current implementation of the CVS are alleviated.

On-line communication refers to a bi-directional flow of communication between two parties, where data can be exchanged immediately and does not need to be stored until a communication link becomes available. It is similar to a conversation between two people where one person says something and waits for a response from the other. The first person cannot reach a conclusion of what the other person thinks until the response has been received.

On-line vending refers to transactions that can only be concluded at a POS device once the transaction has been authenticated, authorised and recorded by a higher-level management system. This implies that, for a transaction to be processed, there must be a communication link available between the POS device and the higher-level management system. If the communication link is interrupted then transactions are not possible. Furthermore, on-line

systems imply that all information and transactions have to be recorded at a single location (the higher-level management system) only. This single location is the vending server or TM in the CVS.

3.5 ARCHITECTURAL DESCRIPTION

In terms of the CVS, on-line vending means that communication occurs between a CDU and the corresponding SMS for every transaction. This is indicated in Figure 3.4, where it is shown that the various CDUs can also communicate directly with the centralised database. An SMS is no longer required to concentrate data for analysis in the centralised database. This is because communication is no longer limited by geographical constraints as communication takes place electronically and no longer by physical means such as diskettes.

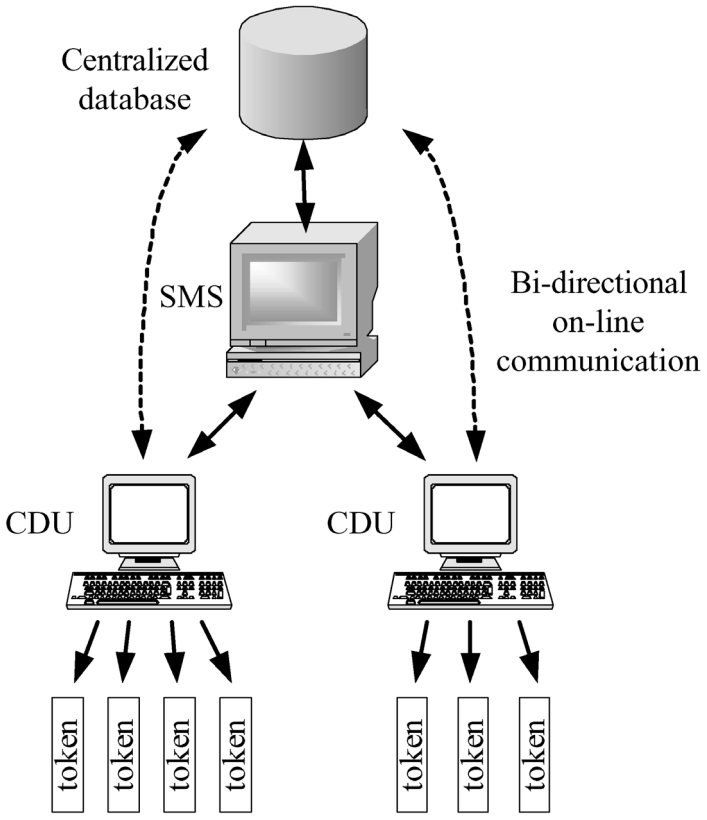

Figure 3.4: Information movement in the proposed on-line CVS

The chain of communication consists of the following entities: centralised database, SMS, CDU, token, and ED at the customer's premises. There are bi-directional communication links between the centralised database and the SMSs. There are also bi-directional communication links between the SMSs and the CDUs. (If there are no SMSs, then there are bi-directional communication links directly between the centralised database and the CDUs.) The CDUs generate tokens that are used in the EDs at the customer's premises, and this communication link is thus only in this direction and is implemented by the STS.

The communication between CDU and SMS (or between CDU and centralised database) can be via any link that is available when a transaction is desired. One likely candidate for this type of communication is the IP-based (Internet Protocol-based) Internet, due to its ubiquity and rapidly decreasing cost.

The concept of an SMS was initially used so that CDUs in a geographical area could be grouped together for logistical reasons. Given that the telecommunication facilities available today allow computers to communicate with other computers anywhere in a country (or almost anywhere in the world), SMSs are no longer required.

Given the above changes, the structure of the proposed system, indicating which parties communicate with which, is shown in Figure 3.5. All CDUs communicate directly with the TM. A customer can obtain a token from any CDU within the system as, for every transaction, the CDU must communicate with the TM as shown in the next section. The KDC in the existing system is upgraded to include a CA, and so the KDC is from now on going to be termed the KMC (Key Management Centre).

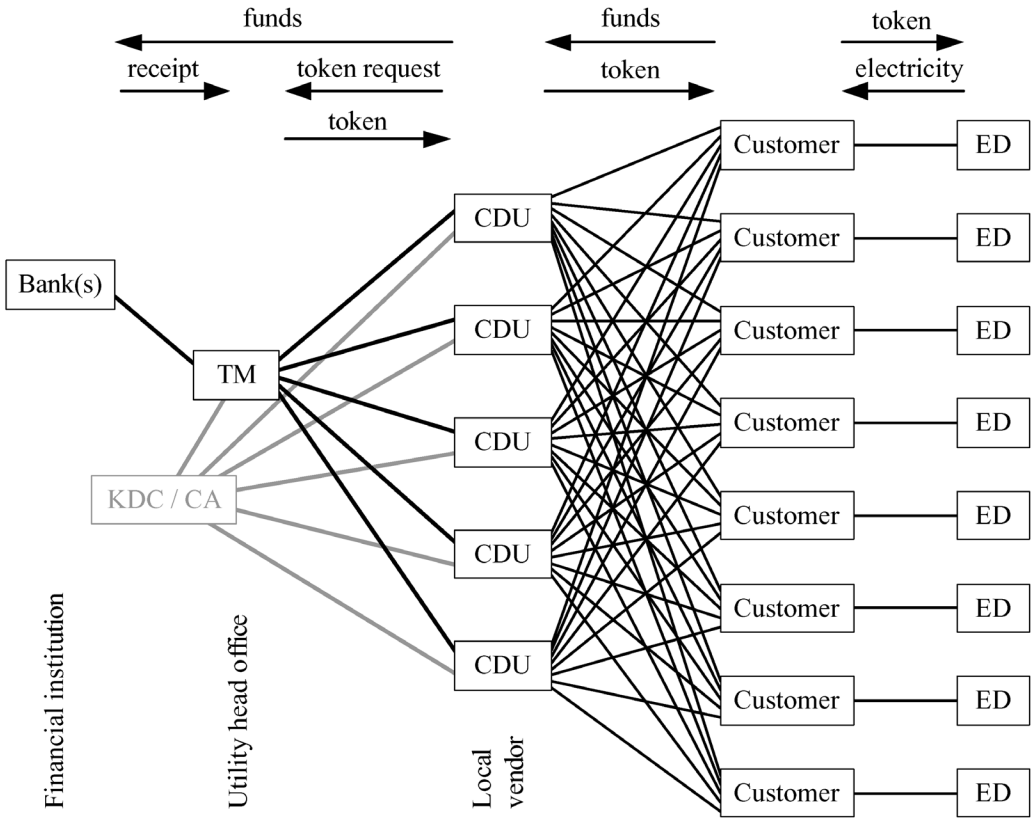

Figure 3.5: Structure of proposed system,
including type of information conveyed between components

3.6 OPERATION

In the on-line vending system, the sequence of events for the purchase of a token is as indicated in Figure 3.6.

The messages in Figure 3.6 have the following meanings:

M_1: The customer goes to the CDU and requests a token for a specific ED for a certain value. Part of this request is the up-front payment for the token.

M_2: The CDU immediately passes this request on to the SMS without modification.

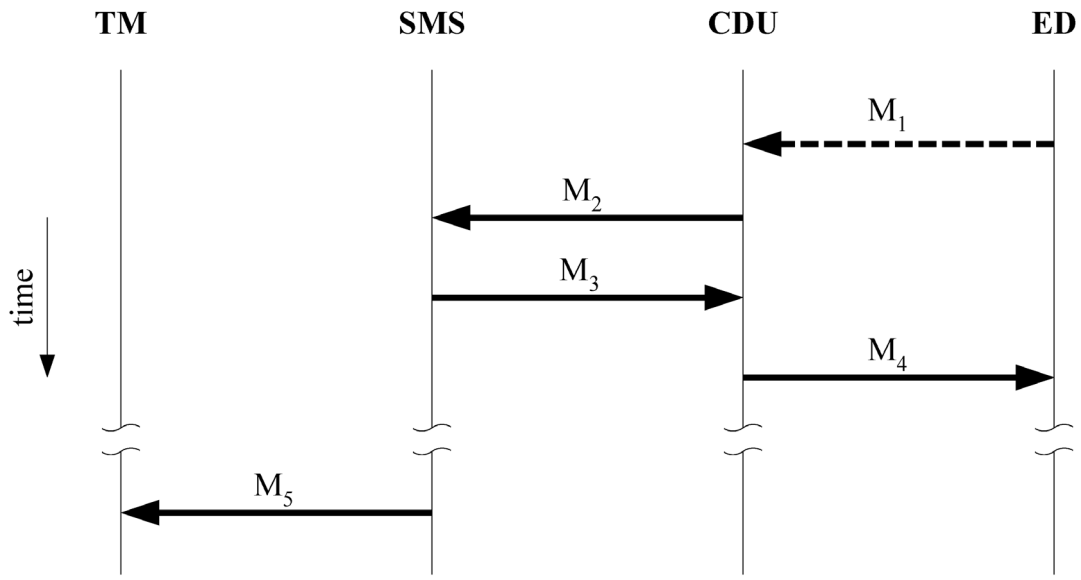

Figure 3.6: Sequence diagram for purchase of token in the improved CVS

M_3: If the SMS determines that the information supplied in the request is valid, and that the CDU is credit worthy for the value of the transaction, it records the purchase, generates the token and sends the token to the CDU. Otherwise the SMS sends the appropriate error message.

M_4: The CDU passes the token received from the SMS to the customer. The customer can then enter the information into his ED to have the appropriate increase in electricity credit.

M_5: At some later point in time, a batch indicating all tokens sold in the last period is transmitted from the SMS to the TM.

In the steps listed above, the CDU requires no intelligence. The CDU no longer needs to have the ability to generate a token, but merely acts as a means of access to the SMS.

In off-line vending, a transaction was completed by a customer going to some outlet that had a CDU, requesting a token for a specific ED, and paying the appropriate amount of money. The customer was then given the token that the CDU made without communicating with any other system.

In on-line vending, a transaction involves more communication between entities. After a customer has requested a token from a CDU and paid appropriately, the CDU sends the request to a vending server. This server, after checking the credibility of the CDU, records the transaction, generates the token, and sends the token back to the CDU. Because of this order of operation, it is not possible for a CDU to defraud the system and not pay for all of the prepaid electricity that it has sold.

Given that on-line communication is available, it is not necessary for a CDU to communicate with an SMS. A CDU can instead communicate directly with the appropriate TM. For our purposes, we are henceforth going to assume that there is no longer a need for SMSs and the communication is between CDU and TM only. The sequence diagram for the purchase of a token is thus as indicated in Figure 3.7.

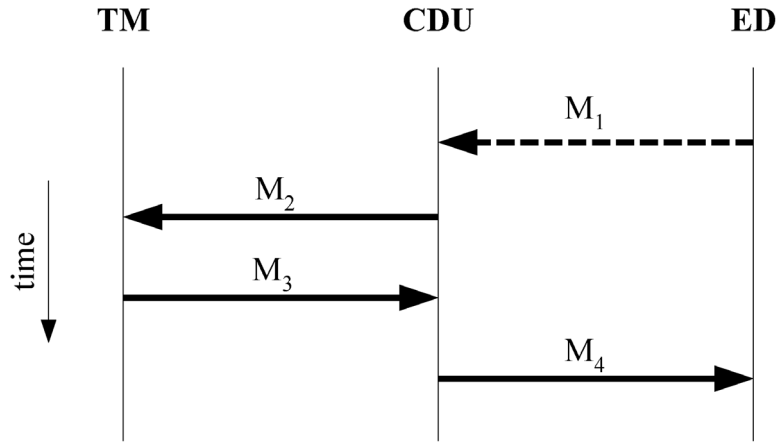

Figure 3.7: Sequence diagram for purchase of token in the improved CVS with unnecessary SMS removed

In Figure 3.7, there are only four messages for the sale of a token. These messages are similar to those in Figure 3.6 except that the TM replaces the SMS. This allows the last message, M_5, to be discarded as the TM already has all of the necessary information after M_2.

The information and money flow in the proposed system is shown in Figure 3.8. As can be seen from this figure, no transactions are pending once a token has been generated and passed to the party that requested it.

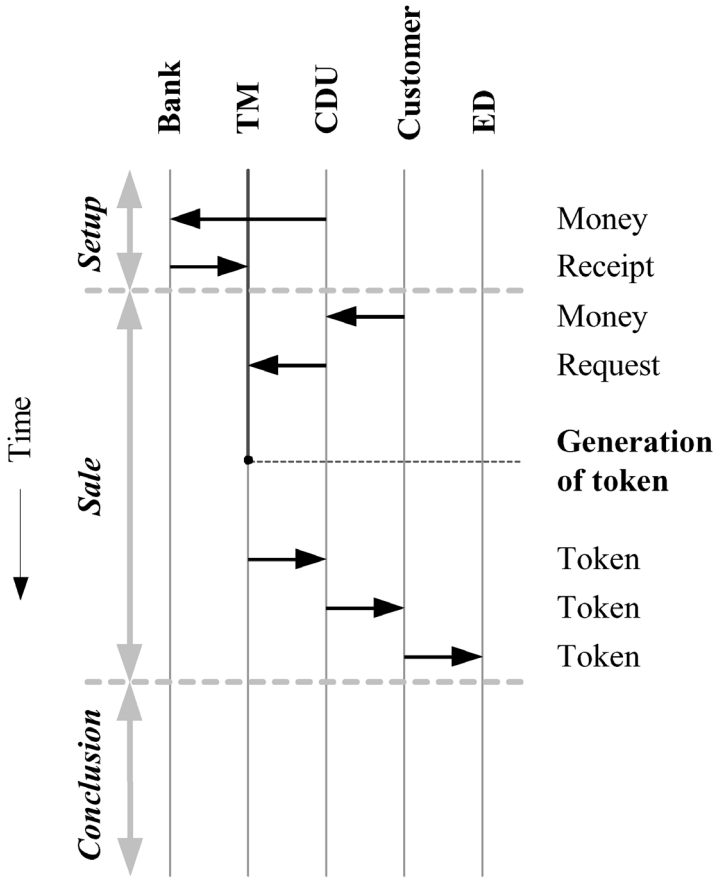

Figure 3.8: Information and money flow before, during and after the sale of a token in the proposed system

3.7 PROBLEM STATEMENT

A means is required by which the problems with the current off-line system can be alleviated in a cost-effective and elegant manner. Such a system must minimise the number of vulnerable areas where a high level of physical security is required. Tokens must also be provided in a timely manner, and the party that sells a token must be held accountable, by means of sufficient proof, for the value of the token.

The design goals of the system are thus:

Goal 1: It must not be required that SMs be present in vulnerable places in the field, such as in CDUs.

Goal 2: Transaction data must be synchronised within a fixed maximum time.

Goal 3: The risk of financial exposure to the electricity utility must be removed by having appropriate credit systems and credit limits in place.

Goal 4: The intelligence of how money and electricity are related must be put into a single place or a small number of easily modifiable places so that changes can be readily implemented.

Goal 5: Tokens must be provided within 10 s of the request being entered into the system.

Goal 6: Proof must be available that holds a party that sells a token accountable for the value of the token. This proof must be legally enforceable.

Goal 7: The new system must use the Internet as a transport medium because it is widely available and its costs are falling.

3.8 ADVANTAGES OVER THE CURRENT SYSTEM

An important advantage of on-line vending as outlined in this chapter – for the purposes of fraud prevention – is that CDUs no longer need to be able to generate tokens independently, i.e. an SM does not need to be present at the CDU. Given that on-line vending allows communication between CDU and centralised database during every transaction, the CDU becomes merely an interface for a customer to request a token from the centralised database. A CDU is no longer an intelligent device with storage as the intelligence of the system has been moved to the vending server. This has the following far-reaching benefits:

- Stolen CDUs are not capable of producing valid tokens. CDUs no longer need to have security modules in them, and thus the reason for their being stolen no longer exists.

- The logistic problems associated with transaction uploads from the CDUs no longer exist. The record of transactions is built up by the TM directly and does not need to be conveyed periodically from the CDUs to the TM.

- Fraud will be reduced since credit control can be implemented more elegantly. Because the transaction information is immediately available, credit limits for CDUs can be implemented to reduce the maximum financial risk. This can be enforced by the centralised database as it can stop issuing tokens when the corresponding CDU's credit limit has been reached.

- As long as sufficient security is in place, there is digital proof of every transaction. This means that the amount of money that each CDU owes the centralised authority is recorded and can be audited by third parties.

- The management, maintenance and security costs associated with the SMs will be vastly reduced because fewer SMs exist. Also, the requirement for the distribution of keys to SMs is vastly reduced and they can all be physically located close to one another in a few locations per country as opposed to one for every CDU.

- Support for debt recovery can easily be integrated due to up-to-date information being available for every client. This can be implemented by providing a communication channel between the centralised database and financial institutions that are owed money by clients of the CVS.

- In implementing the logic required to produce tokens in a centralised location, other communication channels can also be used to vendor prepaid tokens. Examples of this would be ATMs, cell phone banking, and web sites on the Internet. Security is required only to ensure that the centralised database authority receives that which is due for electricity tokens issued.

- Upgrading of the CVS, especially where rate changes and the manner in which fees for a certain amount of electricity are calculated, will be simplified as most of the logic is concentrated into a single place at the centralised database.

An important fact that has to be kept in mind is that the existing installed ED base does not need to be modified in any way for the on-line operation of the CVS to be implemented. Only the CDUs, SMSs, and TMs need to be updated. This represents a significant cost-saving in terms of logistic and capital expenditure, since there are only thousands of CDUs in operation, compared to millions of EDs that have been deployed in South Africa. Only the CVS vending system needs to be modified. Nothing needs to be changed in the STS as the way in which the EDs function remains unchanged.

3.9 DISADVANTAGES WHEN COMPARED TO THE CURRENT SYSTEM

The disadvantages of on-line vending as opposed to off-line vending are:

- The communication required is comparatively more expensive at it must be available for every transaction, and not only once several transactions have been completed.

- No vending can take place if the central vending server or TM is down. This is more significant than in off-line vending because the TM is required by *all* of the CDUs and thus high reliability is essential. An alternative means must be provided if the TM is down, such as backup TMs.

- No vending can take place if communication links are inoperable. The CDUs no longer have the necessary logic to generate tokens independently as they are no longer complete systems by themselves but require the centralised database to be operational.

- Network security needs special attention since the CVS utilises public, insecure network infrastructure for communication. This is addressed in the next chapter.

3.10 CONCLUSION

This chapter has focused on the need to improve the operation of the CVS, and has shown that the required improvement can be achieved by changing the CVS from an off-line to an on-line mode of operation.

The manner in which communication between the various entities in the existing system occurs has been described, as well as how the proposed system will operate.

A list of the design goals of the proposed system has been given. These are used as a guideline during the design of the system in the subsequent chapters.

The consequences of the change from off-line to on-line have been discussed. The advantages achieve all of the required design goals. The disadvantages of the on-line vending approach are deemed to be insignificant when compared to the administrative and financial problems that are being caused by the current off-line system.

Four

SECURITY BUILDING BLOCKS

4.1 INTRODUCTION

This chapter examines the services that can be provided by cryptography from a security perspective. The security building blocks that are relevant to the proposed CVS are briefly described. These include secret and public key cryptography, X.509 authentication and PKI. The relevant notation is also provided.

> "The most basic building block of cryptography is called a *cryptosystem* or *encryption algorithm*. A cryptosystem defines a pair of data transformations called *encryption* and *decryption*. Encryption is applied to data, known as *plaintext*, that directly represents information such as the words or numbers constituting a message. Encryption transforms the plaintext data into unintelligible data called *ciphertext*. A decryption transformation, applied to ciphertext, results in the regeneration of the original plaintext." [11], p. 101.

The use of cryptography allows the secure transmission of information over a data link. Cryptography allows confidentiality to be realised. This in turn allows integrity, authentication, non-repudiation, access control and availability services to be implemented. These services are explained below.

Cryptography is thus the key to implementing the services that are required for the communication within the proposed CVS to occur over insecure public networks.

Also given in this chapter is a briefly outline of the X.810 security frameworks in order to have a reference against which the security of the current and the proposed CVS can be measured.

This chapter ends with the required application of security services in the proposed CVS.

4.2 SECURITY SERVICES

In order to describe the security functionality that is expected of a system, the requirements of the system must be specified in terms of the applicable security jargon. These terms include [12]: confidentiality, authentication, integrity, non-repudiation, access control and availability.

Confidentiality ensures that the information in a computer system and transmitted information can only be meaningfully interpreted by authorised parties. It is not computationally feasible for a third party to intercept the transmitted information and be able to interpret it meaningfully.

Authentication ensures that the origin of a message or electronic document is correctly identified, with the assurance that the identity is not false. The third party is provided with digital proof that can be verified by an external third party that the message was indeed originally sent by the claimed party.

Integrity is the guarantee that only authorised parties are able to modify computer system assets and transmitted information without changes being detected. This mechanism allows a recipient to be sure that any party other than the one that sent it has not modified the message it has received. Again, appropriate proof is available by means of which a third party can prove that the message was not modified by a party other than the original source.

Non-repudiation refers to the fact that neither the sender nor the receiver of a message is able to deny the transmission of that message. Both the sender and the receiver have sufficient proof that the corresponding party received or sent the message.

Access control ensures that access to information resources may be controlled by or for the target system. This means that no unauthorised parties may access a resource without the permission of the target system or some entity acting on behalf of the target system.

Availability requires that computer system assets be available to authorised parties when needed. This implies not being available to unauthorised parties.

4.3 SECRET KEY AND PUBLIC KEY CRYPTOGRAPHY

There are two types of cryptosystems: secret key and public key. In secret key (or symmetric) cryptography, a key is used to encrypt the plaintext, and *the same key* is used to decrypt the ciphertext. Examples of secret key algorithms are Data Encryption Standard (DES), Advanced Encryption Standard (AES), triple-DES and the Rivest Ciphers RC2, RC4, RC5 and RC6. An advantage of secret key cryptography is that it is extremely fast and easy to implement, especially in hardware.

In public key (or asymmetric) cryptography, a key is used to encrypt the plaintext, and *a different key* is used to decrypt the ciphertext. The key that is kept private by the system concerned is called the private key, while the other key, the public key, can be publicly disclosed. It is not mathematically feasible to derive the private key from the public key, or vice-versa. Examples of public key algorithms are RSA (Rivest Shamir Adleman) and elliptic curve cryptography. An advantage of public key cryptography is the fact that there are two separate keys for encryption and decryption. This allows the creation of digital signatures (explained shortly). A disadvantage of public key cryptography is that it requires significantly more processing power than a similarly strong secret key cryptographic equivalent.

A Message Authentication Code (MAC) (also known as a message digest) is an additional piece of data that is generated by the message originator and which accompanies or is logically associated with the message in transit. The value of the MAC depends on all of the bits of the message being transmitted in such a way that if any bit or number of bits in the original message are changed, the probability that the value of the MAC will also change is extremely high. It is thus possible to use the MAC to determine whether the message has been changed since the MAC was calculated. A MAC is typically calculated by means of a one-way hash function that is easy to compute, but difficult to reverse. A

commonly used example of a hash function is Secure Hash Algorithm (SHA-1), according to [11], p. 113.

If a message and its corresponding MAC or hash code are transmitted, a third party can modify the message and then recalculate the MAC. This will result in the receiving party having no way of knowing whether the message has intentionally been modified. Digital signatures can be used to prevent this by making use of public key cryptography. The sending party calculates the MAC and encrypts it using its private key. Any party can then check that the MAC is correct by decrypting it with the publicly available key. No party other than the sending party can generate such an encrypted MAC, hence the term digital signature. Public key cryptography, such as RSA, can be used to create a digital signature from a MAC. An algorithm specifically designed to sign digitally, such as Digital Signature Algorithm (DSA), can also be used to sign, given a MAC. DSA is based on the difficulty of computing discrete logarithms.

In order to distribute public keys, a mechanism is needed by which parties can reliably receive such information. Digital certificates are used for this purpose. A digital certificate contains information about an entity, such as the organisation name and its public key. This certificate is digitally signed by a third party that is mutually trusted by everyone involved in the communication. The digital certificate indicates the details related to the entity described. The party that signs the certificate also indicates the degree of certainty that it has that the information contained in the digital certificate is correct. This party, called a Certificate Authority (CA), has policies that describe the methods that it uses to assure that the information provided is correct. The most common standard for digital certificates is X.509 [12], pp. 341 - 349, and is described shortly.

A CA gives a level of assurance for a particular digitally signed certificate. One CA can digitally sign a certificate belonging to another CA. In this way, a tree of trust can be established whereby certificates can be traced back to the top level of the CA hierarchy, also known as a root CA. This is called a chain of trust. Examples of root CAs are Thawte [13] and VeriSign [14]. The infrastructure required to issue and maintain digital certificates

is called PKI and is based on CAs that are responsible for creating and distributing certificates.

4.4 RSA

The RSA algorithm makes use of an expression with exponentials [12]. Plaintext is encrypted in blocks, with each block having a binary value less than some number n. In practice, the block size is 2^k bits, where $2^k < n \leq 2^{k+1}$. The key generation, encryption and decryption algorithms are given in Figure 4.1.

Key generation		
	Select p, q	p and q both prime
	Calculate $n = p \times q$	
	Calculate $\varphi(n) = (p-1)(q-1)$	
	Select integer e	$\gcd(\varphi(n), e)=1$; $1 < e < \varphi(n)$
	Calculate d	$d = e^{-1} \bmod \varphi(n)$
	Public key	$KU = \{e, n\}$
	Private key	$KR = \{d, n\}$
Encryption		
	Plaintext:	$M < n$
	Ciphertext:	$C = M^e \pmod n$
Decryption		
	Plaintext:	C
	Ciphertext:	$M = C^d \pmod n$

Figure 4.1: The RSA Algorithm

The RSA algorithm is based on Euler's theorem: given two prime numbers, p and q, and two integers, n and m, such that $n = pq$ and $0 < m < n$, and arbitrary integer k, the following relationship holds:

$$m^{k\varphi(n)+1} = m^{k(p-1)(q-1)+1} = m \bmod n, \tag{4.1}$$

where $\varphi(n)$ is the Euler totient function, which is the number of positive integers less than n and relatively prime to n. If p and q are prime, then

$$\varphi(pq) = (p-1)(q-1). \tag{4.2}$$

Thus, the desired relationship can be achieved if

$$ed = k\varphi(n) + 1, \tag{4.3}$$

which is equivalent to saying

$$ed \equiv 1 \bmod \varphi(n) \tag{4.4}$$

$$d \equiv e^{-1} \bmod \varphi(n). \tag{4.5}$$

So the relationship of Equation 4.3 can be achieved if e and d are multiplicative inverses mod $\varphi(n)$. According to the rules of modular arithmetic, this is only true if d (and therefore e) is relatively prime to $\varphi(n)$, which means that gcd($\varphi(n)$, d) = 1.

In the RSA scheme, the ingredients are:

- p and q which are two prime numbers (private, chosen)
- $n = pq$ (public, calculated)
- e, with gcd($\varphi(n)$, e) = 1; $1 < e < \varphi(n)$ (public, chosen)
- $d \equiv e^{-1} \bmod \varphi(n)$ (private, calculated)

The difficulty in decrypting ciphertext that was encrypted with RSA is that, given n, it is very difficult to calculate p and q for large values of p and q.

4.5 NOTATION

The following notation will be used throughout:

A secret key will be represented by K_m, where m is some modifier, for example K_s could be a session key and K_{sAB} could be a session key shared by users A and B.

A public key is represented by KU_A for user A, and the corresponding private key is represented by KR_A for the same user.

Encryption performed with a key K_m on plaintext P will be denoted by $E_{Km}[P]$. Similarly, decryption of ciphertext C with a key K_m is will be denoted by $D_{Km}[C]$. Thus, one can write

$$D_{Km}[\ E_{Km}[P]\ ] = D_{Km}[C] = P.$$

If several fields, such as ID_A and stamp, are concatenated, it will be represented by

$$ID_A \parallel \text{stamp}.$$

If a message M is signed by X, it will be represented by

$$M \parallel E_{KRx}[\ H[\ M\ ]\ ] = KR_X\{\ M\ \},$$

and the value is equal to the original M with the hash code of M encrypted with X's private key KR_X appended at the end.

4.6 X.509 AUTHENTICATION SERVICE

ITU-T (ITU (International Telecommunication Union) Telecommunication Standardisation Sector) recommendation X.509 is part of the X.500 series of recommendations that define a directory service. The directory is a server or distributed set of servers that maintains a database of information about users [15]. The purpose of the directory service is to provide a means of authenticating entities in a computer-based environment.

The core of the X.509 scheme is the public-key certificate associated with each user. This certificate contains information about a user that a trusted authority believes is correct. In order to do this, X.509 is based on the use of public-key cryptography and digital signatures. The authority signs the certificate once it is convinced that the details contained in the certificate are accurate. The criteria that the authority uses is usually publicly available so that the level of certainty that the authority had when signing the certificate is substantiated.

4.6.1 X.509 certificates

The X.509 certificate identifies an entity together with information about that entity. Most importantly, it contains the entity's public key that can be used to verify all of the entity's signatures that were created with the entity's private key.

The general format of the X.509 certificate is shown in Figure 4.2. Version 1 contains all of the information up to the subject's public-key information, and the CA's signature. Version 2 contains all of the information up to the subject unique identifier, and CA's the signature. Version 3 contains all of the information shown.

The elements in the X.509 certificate have the following meanings [15], pp. 341 - 349:

Version: Differentiates between successive versions of the certificate format.
Serial number: An integer value that is unique within the issuing CA's certificates.
Signature algorithm identifier: The algorithm used to sign the certificate together with any associated parameters. This is repeated at the end of the certificate.
Issuer name: X.500 name of the CA that created and signed this certificate.
Period of validity: The date range over which the certificate is valid.
Subject name: The name of the user to whom the certificate refers and who holds the corresponding private key.
Subject's public-key information: The public key of the subject with the algorithm used and associated parameters.
Issuer unique identifier: Uniquely identifies the issuing CA in the event that the X.500 name has been reused for different entities.

Version
Certificate serial number
Signature algorithm identifier *(algorithm and parameters)*
Period of validity
Subject name
Subject's public-key information *(algorithms, parameters and key)*
Issuer unique identifier
Subject unique identifier
Extensions
Signature *(algorithms, parameters and signature itself)*

Figure 4.2: Content of an X.509 version 3 certificate

Subject unique identifier: Uniquely identifies the subject in the event that the X.500 name has been reused for different entities.

Extensions: Possibly information that was added in version 3 of the certificate, such as key and policy information, certificate subject and issuer attributes and certification path constraints.

Signature: The CA's signature over all other fields of the certificate. This is the hash code of the other fields and is encrypted with the CA's private key and thus cannot be duplicated by another party.

Because the signature uses the CA's private key, no other party can modify the certificate without this being detected. This allows the certificate to be publicly available with the assurance that it could have been created only by the specified CA.

Every entity trusts the CA that generated its certificate. For this to be justified, the entity must have a reliable copy of the CA's public key that has not been changed in any way. This allows the entity to verify its own certificate and the certificates of other entities that were made by the same CA.

If there is a large community of users that require X.509 certificates, it is not practical that they all subscribe to the same CA. In a situation such as this, the CAs form a hierarchy where one CA signs the certificates of the CAs immediately below it. An entity can then follow a chain of trust up the hierarchy where one CA confirms the next CA's identity. In this manner, a certificate signed by one CA can be proved correct by another CA.

4.6.2 Certificate revocation

A certificate normally has a period over which it is valid. However, it may be desirable to declare the certificate invalid (revoke the certificate) for one of the following reasons: the user's secret key is assumed or known to be compromised, the user is no longer certified by the specific CA, or the CA's certificate is assumed or known to be compromised.

To achieve the goal of revoking certificates, a Certificate Revocation List (CRL) is maintained by each CA and contains the list of certificates issued by that CA that have been revoked but not yet expired. The format of a CRL is shown in Figure 4.3.

When a user receives a certificate from any source, the user must first determine whether the certificate has been revoked before it can be used. This is normally done by maintaining a local copy of the CRL for the relevant CA.

Signature algorithm identifier *(algorithm and parameters)*
Issuer name
This update date
Next update date
Revoked certificate #1 *(certificate serial number and revocation date)*
. . .
Revoked certificate #n *(certificate serial number and revocation date)*
Signature *(algorithms, parameters and signature itself)*

Figure 4.3: Content of an X.509 CRL

4.6.3 Authentication procedures

X.509 provides three different authentication procedures that can be used whenever authentication of communicating parties is required. All three of these procedures make use of public key signatures. It is assumed that the two parties know each other's public key, either by obtaining each other's certificates from the directory or because the certificate is included in the initial message from each side.

One-way authentication involves a single transfer of information from one user (A) to another user (B). The message that is transferred is

$$KR_A\{t_A, r_A, B, sgnData, E_{KUb}[K_{ab}]\},$$

where t_A is a timestamp, r_A is a nonce (unique number used once), B is the identity of B, sgnData is some data that is signed, and $E_{KUb}[K_{ab}]$ is a session key that is signed using B's public key. This establishes the identity of A and that the message was generated by A, that the message was intended for B, and the integrity and originality of the message.

Two-way authentication involves the same transfer as above and also the message

$$KR_B\{t_B, r_B, A, r_A, sgnData, E_{KUa}[K_{ba}]\},$$

where the nonce sent by A is returned. This establishes the following in addition to the above: the identity of B and that the reply message was generated by B, that the message was intended for A, and the integrity and originality of the reply.

4.7 SECURITY FRAMEWORK AS REFERENCE

In order to compare the existing CVS and the proposed CVS in terms of security, both must be compared to known security frameworks as frames of reference. A widely accepted set of frameworks is the ITU's X.810 series of recommendations (X.810 to X.816) which are security frameworks for open systems. X.810 is an overview of the security frameworks [16].

X.811 is an authentication framework [17]. It describes in detail what is meant by authentication in an open system and it gives the following possible attacks on systems in terms of authentication:

- replay attack, where an intruder duplicates a message,
- relay attack, where an intruder passes on a message, initiated by the intruder, and
- relay attack, in which the intruder responds.

X.812 is an access control framework [18]. The primary goal of access control is to counter the threat of unauthorised operations involving a computer or communications

system. The threats are frequently subdivided into the classes of unauthorised use, disclosure, modification, destruction and denial of service.

X.813 is a non-repudiation framework [19]. It gives a number of methods by which non-repudiation services can be provided. Possible threats to non-repudiation include:

- the compromise of keys,
- the compromise of evidence, and
- the falsification of evidence.

X.814 provides a confidentiality framework [20]. In the case of confidentiality being provided through access prevention, threats can be penetration of the access prevention mechanism or penetration of the services that the access prevention mechanism relies on. When confidentiality is provided through information hiding, threats include penetration of the cryptographic mechanism or traffic analysis.

X.815 is an integrity framework [21]. In terms of the services provided in a system, integrity threats can be classified as unauthorised creation / modification / deletion / insertion / replay in environments that support data integrity through prevention. Alternatively, integrity threats can be classified as unauthorised creation / modification / deletion / insertion / replay in environments that support data integrity through detection.

X.816 is a security audit and alarms framework [22]. Some examples of security audit and alarms mechanisms are:

- comparing the activity of an entity against a known profile, e.g. unusual access based on time or geography,
- detecting the accumulation of one or several event types within some period of time, and
- observing the non-occurrence of one or several event types within some period of time.

At a later point in this document, the susceptibility of the current CVS and the proposed CVS to each of the above attacks is discussed so that the security features of the two systems can be compared.

4.8 SECURITY REQUIRED IN THE CVS

Whenever information is transmitted over digital communication links, there are possible ramifications if information becomes available to unintended parties. For example, when a party requests a token from another party, a third party can eavesdrop and obtain a copy of that token. Alternatively, a party can request information, receive a response and deny requesting the information initially and thus deny having to pay for the information. Security is required to prevent these types of scenarios.

In the on-line CVS, because customer information and requests are transmitted from the CDU to the centralised database and back, privacy over the communication link is required so that customer details are not unnecessarily disclosed.

It is very important that there be sufficient proof of financial transactions that take place. This is necessary so that the occurrence of the transaction cannot later be denied in a court of law. In order to have proof of the financial transactions, the identity of the parties involved in the communication must be proved to a high enough degree of certainty.

Also, if a party sends a request to another party, the party must not be able to deny having sent the original request. This is required so that there can be integrity in the dealings of every party and every party can be held accountable for every transaction that they make.
It is important that the records that each party holds can be audited by an independent party.

4.9 CONCLUSION

In the previous chapter, the functions that the proposed on-line CVS should provide were given. In this chapter, the basic tools to provide the necessary services were briefly

discussed. The most significant of these is public key cryptography as this is the basis for the authentication of identity, thereby proving origin and ownership of digital messages. The RSA algorithm was given together with an explanation of how it works.

The notation that is necessary for describing the cryptographic services was given in this chapter. This will be used in later chapters when the content of messages is given. The X.509 authentication service was briefly described, including X.509 certificates. Authentication procedures were also briefly described. These make use of the public key cryptography so that digital signatures can be implemented.

The X.810 security frameworks were briefly described so that a standard list of attacks is available against which the current and the proposed CVS can be compared.

The security that is required in the proposed on-line CVS was also described. This will be referred to later when motivating design choices.

In the next chapter, various security protocols are examined so that an appropriate combination can be selected to provide the required security services to acceptable levels of certainty.

Five

RELEVANT SECURITY PROTOCOLS

5.1 INTRODUCTION

A means is required by which information can be conveyed between the CDU and centralised database securely and reliably. The transfer must be secure in that it is not modified in any way and that fabricated information is not accepted as valid. The transfer must be reliable in that all information transmitted is received, and proof must be available to the sender that the information was in fact received.

To this end, a communication link is required between every CDU and the TM. This bi-directional communication link must be available every time a transaction occurs. Due to the ubiquity of IP, it can be used to provide the necessary communication. However, IP provides no security directly – it only provides a means of sending a packet of information from a source to a destination. A means of providing security must be used in conjunction with IP to provide the required communication services.

The previous chapter briefly described the relevant security building blocks. This chapter examines protocols that use these building blocks to achieve security services. The proposed on-line CVS requires confidentiality, authentication, integrity, non-repudiation, access control and availability. In this chapter, SSL (Secure Sockets Layer) and XML (eXtensible Markup Language) / SOAP (Simple Object Access Protocol) will be examined to determine which provides the required security services. A brief look is also taken at the relationship between security algorithms and protocols, and the languages used to implement them.

The selected security system must be used to protect the information that is transmitted between the CDU and the vending server. It must ensure that information transmitted is not modified so that owners of CDUs can be sure that they will not be overcharged for tokens sold. Customers buying tokens must be sure that the tokens received represent the value

that was paid for and will work with the corresponding ED. The security system must also ensure privacy so that customer information is not available outside the system.

5.2 SECURE SOCKETS LAYER

SSL provides point-to-point security between a Client and a Server, connected by means of an insecure public network [23]. Specifically, SSL is designed to make use of the TCP (Transmission Control Protocol) / IP protocol stack to provide a reliable end-to-end secure service. SSL consists of two layers of protocols and is positioned immediately above the TCP layer in the TCP/IP model, as illustrated in Figure 5.1. The SSL Record Protocol provides basic security services to the higher layer protocols. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, the Change Cipher Spec Protocol, and the Alert Protocol. These SSL-specific protocols are used in the management of SSL exchanges.

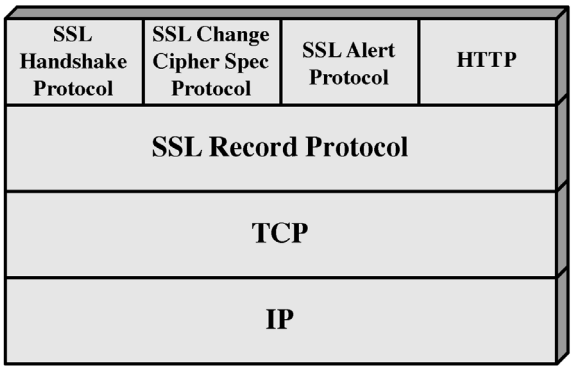

Figure 5.1: SSL protocol stack ("Cryptography and network security", 2nd edition, Stallings, Figure 14.2, p. 444 [12])

Netscape developed SSL. The original version was SSL 2.0, which contained a number of security flaws. Subsequently, Version 3 was designed with input from industry and public review to resolve these flaws. It was then published as an Internet draft document. Subsequently, the Transport Layer Security (TLS) working group was formed within the IETF (Internet Engineering Task Force) to develop a common standard.

SSL provides confidentiality, entity and data authentication and integrity of information transmitted over a connection-orientated TCP/IP link. It is possible for SSL to provide security for any connection that operates over a TCP/IP link. SSL uses public key cryptography for the authentication of servers and clients, and requires a CA to provide authentication services. SSL is located immediately above the TCP/IP layers and below the application layer. This means that application level protocols such as HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), Telnet, etc. requiring secure communications can operate as usual, since the SSL layer will transparently provide the required security facilities.

Two important SSL concepts are the SSL session and the SSL connection, which are defined in the specification as follows:

Connection: A connection is a transport (in the OSI (Open Systems Interconnection) layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

Session: An SSL session is an association between a client and a server. Sessions are created by means of the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

5.2.1 Record protocol

In normal operation, SSL makes use of the Record Protocol. The SSL Record Protocol provides two services for SSL connections:

- *Confidentiality:* The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.
- *Message Integrity:* The Handshake Protocol also defines a shared secret key that is used to form a MAC.

Figure 5.2 shows the overall operation of the SSL Record Protocol. The secure record protocol operates as follows:

- The application layer data is fragmented into manageable blocks.
- The first fragment is optionally compressed.
- To ensure integrity of the fragment, a MAC is computed using the negotiated algorithm and cryptographic keys. The MAC is then attached to the fragment.
- To ensure confidentiality of the fragment and MAC, it is encrypted using the negotiated symmetric encryption algorithm and cryptographic keys.
- The SSL record header is attached to encrypted data, and transmits the resulting unit in a TCP segment.
- The process is repeated for all remaining fragments.

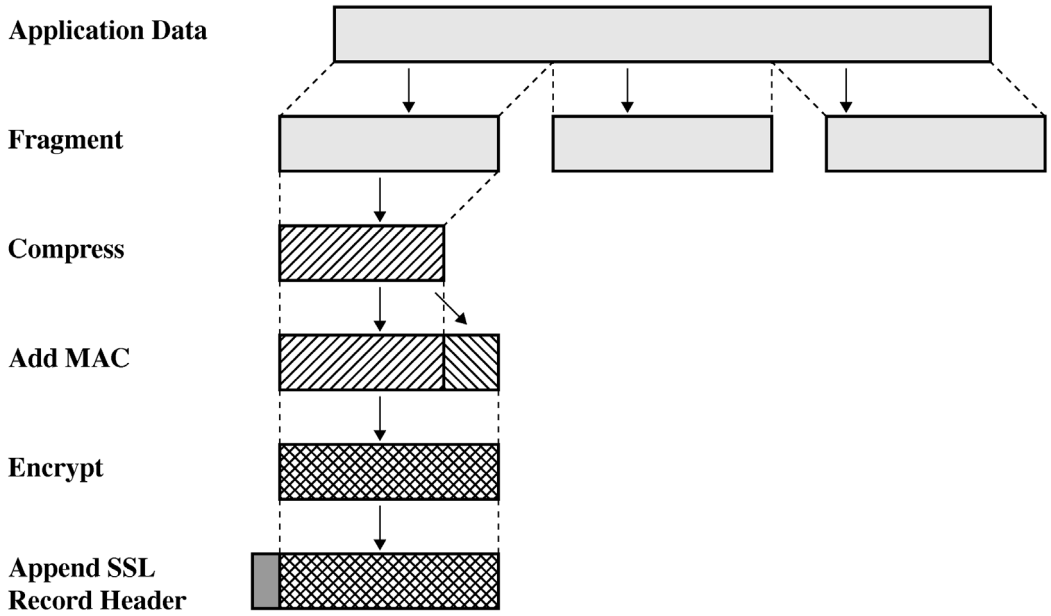

Figure 5.2: SSL record protocol operation ("Cryptography and network security", 2nd edition, Stallings, Figure 14.3, p. 446 [12])

The MAC is calculated using either MD5 or SHA-1. This prevents the message from being modified without being detected. Symmetric encryption is used, and may be one of IDEA

(International Data Encryption Algorithm), RC2-40, DES, DES-40, Triple DES, Fortezza, RC4-40 or RC4-128.

5.2.2 Handshake Protocol

The security parameters, such as MAC and encryption algorithm, need to be negotiated before they can be used. This is done by the Handshake Protocol and allows the server and client to authenticate each other before exchanging the security parameters for the connection. The Handshake Protocol is used before any application data is transmitted. It consists of four phases: establish security capabilities, server authentication and key exchange, client authentication and key exchange, and finish.

The Handshake Protocol consists of a series of messages exchanged by client and server. Figure 5.3 shows the initial exchange needed to establish a logical connection between client and server.

The Handshake Protocol is used to negotiate the protection algorithms that are used to authenticate the client and server to each other, to transmit required public key certificates and to establish the session keys for use in the integrity-check and encryption processes of the Record Protocol. Once the Handshake Protocol has been successfully completed, the Record Protocol allows for the actual communication of information using the parameters agreed upon by the Handshake Protocol.

The SSL Handshake Protocol introduces a significant amount of overhead [24]. In experiments done on web servers, using SSL to secure connections resulted in the number of web requests serviced on given hardware decreasing by a factor of 100. This is due to the overhead that the Handshake Protocol introduces. This bad performance, however, can be vastly improved if sessions can be reused, thus avoiding a large part of the Handshake Protocol overhead.

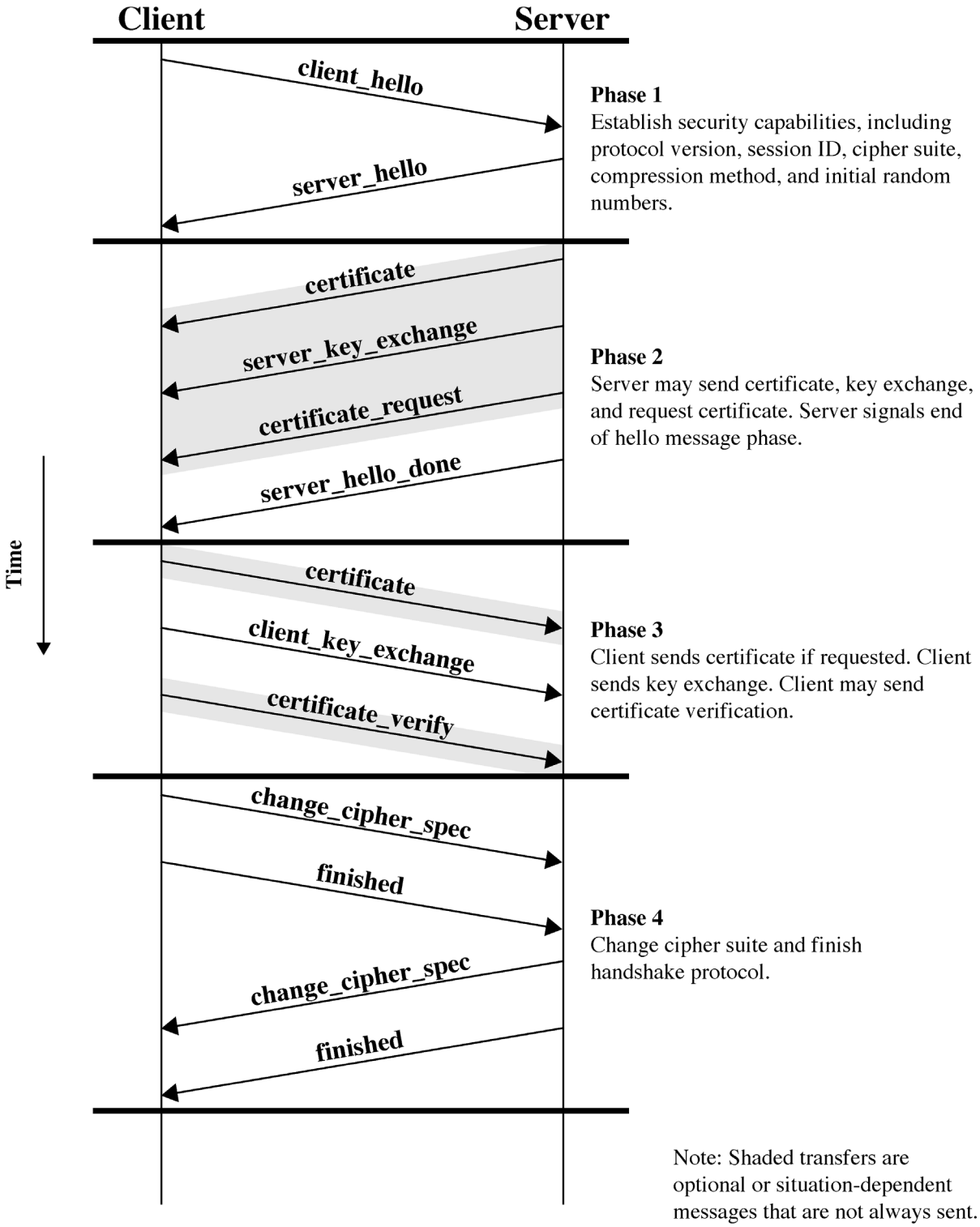

Figure 5.3: Handshake Protocol action ("Cryptography and network security", 2nd edition, Stallings, Figure 14.6, p. 451 [12])

5.2.3 Advantages

The advantages of SSL for the on-line CVS:

- Required confidentiality, authenticity and integrity of traffic are provided.
- Authentication of communicating parties is performed.
- SSL has been proved to be effective in providing security services in the Internet.

5.2.4 Disadvantages

The disadvantages of SSL:

- Communication is state based (both parties involved need to record details relating to the communication). This is unnecessary waste of storage at both sides of the connection.
- All data is unnecessarily encrypted as not all information is sensitive enough to warrant this.
- A significant amount of time and processing power are wasted in setting up and using SSL connections.
- There is no inherent form of non-repudiation that can later be used as proof, as SSL replaces the entire communication channel with a tunnel.

5.3 XML/SOAP

5.3.1 XML

XML is a subset of Standard Generalised Markup Language (SGML). The goal of XML is to enable generic SGML to be served, received and processed on the Web in the way that is now possible with HTML (HyperText Markup Language). To this end, XML was designed for ease of implementation and for interoperability with both SGML and HTML.

XML is a text-based public markup language standard, that is generally used for specifying and exchanging self-descriptive data [25]. An XML document is made up of storage units called entities, where each entity contains a description of the purpose of a piece of data,

and the data itself. The standard has been developed and is maintained by the World Wide Web Consortium (W3C), and is available at [25]. It has been widely accepted in industry [26]. The current version of XML is version 1.0, second edition (combination of first edition and errata).

XML documents are made up of storage units called entities, which contain either parsed or unparsed data. Parsed data is data that is formatted according to the XML specification. Parsed data is made up of characters, some of which form character data and some of which form markup. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

The design goals for XML [25] are:

- XML shall be straightforwardly usable over the Internet,
- XML shall support a wide variety of applications,
- XML shall be compatible with SGML,
- it shall be easy to write programs that process XML documents,
- the number of optional features in XML is to be kept to an absolute minimum, ideally zero,
- XML documents should be human-legible and reasonably clear,
- the XML design should be prepared quickly,
- the design of XML shall be formal and concise,
- XML documents shall be easy to create, and
- terseness in XML markup is of minimal importance.

An XML document consists primarily of Start-Tags, End-Tags and attributes. Attributes are used to associate name-value pairs with elements. Examples of XML documents are given in Figures 5.4 and 5.5.

A feature of XML is that it makes provision for the identification of the natural language in which the content of the document is written by means of the xml:lang attribute. This enhances portability between various locales.

XML also provides support for X.509 certificates by means of an X509DataElement. X.509 certificates are a widely used format for public-key certificates. They allow the identity of an entity to be proved and make the public key for that entity available.

XML documents can either be partially or wholly encrypted by means of triple DES, AES-128 (Advanced Encryption Standard – 128 bit) or AES-256 [27]. (Partially encrypting data is beneficial when not all data is confidential and the devices that encrypt and decrypt the messages have limited computing power.) Encryption is based on the namespace

xmlns:xenc='http://www.w3.org/2001/04/xmlenc#'.

An example of an XML document containing encrypted data is shown in Figure 5.4. In this example line s1 indicates that encryption is being used. Line s2 indicates that triple DES is being used in Cipher Block Chaining mode. Lines s3 to s5 indicate the key that should be used for the decryption, and finally line s6 contains the encrypted data itself.

```
[s1] <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
        Type='http://www.w3.org/2001/04/xmlenc#Element'/>
[s2]   <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#tripledes-cbc'/>
[s3]   <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
[s4]     <ds:KeyName>John Smith</ds:KeyName>
[s5]   </ds:KeyInfo>
[s6]   <CipherData><CipherValue>DEADBEEF</CipherValue></CipherData>
[s7] </EncryptedData>
```

Figure 5.4: An example of encrypted data with a symmetric key (taken from [28])

Typically not all information that is transmitted is confidential – only parts of messages are required to be kept confidential. XML documents that contain encrypted data can either be partially encrypted or completely encrypted. Because encryption requires significant

mathematical processing power, partially encrypting messages that are sent can significantly reduce processing requirements.

The encryption algorithms that must be supported by all implementations of XML are triple DES, AES-128 and AES-256.

An XML signature, known as XML DSig, is based on the namespace [29]

xmlns:ds='http://www.w3.org/2000/09/xmldsig#'.

An example of a detached signature (an item outside the current item is signed) is shown in Figure 5.5. The digest is calculated using SHA1. The signature is calculated using DSA.

```
[s01] <Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
[s02]   <SignedInfo>
[s03]   <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
          c14n-20010315"/>
[s04]   <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
[s05]   <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
[s06]     <Transforms>
[s07]       <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
[s08]     </Transforms>
[s09]     <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
[s10]     <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
[s11]   </Reference>
[s12] </SignedInfo>
[s13]   <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
[s14]   <KeyInfo>
[s15a]    <KeyValue>
[s15b]      <DSAKeyValue>
[s15c]        <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
[s15d]      </DSAKeyValue>
[s15e]    </KeyValue>
[s16]   </KeyInfo>
[s17] </Signature>
```

Figure 5.5: An example of a detached signature of the content of the HTML4 in XML specification (taken from [29])

5.3.2 SOAP

SOAP provides the definition of XML-based information that can be used for exchanging structured and typed information between peers in a decentralised, distributed environment [27], [30]. A SOAP message is a one-way transmission. Multiple SOAP messages can be combined to create conversational exchanges.

SOAP envelopes and messages can be represented in XML 1.0 documents. A SOAP message is fundamentally a one-way transmission between SOAP nodes, from a SOAP sender to a SOAP receiver. SOAP messages can be combined by applications to implement more complex interaction patterns ranging from request / response to multiple, back-and-forth "conversational" exchanges. Various usage scenarios are given in [30].

An example of a travel reservation is given in Figure 5.6. As seen from this figure, it is easy to understand the format of the message and the data contained within the message. Since the entire message is text based, the format is extremely portable across various hardware and software platforms.

SOAP is a public standard for XML-based messaging. Like XML itself, SOAP is also maintained by the W3C and is available at [30]. SOAP has rapidly gained industry-wide acceptance as the standard way of doing business-to-business integration. It is supported by a large part of the IT (Information Technology) industry. The two dominating enterprise architectures, J2EE (Java 2 Enterprise Edition) and Microsoft .Net, both have extensive support for SOAP messaging.

SOAP allows information to be encrypted and digitally signed. This is done by means of XKMS (XML Key Management Specification). XKMS defines a web service interface for a public-key infrastructure to manage keys for use with protocols like XML DSig, which was mentioned previously [31]. XKMS contains two sub-protocols: XML Key Information Service Specification (X-KISS) and XML Key Registration Service Specification. X-KISS locates and retrieves public keys from a key server to be used in, for example, encryption or signature verification. An application can also use X-KISS to verify that a certain key

has not been revoked. X-KISS defines service interfaces for registering, revoking, and recovering escrowed keys from a key server.

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
 <env:Header>
  <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
          env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
           env:mustUnderstand="true">
   <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
   <m:dateAndTime>2001-11-29T13:20:00.000-05:00</m:dateAndTime>
  </m:reservation>
  <n:passenger xmlns:n="http://mycompany.example.com/employees"
          env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
           env:mustUnderstand="true">
   <n:name>Åke Jógvan Øyvind</n:name>
  </n:passenger>
 </env:Header>
 <env:Body>
  <p:itinerary
    xmlns:p="http://travelcompany.example.org/reservation/travel">
   <p:departure>
     <p:departing>New York</p:departing>
     <p:arriving>Los Angeles</p:arriving>
     <p:departureDate>2001-12-14</p:departureDate>
     <p:departureTime>late afternoon</p:departureTime>
     <p:seatPreference>aisle</p:seatPreference>
   </p:departure>
   <p:return>
     <p:departing>Los Angeles</p:departing>
     <p:arriving>New York</p:arriving>
     <p:departureDate>2001-12-20</p:departureDate>
     <p:departureTime>mid-morning</p:departureTime>
     <p:seatPreference/>
   </p:return>
  </p:itinerary>
  <q:lodging
   xmlns:q="http://travelcompany.example.org/reservation/hotels">
   <q:preference>none</q:preference>
  </q:lodging>
 </env:Body>
</env:Envelope>
```

Figure 5.6: Travel reservation expressed as a SOAP message (taken from [30])

5.3.3 Advantages

The advantages of XML/SOAP for the on-line CVS requirements:

- XML/SOAP is text based. This allows it to be transported over simple network layers such as HTTP and to be supported by virtually all hardware and software platforms.
- XML/SOAP is widely supported in industry. This means that it will be easy and cost-effective to obtain software that implements the message protocol.
- XML/SOAP documents have a hierarchical structure that allows the logical structuring of information.
- Encryption and digital signatures can be designed to apply to only parts of the data contained in messages. This allows a trade-off between confidentiality and processing power required where only sensitive information can be encrypted.

5.3.4 Disadvantages

The disadvantage of XML/SOAP:

- XML/SOAP is not as compact as other binary protocols due to its text-based nature. This means that communication resources are used slightly more than would be necessary if a protocol were optimised to use resources as efficiently as possible.

5.4 IMPLEMENTING SECURITY FUNCTIONALITY

There are a number of ways in which the security mechanisms described in the previous chapter and the security protocols described in this chapter can be implemented. Almost any programming language can be used to implement security functionality if done in a careful security-conscious manner.

One example of a programming language that already has a large pre-implemented security component is Java [32]. Java has support for all of the cryptographic algorithms that have been described in the previous pages, and it has support for secure communication

protocols, such as SSL. Java also has support for various PKIs, including the X.509 directory authentication framework.

The important point to note when considering security and the implementation of security is that, regardless of the programming environment that is used to implement security, the correct algorithms and protocols for the application must be selected. The security in any system is only as strong as the weakest point. Thus, it is critical that suitable security algorithms and protocols are selected for a particular system. It is also critical that the implementation of the algorithms and protocols be done in a secure manner. Because of this, the selection of security algorithms and protocols should be done independently of the implementation.

5.5 CONCLUSION

In this chapter, SSL and XML/SOAP were examined to determine their suitability for implementing the required security services for proposed on-line CVS. The comparison of the protocols in this chapter shows that these requirements are met by XML/SOAP. SSL does not allow for non-repudiation of originator or the message itself.

XML/SOAP is text-based, widely supported by industry, has a hierarchical structure, supports encryption, and supports non-repudiation by means of digital signatures. The text-based aspect of XML/SOAP is its biggest advantage and downfall. Text-based communication is advantageous since it can be transported by any type of network infrastructure. However, text-based communication unnecessarily wastes communication bandwidth by generally increasing the size of messages. Note that only the size of the messages is increased; the number of messages remains the same. This text-based disadvantage is minor in comparison with the advantages, and thus XML/SOAP is the best means of implementing the required security functionality of the protocols discussed.

At the end of the previous chapter, the security requirements for the system at hand were given. These are met using XML/SOAP with encryption and digital signatures as follows:

The *confidentiality* requirement is met by encrypting the applicable parts of the messages. This is done by using the secret key shared between the two communicating parties (see the next chapter).

A PKI is required to handle the creation and distribution of public and private keys so that the above confidentiality can be addressed. This is analysed in a later chapter.

The *authentication* function is implemented by means of digital signatures. XML provides this function as specified by the xmlns:ds='http://www.w3.org/2000/09/xmldsig#' namespace.

The *integrity* and *non-repudiation* functions are implemented by means of digital signatures.

Access control and *availability* are provided by several layers of protection. The first is a fire wall that protects the TMs so that only data that comes from valid places on the Internet will be entertained. Secondly, only messages that have valid digital signatures will be processed. These measures constitute the necessary access control. Availability will be guaranteed because of the access control measures and having enough redundancy of the TMs that are capable of handling at least twice the expected continuous load.

The final point that was considered in this chapter is the relationship between the security algorithms and protocols and the implementation of these. The selection of the algorithms and protocols and the implementation must be independent of one another.

Six

SYSTEM SECURITY IN THE PROPOSED SYSTEM

6.1 INTRODUCTION

In this chapter, the security that is used in the current CVS will be examined so that it can be determined how it must be changed to meet the security requirements of the proposed CVS. The security that is employed in the proposed system is crucial for ensuring that the necessary security services are provided. For this reason, this chapter examines the flow of keys in the current CVS and migrates to the flow of keys in the proposed CVS. The messages that are focused on are those that convey the keys within the system as any security system is only as secure as its weakest component.

A crucial part of a security system is the keys and how the keys are distributed. The key hierarchy is given and the purposes of the keys are explained. Closely related to the keys is the notion of the identity of the various communicating entities. This leads to the motivation of the requirement of authentication.

The interconnection between the various system components is examined from a security point of view. This is done so that the vulnerable entities can be secured using the necessary precautions.

The algorithms that are used in the proposed system are given together with a motivation for why they were selected.

The vulnerabilities that the system has as a result of its electronic on-line structure are briefly described so that the necessary preventative measures can be taken. The measures that were taken are described in the following two chapters.

6.2 KEY FLOW IN THE EXISTING SYSTEM

The management of keys in the currently deployed CVS is depicted in Figure 6.1.

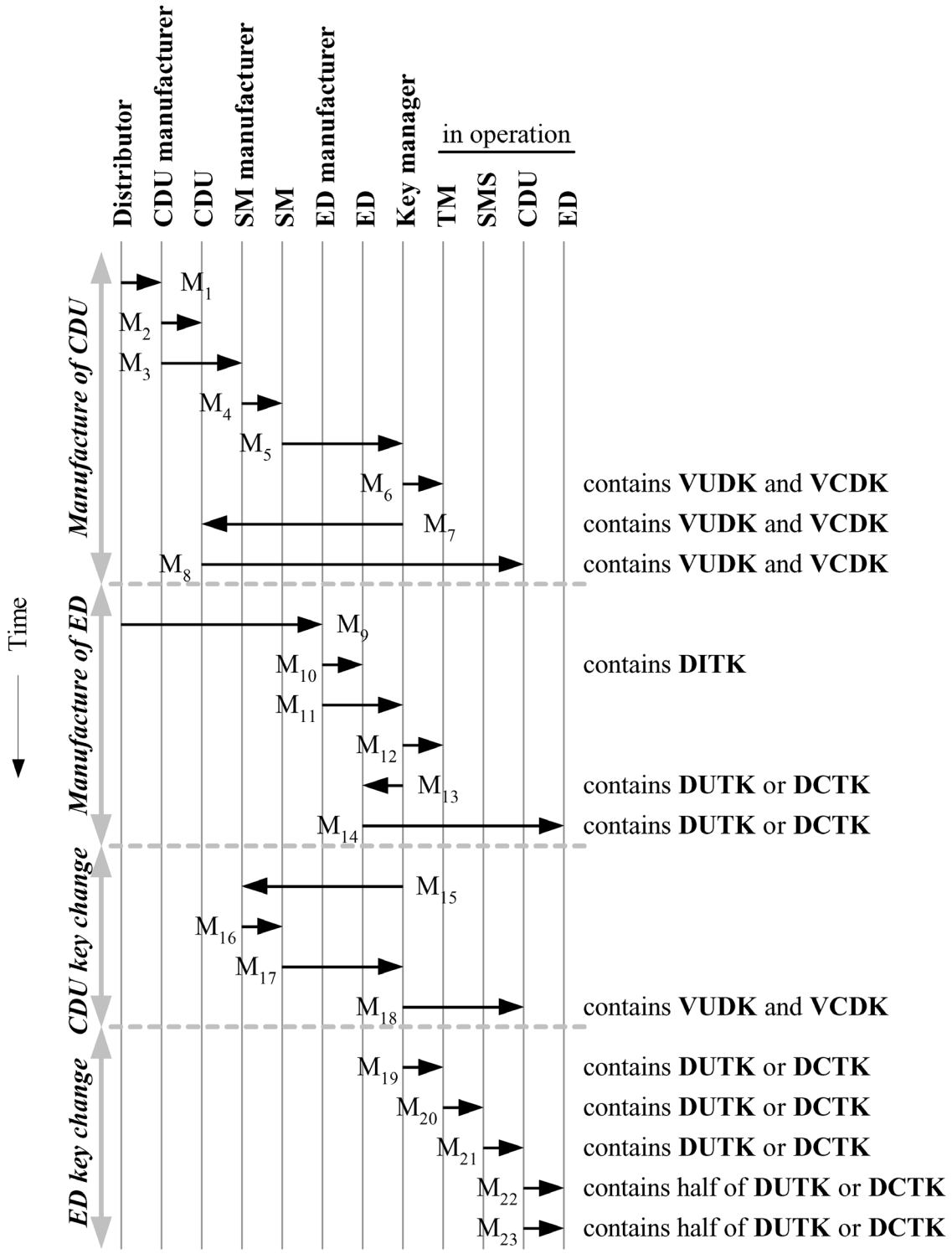

Figure 6.1: Messages supporting the movement of keys during manufacture and field update in the current CVS

As previously mentioned, this system was designed in 1990. At that time there was no communication infrastructure such as is available today with telecommunication facilities and the Internet. In order to generate a token at the site of a sale, some device was needed that had this ability and was secure. This is why an SM is required at the CDU sites in the current CVS.

During the manufacture of a CDU in the current CVS (M_1 to M_8 in Figure 6.1), the Key manager (normally appointed by and acting on behalf of the utility) places the necessary information in the SMs and these in turn are installed in the CDUs. The combination of CDU and SM allows tokens to be generated on request and the transaction information is passed on to the TM via an SMS.

During the manufacture of an ED (M_9 to M_{14} in Figure 6.1), the ED is given its initial key, DUTK or DCTK as appropriate, by the key manager before being deployed in the field. This key depends on the supply group to which the ED belongs as well as the ED's serial number.

The cryptographic core of a CDU is the SM. The SM has the ability to generate tokens given the value of the token required and the serial number of the ED for which the token is to be used. The SM is thus a valuable and precious entity that must be carefully guarded against theft. In the currently deployed CVS, should the identity of a CDU need to be changed, the SM must be replaced (M_{15} to M_{18} in Figure 6.1). Changing the identity of a CDU is not a function that was originally catered for.

The key used by an ED, DUTK or DCTK, must change periodically for security reasons – the tokens used by EDs are available encrypted, and too many of these encrypted with the same key will make cracking the encryption algorithm and keys too easy. The ED's key is changed periodically (M_{19} to M_{23} in Figure 6.1). The key change is merely the new key passing from Key manager to TM, to SMS, to CDU and finally to ED via two tokens (one token does not contain enough data for an entire new key).

As listed in Table 3.1 in Chapter 3, this system is vulnerable to attacks on confidentiality, authentication, integrity, non-repudiation, access control and availability. For this reason, improved handling of security and keys is required.

6.3 CVS KEY HIERARCHY

The key hierarchy in the proposed CVS was implemented as shown in Figure 6.2. The master key pair, consisting of the Master pRivate Key (MRK) and the Master pUblic Key (MUK), that each CDU and TM has, is unique to that entity and should never need to be changed.

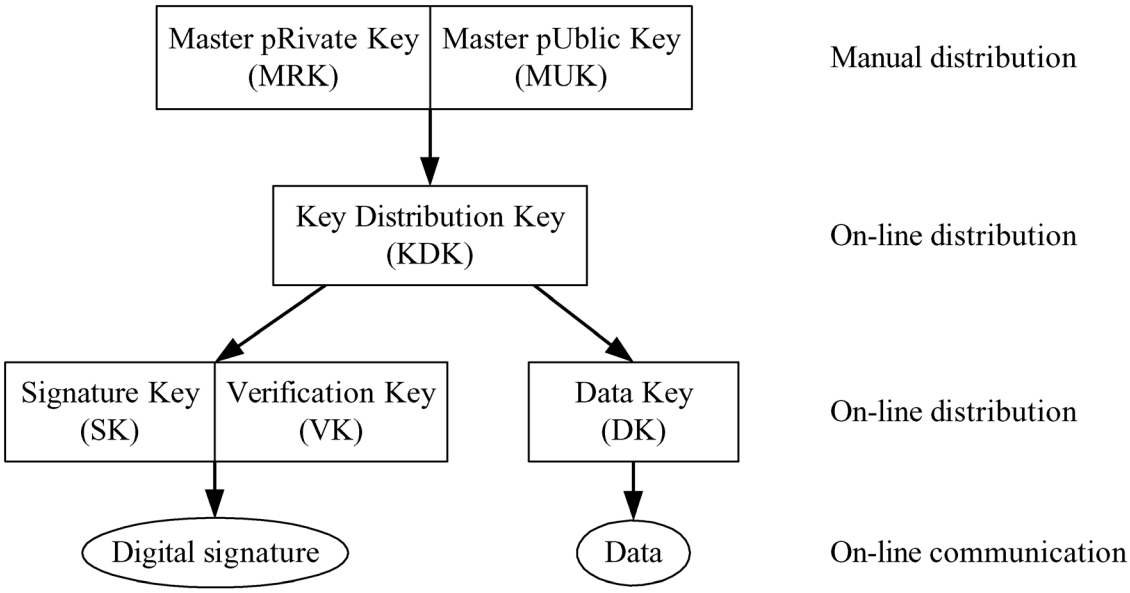

Figure 6.2: The key hierarchy of the proposed CVS, used by both the CDUs and the TMs

Every CDU and TM has one of each of the types of keys indicated in Figure 6.2 allocated to it. The keys change over time (explained below), but at any instant in time, there is one of each of the types of keys allocated to every CDU and TM.

Because the CDUs are exposed to the Internet, the usage of a specific key for encrypting data that traverses the Internet must be limited. This is done in order to limit the amount of

ciphertext that is publicly accessible to make a ciphertext attack infeasible. Thus, the period that a specific key is valid for is a function of how often the key is used.

Every CDU and TM has a master key pair consisting of a private part (MRK) and a public part (MUK). This key pair is rarely used as it is needed only to encrypt and decrypt Key Distribution Keys (KDKs) (the next item in the key hierarchy). It is rarely used for two reasons. Firstly, public key cryptography is computationally expensive and so should be used as seldom as possible. Secondly, limiting the number of times that a key is used limits the samples of ciphertext that are available to be analysed by an attacker and thus makes attacks more difficult. The master key pair is linked to the serial number of the entity with which it is associated, and this link will normally not be broken.

The KDK is used to encrypt Data Keys (DKs) and signing key pairs only. To limit the quantity of ciphertext that is available for attack, this key will only be used for a maximum of 1000 transactions or 3 months. After this it must be replaced.

The signing key pair has a private component, the Signature Key (SK), and a public component, the Verification Key (VK). The SK is used by the entity that owns it to sign data digitally for non-repudiation and authentication purposes. The VK is used by other parties to verify the signature. The VK is available in the form of a certificate from the CA for whoever needs it to prove that a particular CDU or TM created a message. The SK may only be used to sign a maximum of 10000 transactions or be used for 1 week. After this, it must be replaced with a new key assigned by the KMC. The VK, however, must be available indefinitely so that signatures created with the corresponding SK can be verified.

Finally, the DK is used for encrypting the data of transactions. The DK may only be used for a maximum of 10000 transactions or 1 week. After this, it must be replaced with a new key from the KMC. When it is replaced, the KMC encrypts the new DK with the KEK to transport it safely to the CDU or TM.

6.4 KEY FLOW IN THE PROPOSED SYSTEM

In the proposed system, each CDU and TM has a unique identity that can be regarded as a serial number. This is set at the time of manufacture and cannot be changed. Each CDU must be able to communicate with every TM over a secure channel, and thus the communicating CDU and TM must share a secret key. Secret key cryptography was chosen during the design of the system for encrypting normal transactions, because it is significantly more efficient than public key cryptography for a comparable level of security.

The use of a combination of public and secret key cryptography allows the management of keys in the proposed CVS to be handled as shown in Figure 6.3. The changes in communicating entities, as compared to the current CVS, is that the SM no longer exists, and the concept of the SMS is no longer necessary (the CDUs now communicate directly with the TM).

Because public key cryptography is required for the services above to be provided, the Key manager takes up the role of CA, while still acting as the KMC.

During the manufacture of a CDU, a unique identity with corresponding private and public keys is required so that identity can be proved. M_3 is the request for this information to be generated by the Key manager. M_4 is the public information, MUK, being passed to the TM. M_5 is the private part of this information, MRK, being installed on the CDU.

The manufacture of an ED has exactly the same procedures as in the currently deployed system, since the STS works satisfactorily in practice.

A significant advantage of the proposed system is that most of the keys on the CDU can be changed without physically having to replace a part of the CDU. This is indicated by messages M_{13} and M_{14} in Figure 6.3.

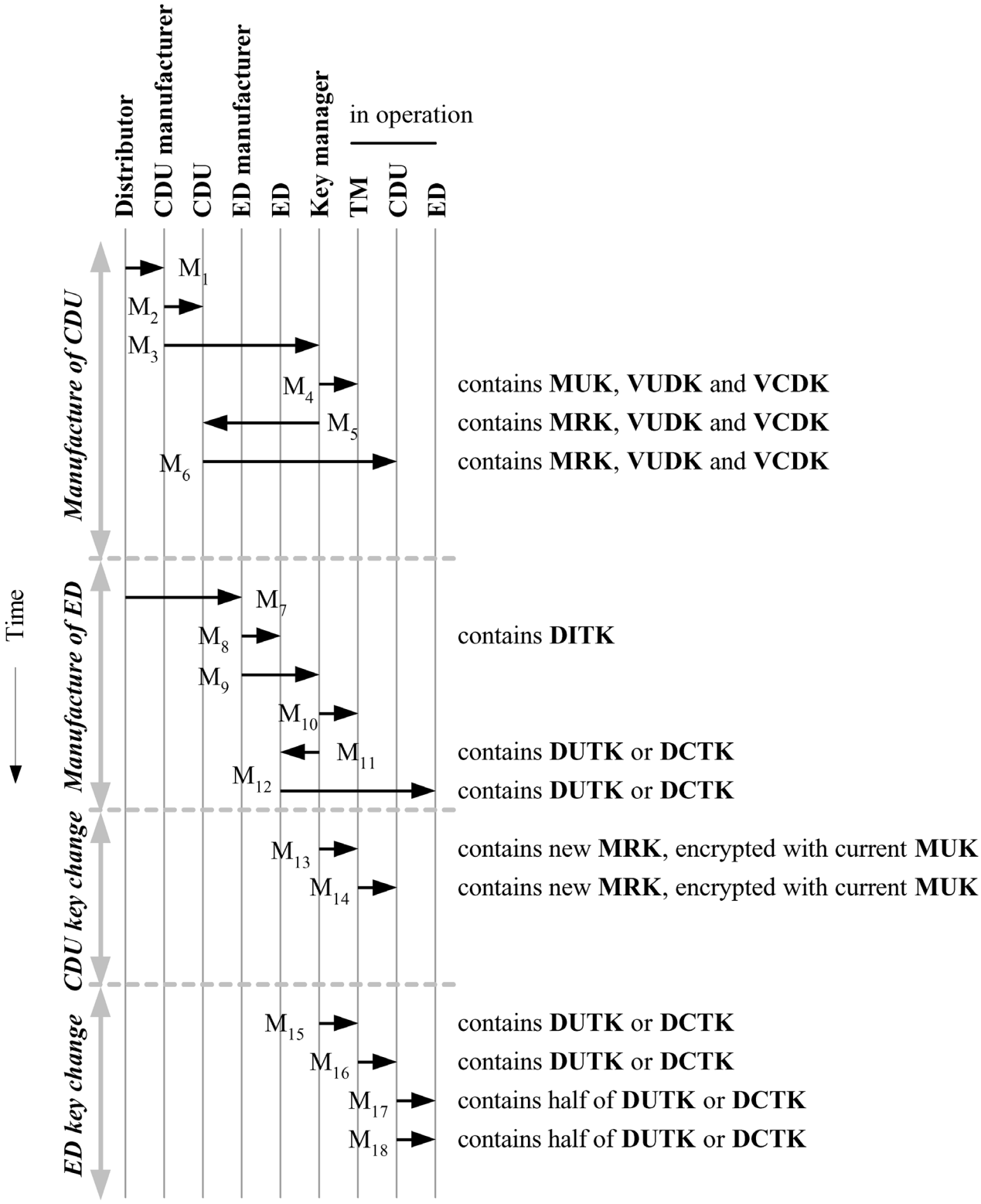

Figure 6.3: Messages supporting the movement of keys during manufacture and update in the field in the proposed CVS

The change of a key in an ED follows a similar procedure to that in the current system except that the request no longer goes through an SMS. The key change of an ED is represented by M_{15} to M_{18} in Figure 6.3.

6.5 SYSTEM INTERCONNECTION

The architecture of the proposed system is shown in Figure 6.4. The TMs have high bandwidth, high availability connections to the CA and the KMC. These entities are in a secure environment protected from the open Internet by a firewall. There are multiple TMs to ensure reliability and availability so that the load of all of the CDUs in the field can be serviced. There are multiple CAs and KMCs to ensure reliability and availability.

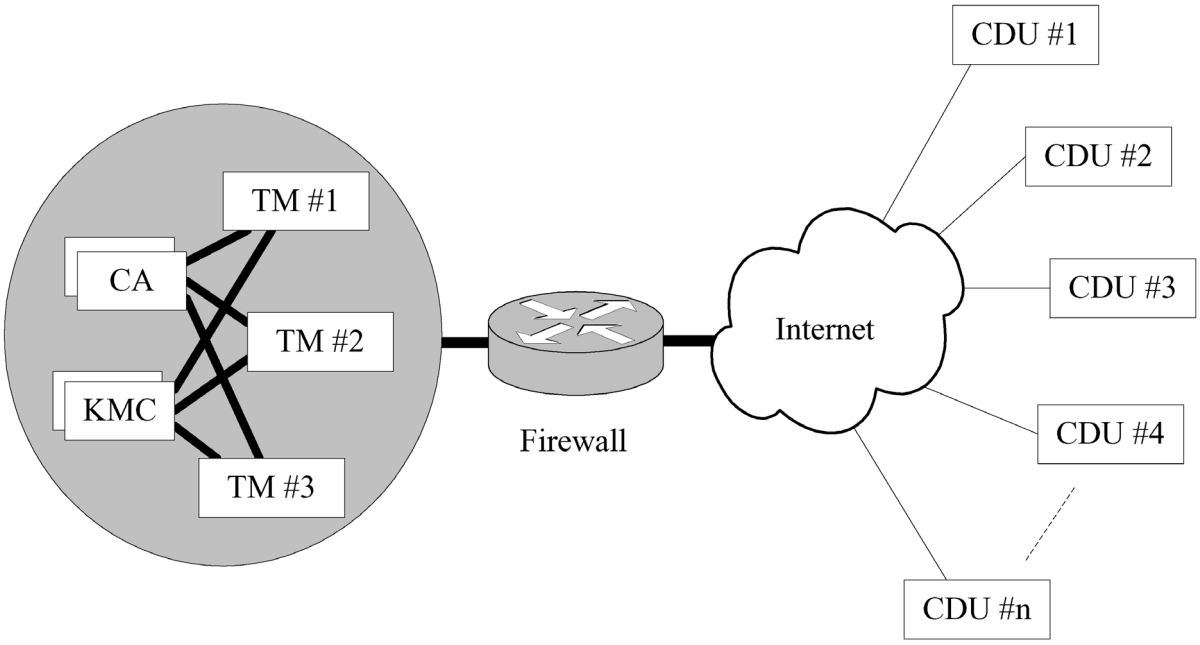

Figure 6.4: Interconnection between CDUs, TMs, the KMC and the CA

The CDUs, however, are exposed to the Internet. This is why it is critical that the keys used for communication between the CDUs and the other entities in the system be renewed on a regular basis.

The firewall protects the availability of the TMs by preventing unsolicited traffic from reaching the TMs from the Internet. This firewall includes the ability to verify that the content of the messages is meaningful and that the messages have valid signatures.

The purpose of the CA is to allow a TM or a CDU to prove it's identity. The purpose of the KMC is to generate new keys as they are needed.

6.6 AUTHENTICATION

The most important motivation for authentication in this system is so that financial accountability can be maintained; there must be sufficient evidence to prove that the organisation in control of a specific CDU is responsible for the tokens that were sent to that CDU.

Each CDU owns a public / private master key pair (MUK / MRK). This is the foundation for proving the identity of the CDU and so must be replaced as soon as it becomes known that the key pair has become compromised, although this should not normally happen. The CDU is programmed with this key pair manually – there must be no possibility that the wrong private key was given to a specific CDU and there must be no possibility of any other entity finding out what the value of a particular CDU's private key is. The CA has the corresponding public key, and so the KMC can generate a new KDK and encrypt it with the destination's MUK and send it to the CDU. The CDU is the only entity capable of decrypting the packet, using its MRK, as no other party has the corresponding private key. This is sufficient to prove that the correct party received the key as only the destined party will be able to decrypt it. This provides entity authentication.

However, it is important that the source of new keys is also authenticated – origin authentication. This can be done by signing the key change data sent to the specific CDU with the KMC's private key. Then any party can verify the authenticity of the message, and in so doing be certain that the source of the message was from the KMC and not some fake source.

The format and contents of the messages that are sent and received are given and explained in Chapter 8.

6.7 ALGORITHMS

In STS, DES (with a 56 bit key length) was originally used for the encryption of tokens. This is currently in the process of being upgraded to 3DES (with a 112 bit key length) because a key length of 56 bits is no longer regarded as being secure for symmetric cryptography.

In the proposed CVS, there are four different kinds of encryption algorithms required – normal purpose and extra secure versions of both secret key and public key algorithms. The normal purpose versions are required for the low risk (and consequently low loss if compromised) applications such as encrypting the request for a single token. On the other hand, higher security is required for the keys that have large financial implications if compromised, for example the MRK of a CDU or even the MRK of the CA or KMC. The cost of using an extra secure version of cryptography is that the computations are significantly more intensive. The advantage of using two different cryptographic strengths is thus that less computational power is required where the lower level of security is needed.

For the symmetric key cryptography, RC5-CBC (Rivest Cipher #5 – Cipher Block Chaining) is used as it was designed for the following characteristics:

- suitable for hardware or software implementation,
- fast (word oriented),
- variable length key,
- low memory requirement, and
- high security.

For the public key cryptography, RSA is used. RSA is currently widely used on the Internet for public key certificates. For the generation of a hash to be used in the

calculation of a digital signature, SHA-1 is used. It is also currently widely used on the Internet for public key certificates, and produces a 160-bit message digest.

The length of the keys used is determined by the amount of security required. RSA Security recommends that, for data that should be secure until 2015, 80 bit symmetric or 1024 bit public key cryptography is sufficient. For data that should be secure until 2035, 112 bit symmetric or 2048 bit public key cryptography will be sufficient [33]. Thus, the security algorithms with relevant key lengths that were used are indicated in Table 6.1.

Table 6.1: Encryption algorithms used for the proposed system

Purpose	Algorithm	Keys used
Protection of normal data used for transactions	RC5-CBC with 80 bit key	DK
Protection of key data used to communicate keys	RC5-CBC with 112 bit key	KDK
Digital signing of individual transactions	RSA with 1024 bit key	SK / VK
Encryption of the root of all keys	RSA with 2048 bit key	MRK / MUK

6.8 PREPARATION OF EACH MESSAGE

The process that the data goes through in preparation for transmission is shown in Figure 6.5. This is similar to the SSL record protocol (outlined in Chapter 4), but the operations not necessary in the CVS have been removed.

Once the data of the message itself has been determined (see Chapter 8), a digital signature is added to the message based on the entire message so it can be proved that the sending party did indeed send the message. This is important for authentication purposes as indicated above.

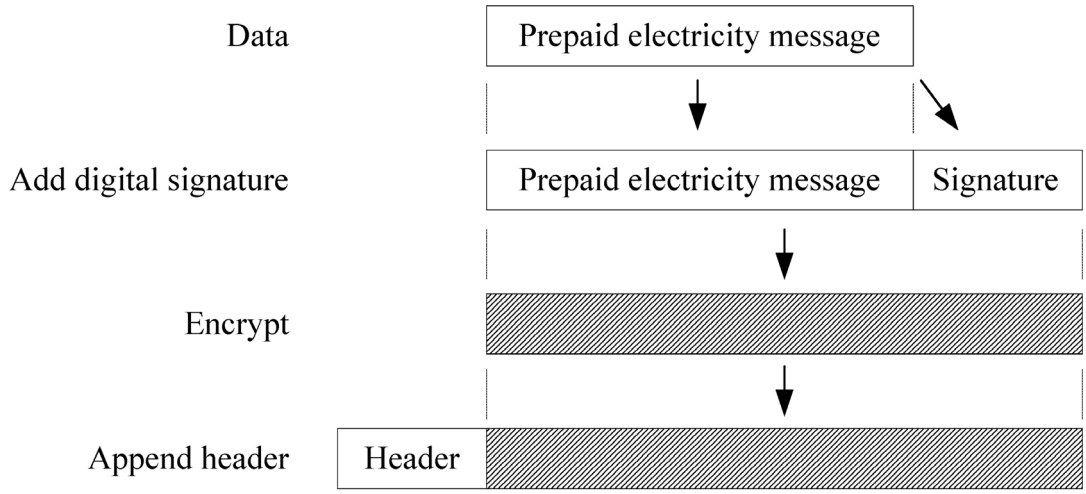

Figure 6.5: Preparation of data for transmission

The appropriate parts of the message and the signature are then encrypted using the current DK shared by the communicating CDU and TM. The encryption is performed after the signing so that a third party can verify the signature once decryption has been performed by one of the communicating parties. This is useful should arbitration be required as any third party can verify a signature, given that the public key certificates are available.

Finally, the header is added to the message. The header contains information such as the addresses of the source and destination entities (see Chapter 8 for details).

6.9 VULNERABILITIES

In the proposed system, the identities of the components are very important to the CDUs as the right to sell tokens is bought before the tokens are sold and is a prerequisite for the sale of tokens. For this reason, authentication must be effective and it must be possible to prove the identities of the communicating parties after the communication has taken place for auditing purposes.

The mechanisms used to provide the necessary services are shown in Table 6.2. This table is similar to Table 3.1, except that the means by which the services are going to be provided is given as opposed to the attacks that challenge the existing services.

Because the issuing of tokens is now on-line, the attacks that relied on the independent operation of the CDU are no longer possible. This means that it is no longer possible to sell a token without the TM having knowledge of it. Because only the TM is capable of generating a token, it can prevent the appropriate token from being made available if the requesting CDU does not have the necessary credit.

Because the responsibility of token generation has been shifted from CDU to TM, non-repudiation, authentication, and integrity are so much more important. To implement these services, public key cryptography is to be used, and this requires the corresponding infrastructure that is discussed in the next chapter.

Table 6.2: Mechanisms used to provide the required services in the proposed CVS

Service	Mechanism
Confidentiality	Encryption is used to keep data confidential. Keys are initialised by means of physical access to the devices (before deployment) and updated using encryption and digital signatures supported by public key cryptography.
Authentication	Digital signatures (supported by public key cryptography) are used to confirm identities of senders and can serve as future proof of transactions.
Integrity	Digital signatures (supported by public key cryptography) are used to ensure integrity of messages transmitted.
Non-repudiation	Digital signatures, in association with the CA, are used to prevent a party from claiming that a particular message was sent.
Access control	Access control to TMs is implemented by whatever means necessary physically. It is also implemented by means of firewalls restricting access to the TMs and in software by refusing to communicate with entities that don't have the necessary credentials. Any customer with a valid request is given access to a CDU. This will be limited physically by limiting the number of people with access to each CDU.
Availability	Availability of the TMs is implemented by ensuring that the TMs have sufficient capacity to handle all of the anticipated requests (determined by the number of CDUs in the field). In conjunction with access control, this ensures that the TMs are sufficiently available. CDUs are sufficiently available to service all requests that comply with access requirements.

6.10 CONCLUSION

The movement of the various types of keys in the current CVS and the proposed CVS were discussed in this chapter. The key hierarchy required to support the proposed CVS was given and the purposes of the various types of keys were also discussed.

The way in which the CDUs, TMs, CAs and KMCs are interconnected in the proposed system was given and explained. The details of the encryption algorithms and key lengths that were used to implement the proposed system were also given, together with motivation for the selections made. Also discussed was the preparation of each message from a security point of view, indicating the process that the data goes through before being transmitted over an insecure network.

A fundamental requirement of the proposed system is that the identities of the communicating parties are reliably known. To this end, the mechanism of authentication in the proposed system was described. In the next chapter, the PKI required to provide the authentication service is discussed.

Seven

PUBLIC KEY INFRASTRUCTURE IN THE PROPOSED SYSTEM

7.1 INTRODUCTION

This chapter describes the manner in which the PKI in the proposed CVS functions. This starts with the authentication requirements that the PKI must facilitate. An architectural description of the PKI is provided. The manner in which keys and security are handled at the CDUs and at the TMs is discussed as well as the procedures to be followed during roll out and normal operation.

A set of security requirements was given in Chapter 3. Protocols that generally implement security requirements were given in Chapter 5. It was determined that XML/SOAP would be the best security protocol to achieve the security requirements. The security functions required in the on-line CVS using XML/SOAP require the use of public key cryptography to provide authentication and non-repudiation, and also to provide the means for securely distributing secret and private keys.

Also discussed in this chapter is the use of multiple keys so that one specific key is never excessively exposed, as well as the concept of backup CAs to ensure high reliability of the PKI.

The messages that are used to implement the PKI are given in the next chapter.

7.2 AUTHENTICATION REQUIREMENTS

Public key cryptography is based on encryption using one key, and decryption using a second key that is related to the first key. One of the keys is public (meaning that it is publicly available), and the other is private. Only the party to which the pair of keys belongs has access to both of them during normal operation. (If a CA generated the pair on

behalf of another party, then it will also have access to the private key at least temporarily.) The public / private pair of keys has the important characteristic that it is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key [12].

Because there are two corresponding keys, public key cryptography allows for:

- authentication of parties,
- non-repudiation of origin,
- key distribution, and
- digital signatures.

However, public key cryptography is significantly more computationally expensive than symmetric encryption. As one of the inventors of public-key encryption stated, "the restriction of public-key cryptography to key management and signature applications is almost universally accepted" [34]. However, public key cryptography can be used to distribute symmetric cryptography keys between communicating parties securely.

The use of public key cryptography requires that each party has its own private key, and that all other parties have access to the corresponding public key. This is the purpose of PKI. PKI is the term given to the systems and entities that are required to facilitate the generation, maintenance, and distribution of public keys.

Once an entity has generated a pair of keys, the public key must be distributed so that parties that receive a message that has been signed using a private key can obtain the corresponding public key to verify the signature. This is typically done by a CA. The sole purpose of a CA is to generate and distribute certificates so that a system based on public key cryptography can perform normal cryptographic functions. The certificates contain details of the entity that the certificate represents, the entity that made the certificate (the CA), the public key itself, and optionally other information.

7.3 SERVICES CURRENTLY AVAILABLE

At present, there are many CA services publicly available on the Internet, such as Thawte [13] and VeriSign [14]. Normally, an institution that wishes to have a certificate from one of these CAs has to comply with a list of requirements so that the CA is sure, to an acceptably high degree of certainty, that the institution is what the certificate will claim it to be. The only proof that an institution is what the certificate claims it to be is the reputation that the issuing CA has, and the procedures, which are publicly known, that the CA follows before signing such a certificate.

In this system, however, the identity of the CA can be guaranteed by the regulatory body that is in control of the prepaid electricity system. Every TM and CDU will be given a copy of the CA's root certificate manually and so do not need to trust any other entities. The CA may thus be self-authenticated in this system. This means that nothing proves the identity of the root CA. For this reason, none of the public CA services are needed in this system.

7.4 ARCHITECTURAL DESCRIPTION

In the proposed system, there are TMs and CDUs that wish to communicate with each other by means of an IP cloud (the Internet or any other similar means of packet-based communication), as shown in Figure 7.1.

All of the TMs and CDUs have direct access to the CA (and backup CA). In a country such as South Africa, the number of TMs and CDUs in the system will not be more than several thousand and thus it is not necessary to have certificate trees but a single CA will suffice. This may change, however, depending on the size of the system that is deployed.

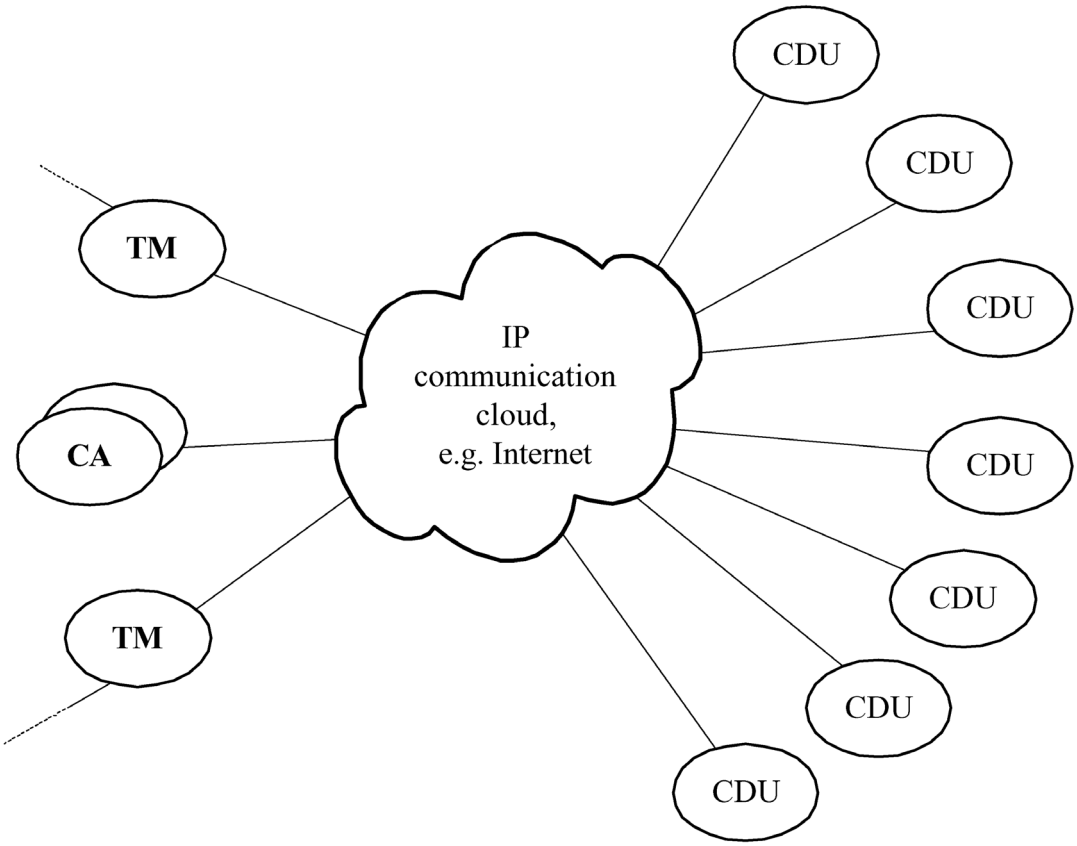

Figure 7.1: Interconnection between CDUs, TMs and the CA

Figure 7.2 shows the practical architecture of the system. The TMs, CA and KMC are grouped together in a secure environment and are protected from the Internet by a firewall. The CDUs, however, are directly exposed to the Internet and are thus more susceptible to security attacks.

During normal operation, a CDU only communicates with a TM (normally only one specific TM). The TMs, in turn, communicate with other information systems, such as Customer Information System (CIS) Database (DB). It is assumed that a secure communication link between the TMs and the CIS DB exists. Because all of these entities are connected via an IP cloud, each entity can communicate with every other entity. It is imperative that a reliable high bandwidth connection is available between all TMs and the CA because without this, the TMs cannot verify new CDU certificates or check that a certificate is not on the CRL. This is available inside the secure environment as described above.

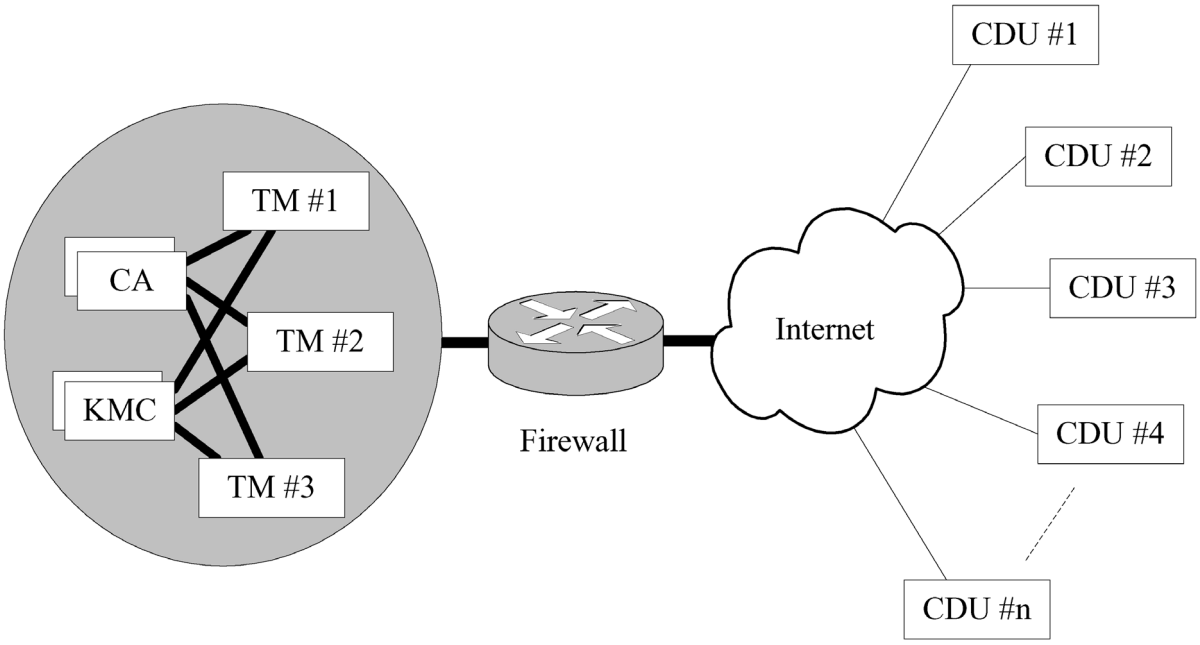

Figure 7.2: Interconnection between CDUs, TMs and the CA (repeat of Figure 6.4)

Because all CDUs are connected to the IP cloud, they can communicate with the CA when a certificate is required. The TMs can similarly communicate with the CA. Logically, there is a connection from every TM to the CA, and also from every CDU to the CA. Multiple CDUs communicate with a single TM.

7.5 SECURITY AT THE CREDIT DISPENSING UNITS

Every CDU has a unique ID (CDU_{ID}) and a private master key (MRK). These two pieces of information are all that differentiate one CDU from another. All of a CDU's keys are obtained using this private master key and thus the secrecy of this key is of the utmost importance.

The private key is issued by the KMC. The corresponding public key is kept by the CA and published in the form of a digital certificate that also contains CDU_{ID}. These two pieces of information are installed on the CDU before the CDU leaves the factory.

The CA's public key certificate, the CDU's unique ID and the CDU's private key are stored at the CDU. These can be stored either on the CDU's local hard disk drive, or, if a higher level of security is required that can be separated from the CDU at a specific venue, they are stored on a smart card. The smart card is then be locked in a safe or removed from the site of the CDU when not in use. A specific person is held responsible for the safety of each individual smart card.

A password is required to access the information stored on the CDU's local hard disk or smart card. Each supervisor at the CDU has a password that can be used for access.

Each smart card has a master password that is known only to the CA. This master password is used by the CA when installing the keys on the smart card. Only the CA is thus capable of resetting passwords on a given smart card. Practically, however, smart cards will normally be replaced should a private master key become compromised.

Should the information contained on the hard disk drive or on the smart card be copied or stolen, it is possible to use part of the CDU's credit at the TM. Since a CDU is held liable for tokens that are issued in its name, it is in the vendor's best interest to inform the CA as soon as possible that the CDU's private key has been compromised. As soon as this is recorded, the TM will no longer interpret a digital signature from that CDU as being valid and thus the authenticity of the CDU's corresponding signatures will be void.

Should a CDU's private key be compromised, a new key must be placed on the CDU's hard disk drive (in the case of no smart card) or a new smart card must be issued for the CDU.

7.6 SECURITY AT THE TMS

Every TM also has a unique ID (TM_{ID}) and a corresponding private key (MRK). These two pieces of information are all that distinguish one TM from another. All of a TM's keys are obtained using this private master key and thus the secrecy of this key is of the utmost importance.

Because it is assumed that TMs are in a physically secure environment, the ID and private key of each TM are simply stored on the local hard disk drive. Should the security of the TM become compromised in any way, the CA is merely informed and the TM's public key certificate is revoked. The process of issuing a new private key is similar to that of deploying a TM for the first time as outlined below.

7.7 PKI PROCEDURES

7.7.1 Roll out

In setting up the PKI, the following are required:

1. The CA must have a master public / private key pair. This is generated by the CA itself.
2. This root certificate must be signed either by the CA itself or another well known CA. (Because the CA's public certificate can be placed on the TMs and the CDUs in a secure manner at the end of manufacture, it does not need to be signed by a publicly-accepted CA.)
3. All parties (CDUs, TMs and the KMC) must initially be issued with private keys in a secure manner. This will also occur during installation.
4. All parties must be issued with an unaltered version of the CA's public key certificate.

When a CDU is deployed:

1. The CA must generate a master public / private key pair (MUK / MRK) for the CDU.
2. The CA must create a public key certificate for the CDU that contains the newly-generated public key.
3. The CA's public key certificate, the CDU's ID, and the CDU's private key are copied onto the hard disk (and then the transport mechanism (e.g. a diskette) is destroyed) or alternatively a smart card or other transport mechanism is provided for the CDU.
4. Each supervisor at the site must select a password that is encoded onto the hard disk or smart card. This password is required to access the CDU's private key. Without such a

password, the CDU cannot prove its identity. (Every time that the CDU is turned on or a shift changes, a valid user name and password must be entered to obtain access to the CDU's private key on the hard disk or smart card.)

Similar steps to the above are followed when a TM is deployed.

7.7.2 Maintenance

The following steps are required for the maintenance of the PKI:

1. The CA must make the required certificates available as they are requested.
2. All parties must periodically be issued with new private keys and the corresponding certificates must be made available (see Chapter 6 for key details).
3. The CA must maintain a CRL, which is a list of certificates that can no longer be trusted because the keys are known to have been compromised.
4. Should a private key be compromised, it must be revoked immediately, and a new private key must be securely issued.
5. Every CDU must obtain a new copy of the CRL from the CA at least once every 24 hours. By doing this, should the private key of a TM be compromised, it can be illegitimately used by a CDU for a maximum of 24 hours. (This will be very rare as the TM is always in a secure environment.)
6. Every TM must obtain a new copy of the CRL from the CA at least once an hour. This limits the amount of time that a CDU can be used since it is compromised to a maximum of one hour after the incident is reported.

When a CDU's private key is compromised, the hard disk is damaged, or the smart card is lost or damaged:

1. The public key certificate corresponding to the private key must be revoked. This is done by adding the ID of the certificate to the CRL.
2. The same process as for the deployment of a CDU is then followed to recreate the CDU so that transactions can again occur.

The alarm system of the building containing the CDU can be linked to the CA so that, should the alarm be triggered, the CA can immediately revoke the public key certificate for the CDU in question.

Similar steps to the above are followed when a TM's private key is compromised.

7.8 MULTIPLE KEYS

The more often a key is used, the greater is the possibility that it will be hacked by a malicious third party. This is because the difficulty of finding the value of the key is inversely proportional to the number of pieces of information that were encrypted with it. This is especially true if it is transmitted on an insecure public network as there is then more ciphertext available with which to attack the key.

For this reason, it was decided that each CDU has two sets of public and private keys: a set for key distribution (the master key pair – MRK / MUK) and a set for signing normal communication (the signing key pair – SK / VK). The master key pair is used by the CA only for sending new keys to the CDU. This key pair is used typically once every couple of months when the CDU's KDK is updated. The SK / VK pair, and the DK are used for every transaction. For this reason, they need to be updated much more frequently, as described in Section 6.3.

With two sets of keys, the number of times that a new KDK must be provided to a CDU for key update purposes is minimised. Normally it will not happen due to the mathematical unfeasibility compared to the gain that can be obtained by cracking one CDU's master key pair.

The same philosophy of two key pairs was also implemented for TMs. The CA will provide two sets of keys to each CDU and TM. When updating the KDK, the CA will use only the master key set for the applicable CDU or TM.

The CA has one MRK with a corresponding root certificate. The CA issues a new key for itself once every couple of months as deemed necessary. This depends on the size of the on-line vending system that it serves. The larger the system, the more times its private key is used to encrypt new certificates, and so the more often that its key needs to be replaced.

7.9 BACKUP CERTIFICATE AUTHORITY

Because the availability of a CA is required for communication to take place, a CA must always be available. Of course, one specific computer cannot always be available, and so at least one backup CA is required. The high availability of CAs is important as the relevant certificates must be accessible should a TM or CDU not already have a copy of it.

The backup CA is required to make available all certificates that the primary CA has. However, it is a stand-alone CA in its own right and can generate certificates as the need arises. Because of this, every TM and CDU must be issued with two CA root key certificates.

7.10 CONCLUSION

This chapter described the need for a PKI and described the PKI implementation for the proposed CVS. A successful PKI is fundamental to ensuring entity authentication and thus providing a solid foundation for the security in the proposed CVS.

The requirements that the PKI had to meet were first given, together with a description of the architecture that was used to meet the requirements. Then the security at each of the CDUs, TMs and CA was described, including the procedures to be followed during roll out and maintenance. A motivation for multiple keys was given, together with the requirement for a backup CA.

The messages that are used to implement the PKI are described in the next chapter.

Eight

MESSAGES IN THE PROPOSED SYSTEM

8.1 INTRODUCTION

A set of messages is required so that the security services discussed in Chapters 3 and 6 can be provided. Providing this set of messages is the focus of the current chapter. First, the context in which the messages are used is given. This is done so that the requirements and functions of the messages can be easily understood.

Next, the design goals of the messages used in the system are given. This leads to the messages used for the prepaid electricity functions, the messages used for the CVS management functions, and the messages used for the PKI functions. For each group of messages, the information that is contained in the messages, as well as the purpose of sending the various components of the messages, is given.

The prepaid electricity messages provide the required prepaid services. The CVS management messages allow the necessary CVS management functions to be carried out so that the prepaid electricity messages have suitable support. The PKI messages provide the required environment for the authentication service that is required before any security requirements can be met.

8.2 MESSAGE CONTEXT

The current CVS is used to provide the communication link between the TM in the MIS, and the CDUs as described in Chapter 2. The communication between the CDUs and the SMSs in the current system is by means of floppy disk or modem communication, as is the link between the SMSs and the TM. In this system configuration, the CDUs and the SMSs were required to have databases in order to buffer transactions as they occurred until a communication link became available, the data was transferred, and an acknowledgement was received.

It was shown in Section 3.4, that by changing the architecture of the system to on-line, SMSs are no longer required and the configuration of the system becomes as shown in Figure 8.1, where multiple TMs are grouped for redundancy. The current CVS has thus been replaced with on-line communication that allows direct communication between CDU and TM. This is only possible if a communication link between CDU and TM is available whenever there is a need for a transaction to occur.

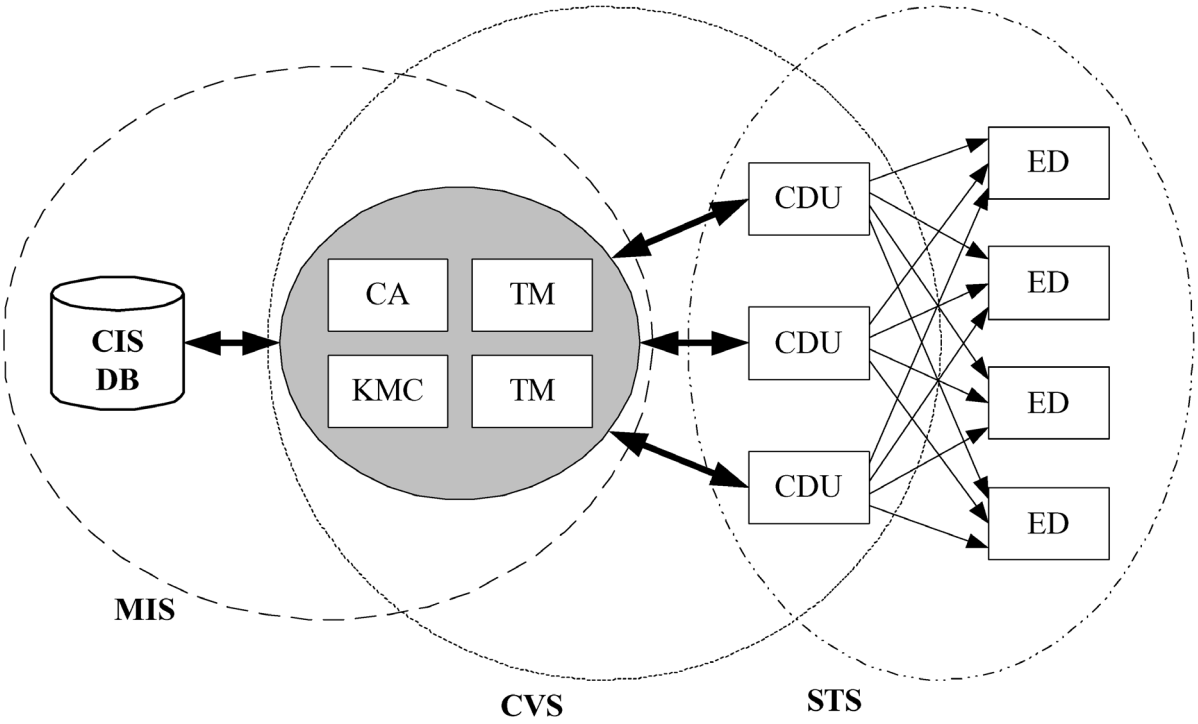

Figure 8.1: On-line CVS architecture showing interconnections between entities; the arrows indicate direction of information transfer

The actual content of the communication messages must now be defined. The communication for the purchase of a prepaid token consists of several messages in a challenge-response fashion, where the CDUs challenge the TMs. The fact that the system uses on-line communication allows the response to be fast enough that the customer perceives it as instantaneous.

All messages in the proposed system are XML/SOAP. These are simple text-based messages that are sufficient for the requirements of this system. SOAP is a web-service communication mechanism, so a request is made to which an answer is given. This is in line with the requirements of this system.

In all of the messages discussed, the party that sends the request regards the message as sent only when it receives a response. If a response is not received within a specified maximum time, the exact same request is retransmitted. The receiving party does not attempt error recovery and assumes that the response that it sent is received. If it was not, the receiving party will get the same query again.

Three types of messages are discussed in this chapter: those that are used for the STS prepaid electricity functions (such as providing a token), those used to maintain the keys in the CVS, and those that are necessary to maintain the PKI.

8.3 DESIGN GOALS

The following requirements had to be met for the communication of the on-line vending system to be successful:

- Each CDU had to be able to communicate with a TM when a transaction was requested and obtain an almost immediate response by means of the on-line communication.

- The TM had to provide the required tokens (obtained from the CIS DB) if the CDU had sufficient credit and valid details were provided to the TM.

- Proof that each message was sent and the entity that sent it had to be available to the recipient. This proof had to be sufficient to show an independent party that a specific sum of money was owed in the case of a dispute.

- The cost of the communication facilities (both installation and running costs) had to be as low as possible while still allowing the on-line functionality of the system to be available.

- The functionality had to be provided that vending of any prepaid system could be done by means of the communication provided by this on-line system, although the focus of the proposed system was on STS prepaid electricity.

8.4 MESSAGES USED FOR PREPAID ELECTRICITY FUNCTIONS

The communication that is required from the system replacing the current CVS must provide the following prepaid electricity functions using a set of messages:

- The purchase of a token for a specific ED.
- The re-issuing of the most recent token for a specific ED should it be lost. (This is valid since the EDs keep track of tokens that have been processed; tokens are for single use only.)
- Updating of customer details in the customer database.
- Issuing management tokens such as changing of keys, changing maximum allowed load or tokens to allow the testing of EDs.
- Providing a list of purchase history that the CDU can use for reconciliation purposes.

The following types of exchanges are required to implement the functionality of a basic CVS:

- transaction, used for issuing and re-issuing tokens, and
- management, used for updating customer details and maintaining keys.

The functions that are required over the TM – CDU link are listed in Table 8.1.

Table 8.1: Required prepaid electricity functions

	Function	Type	Purpose
1	Purchase token	Transaction	Allows a customer to purchase a token representing a quantity of electricity. This transaction is required every time that a token is purchased – most common transaction in the system.
2	Re-issue token	Transaction	Allows the last token for a specific ED to be reprinted in the case of its being lost (allowed for paper tokens as they can be lost and will only be accepted as valid by the specific ED once).
3	Update customer details	Management	Allows customer details to be provided or updated (such as new EDs being installed or customers moving house).
4	Specific management token	Management	Allows management tokens for specific EDs to be generated, such as key change tokens, maximum load change, etc. (used by installation / repair personnel).
5	General management token	Management	Allows management tokens for any ED to be generated, such as test tokens (used by installation / repair personnel).
6	Purchase history	Management	Allows the CDU to obtain a list of transactions for reconciliation purposes.

Each message that is sent between TM and CDU contains a data part that has the necessary parameters for the function being performed. The exact data that each message contains is now given. The rest of the information that constitutes the message is given in the next section.

In each transaction that operates on a specific ED, the ED serial number is repeated. This is done so that both parties involved in the communication do not need to keep record of all the transactions that are occurring. So, when a token is being purchased, the CDU receives the token and also the serial number of the ED that the token is meant for.

8.4.1 Purchase token

Request sent from CDU to TM contains:

- the action required: token purchase,
- the ED's serial number, and
- the amount of credit required.

If the ED serial number that is given is valid, and the CDU has sufficient credit, a positive response is given, otherwise a negative response is given.

Positive response from TM to CDU contains:

- the ED's serial number,
- the key change token pair (if applicable), and
- the requested token.

Negative response from TM to CDU contains:

- the ED's serial number, and
- the appropriate error message.

The actual messages that are sent are indicated in the sequence diagram of Figure 8.2.

The messages indicated in Figure 8.2 have the following meanings:

M_1: The customer goes to the CDU and requests a token for a specific ED for a certain value. Part of this request is the payment for the token.

M_2: The CDU immediately passes this request on to the TM after signing the request.

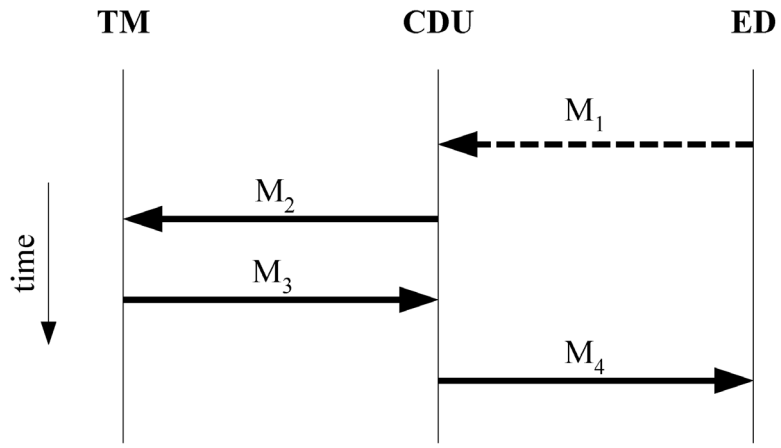

Figure 8.2: Sequence diagram for the purchase of a token (repeat of Figure 3.7)

M_3: The TM responds with the appropriate message. This could be the token itself, or an error message if there was an error processing the request.

M_4: The CDU provides the result to the customer.

The sequence diagram of Figure 8.2 assumes that the TM and the CDU have the corresponding valid certificates of the CDU and the TM, respectively. This is normally the case. If they do not, however, then the message sequence will be expanded as shown in Figure 8.3.

Messages M_1, M_2, M_5, and M_8 of Figure 8.3 map to messages M_1 through M_4 of Figure 8.2. In the case where the TM does not have the appropriate certificate to verify the message sent by the CDU, M_3 and M_4 occur so that the appropriate certificate can be obtained using the following messages:

M_3: The TM, upon determining what certificate is required to verify the contents of the message received, sends a request to the CA for the required certificate of the CDU.

M_4: The CA returns the requested certificate to the TM.

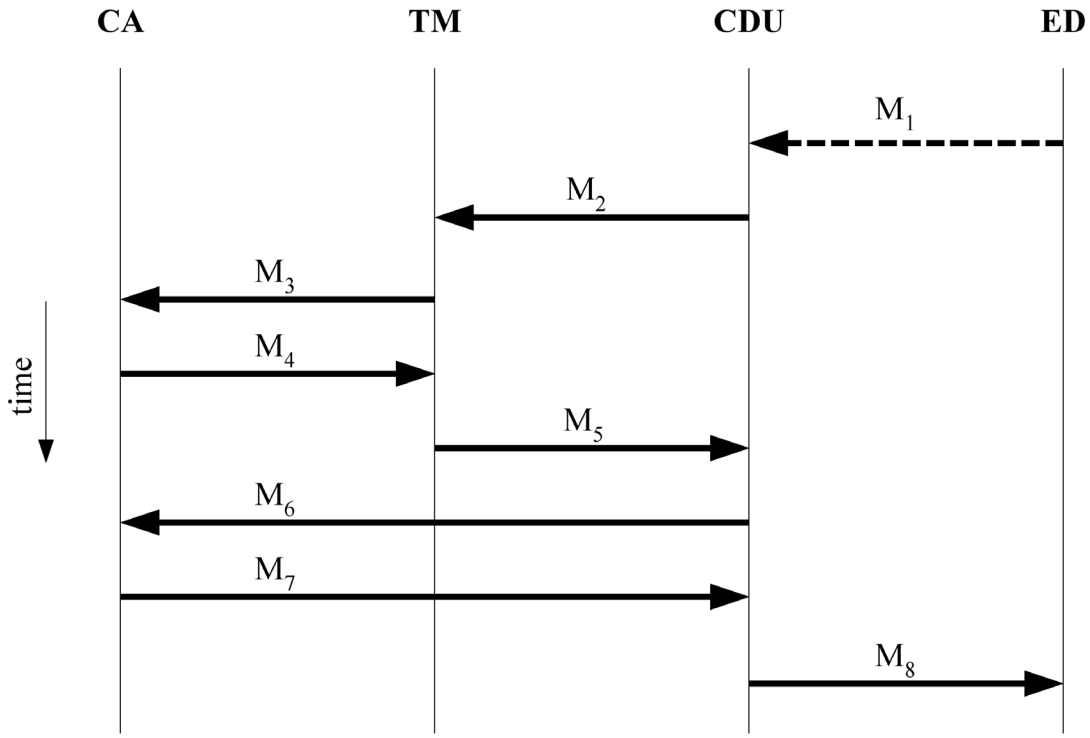

Figure 8.3: Sequence diagram for purchase of token when the certificates of the CDU and the TM must be obtained from the CA

Only once the TM has verified the validity of the message that it received from the CDU will it proceed to process the token request. Similarly, in the case where the CDU does not have the TM's certificate, it obtains the certificate using messages M_6 and M_7:

M_6: The CDU, upon determining what certificate is required to verify the contents of the message received, sends a request to the CA for the required certificate of the TM.

M_7: The CA returns the requested certificate to the CDU.

8.4.2 Re-issue token

If a re-issue of a token is requested, a sequence of messages similar to that of Figure 8.2 occurs.

Request sent from CDU to TM contains:

- the action required: token re-issue, and
- the ED's serial number.

If the ED serial number that is given is valid, a positive response is given, otherwise a negative response is given.

Positive response from TM to CDU contains:

- the ED's serial number,
- the key change token pair (if applicable), and
- the requested token.

Negative response from TM to CDU contains:

- the ED's serial number, and
- the appropriate error message.

8.4.3 Verify customer details

If a request to verify a customer's details is made, a sequence of messages similar to that of Figure 8.2 occurs. Verifying a customer's details is useful to ensure that the correct ED number is given so that a token is not issued for the incorrect ED. This verification will typically occur as a precursor to the purchase of a token.

Request sent from CDU to TM contains:

- the action required: verify ED serial number, and
- the ED's serial number.

If the ED serial number that is given is valid, a positive response is given, otherwise a negative response is given.

Positive response from TM to CDU contains:

- the ED's serial number, and

- the customer's name and address.

Negative response from TM to CDU contains:

- the ED's serial number, and
- the appropriate error message.

8.4.4 Update customer details

If a request to update a customer's details is made, a sequence of messages similar to that of Figure 8.2 occurs.

Request sent from CDU to TM contains:

- the action required: update customer details,
- the ED's serial number,
- the new customer name (if applicable), and
- the new customer address (if applicable).

If the ED serial number that is given is valid, a positive response is given, otherwise a negative response is given.

Positive response from TM to CDU contains:

- the ED's serial number,
- the positive result of the change, and
- the customer name and address.

Negative response from TM to CDU contains:

- the ED's serial number, and
- the appropriate error message.

8.4.5 Request specific management token

If a request for a management token for a specific ED is made, a sequence of messages similar to that of Figure 8.2 occurs.

Request sent from CDU to TM contains:

- the action required: specific management token required,
- the specific management token type,
- the specific management token parameters, and
- the ED's serial number.

If the ED serial number that is given is valid and the CDU is permitted to request the indicated type of specific management token, a positive response is given, otherwise a negative response is given.

Positive response from TM to CDU contains:

- the ED's serial number,
- the description of management token, and
- the specific management token.

Negative response from TM to CDU contains:

- the ED's serial number, and
- the appropriate error message.

8.4.6 Request general management token

If a request for a general management token is made, a sequence of messages similar to that of Figure 8.2 occurs.

Request sent from CDU to TM contains:

- the action required: general management token required, and
- the general management token type.

If the CDU is permitted to request the indicated type of specific management token, a positive response is given, otherwise a negative response is given.

Positive response from TM to CDU contains:

- the description of management token, and
- the general management token.

Negative response from TM to CDU contains:

- the appropriate error message.

8.4.7 Request purchase history

A CDU may request a purchase history from a TM for auditing purposes. If such a request is made, a sequence of messages similar to that of Figure 8.2 occurs.

Request sent from CDU to TM contains:

- the action required: purchase history required, and
- the date range or ED serial number range.

If the parameters in the request are valid, a positive response is given, otherwise a negative response is given.

Positive response from TM to CDU contains:

- the list of purchase history including ED serial numbers, dates of transactions and amounts.

Negative response from TM to CDU contains:

- the appropriate error message.

8.5 INFORMATION CONTAINED IN EVERY PREPAID ELECTRICITY MESSAGE

Every message transmitted that is part of the CVS has a digital signature appended to it to assure the receiving party that the claimed transmitting party sent the message and none of the contents of the message has been changed. This is important for proving the integrity of

the message and for authentication of origin so that financial obligations can later be proved if necessary (see Section 6.6).
Every message that is sent has the following components in addition to those specified for the various types of messages in Section 8.4:

- the vending system ("CVS v2.0", in this case) (sys),
- the ID of the source (the specific CDU or TM) (S_{ID}),
- the ID of the destination (the specific TM or CDU) (D_{ID}),
- the transaction ID so that this message can be uniquely identified (sequentially increasing number from the initiating party) (T_{ID}),
- the instant at which the message was transmitted according to the sending party (stamp), and
- the digital signature (sig).

The vending system, ID of the source, ID of the destination, transaction ID and instant at which the message was transmitted do not need to be encrypted. However, these items should form part of the message that is digitally signed, and so the digital signature is computed over the entire message before it is encrypted with the current session key. It is important that the customer data is kept private, and so encryption with the current session key is necessary before the data is transmitted over the public Internet.

In summary, every message transmitted by a CDU or a TM that is destined for a TM or a CDU, respectively, for the purpose of a prepaid electricity transaction has the vending system, ID of the source, ID of the destination, transaction ID, instant at which the message was transmitted, the action required and the parameters necessary to augment the action request. All of this is then digitally signed using the SK (written as SK_{CDUx} meaning SK of CDU number x) and the sensitive parts of the message are encrypted using the current DK (DK_{CDUx}).

In the case of a request to buy a token, the messages will contain the following information:

Prepaid electricity message

= < action required: token purchase ||
ED serial number || credit required >

CVS message = <CVS message public components> ||
<CVS message confidential components> ||
<CVS signature>

= sys || S_{ID} || D_{ID} || T_{ID} || stamp ||
E_{DK} [<Prepaid electricity message>] ||
E_{SKCDUx} [H [sys || S_{ID} || D_{ID} || T_{ID} || stamp ||
<Prepaid electricity message>]]

Transmitted message = <TCP/IP header> ||
sys || S_{ID} || D_{ID} || T_{ID} || stamp ||
E_{DK} [<Prepaid electricity message>] ||
E_{SKCDUx} [H [sys || S_{ID} || D_{ID} || T_{ID} || stamp ||
<Prepaid electricity message>]]

The last part of the CVS message is the digital signature over the entire CVS message, and so provides authentication and non-repudiation over the contents of the entire CVS message. However, parts of the message are then encrypted using the current DK. (This implies of course that for the signature to be verified, the encrypted parts of the message must first be decrypted by a party that has the required DK. This does not reveal the value of the DK, however, as possession of the encrypted and decrypted versions of the same message do not allow the key to be easily found. Recall that the KMC will store the DK and so it can be used for this purpose.)

8.6 MESSAGES USED FOR CVS MANAGEMENT FUNCTIONS

As previously mentioned, communication is required for the management of the CVS (other than the PKI infrastructure) to operate properly. The functions that are required from TM to KMC, or from CDU to KMC are listed in Table 8.2.

Table 8.2: Required CVS management functions

	Function	Type	Purpose
1	Request key-encrypting key	Normal operation	A key-encrypting key does not yet exist or has expired and so must be issued to the requesting entity.
2	Request session key	Normal operation	A session key does not yet exist or has expired and so must be issued to the requesting entity.
3	Request key-encrypting key be revoked	Alarm	A key-encrypting key has been compromised and so must no longer be used but must be replaced immediately.
4	Request session key be revoked	Alarm	A session key has been compromised and so must no longer be used but must be replaced immediately.

8.6.1 Request new Key Distribution Key

A KDK can be requested as a normal operation in two cases: the key does not yet exist, or has expired and must be replaced. In both of these cases, the KMC generates a new key to be used between the requesting party and the KMC when future key requests are made.

When a KDK is required, the message sent from the CDU or TM to the KMC (derived from X.509 authentication procedures given in Section 3.5.3) is

$$MRK_A \{ t_A \| r_A \| C \| \text{<request KDK>} \| A \},$$

where A is the identity of either the CDU or the TM, t_A is the current time according to A, r_A is a nonce generated by A, and C is the KMC's identity. This is signed using A's private master key (MRK_A) as this is the next key up in A's key hierarchy above the KDK.

The response from the KMC, if the request was valid, is

$$MRK_C \{ t_C \| r_A \| A \| E_{MUKa} [KDK_{AC}] \| \text{<expiry date>} \| C \},$$

where t_C is the time according to C, r_A is the random number that A sent to C, KDK_{AC} is the new KDK (shared between A and C) that has been encrypted with A's public master key (MUK_A) and the expiry date is the last time at which the key is valid. If the request was not valid, the response is

$$MRK_C \{ t_C \| r_A \| A \| \text{<error message>} \| C \},$$

where the error message is the problem that resulted in a new key not being issued.

8.6.2 Request new Data Key

A DK can be requested when the key does not yet exist, or has already expired and must be replaced. In both of these cases, the KMC generates a new key to be used between the two identified parties and issues it to the parties as the requests are made.

When a DK is required, the message sent from the CDU or TM to the KMC is

$$A \| E_{KDKac} [t_A \| r_A \| C \| \text{<request session key>} \| A \| B],$$

where A is either the CDU or the TM, t_A is the current time according to A, r_A is a nonce generated by A, C is the KMC's identity, and B is the identity of the party that the session key is to be shared with. This is encrypted using A's KDK that is shared with C (KDK_{AC}) as this is the next key up in A's key hierarchy. The fact that only A and the KMC have A's KDK proves that A made the request.

The response from the KMC, if the request was valid, is

$$C \parallel E_{KDKac} [\, t_C \parallel r_A \parallel A \parallel DK_{AB} \parallel \text{<expiry date>} \parallel C \,],$$

where t_C is the time according to C, r_A is the random number that A sent to C, DK_{AB} is the new DK, and the expiry date is the last time at which the key is valid. If the request was not valid, the response is

$$C \parallel E_{KDKac} [\, t_C \parallel r_A \parallel A \parallel \text{<error message>} \parallel C \,],$$

where the error message is the problem that resulted in a new key not being issued. If party A is the first to request this session key, the KMC generates it. If party A is the second to request this session key (i.e. B has already requested and received the key), the KMC merely passes this already existing key on to A.

8.6.3 Request Key Distribution Key be revoked

When a KDK has been compromised, it must no longer be used with immediate effect. The message that the TM or CDU will send is

$$A \parallel E_{KDKac} [\, t_A \parallel r_A \parallel C \parallel \text{<revoke KDK>} \parallel A \,].$$

The KDK may be used to encrypt this request because A already has the key, and if any other party has the key, the compromise has been proved. The response from the KMC will be

$$MRK_C \{\, t_C \parallel r_A \parallel A \parallel \text{<result>} \parallel C \,\},$$

where the result is the confirmation that the KDK will no longer be used. All keys that were encrypted with the KDK are also implicitly revoked as a history of those keys will allow the keys themselves to be discovered. The TM or CDU will have to request another KDK and another DK before any further communication may take place.

8.6.4 Request Data Key be revoked

When a DK has been compromised, it must no longer be used with immediate effect. The message that the TM or CDU will send is

$$A \parallel E_{KDKac} [\, t_A \parallel r_A \parallel C \parallel \text{<revoke DK>} \parallel A \,].$$

The KDK is used to sign this request because it is used for communication regarding keys between an entity and the KMC. The response from the KMC will be

$$MRK_C \{\, t_C \parallel r_A \parallel A \parallel \text{<result>} \parallel C \,\},$$

where the result is the confirmation that the session key will no longer be used. The TM or CDU will have to request another DK before any further prepaid electricity messages can be sent or received.

8.7 MESSAGES USED FOR PKI FUNCTIONS

The PKI requires maintenance to stay operational at an acceptable level of security. Various functions are required to maintain the validity of the public / private key pairs in the system. These messages are from TM or CDU towards the CA. The functions relevant for the proposed CVS are listed in Table 8.3.

8.7.1 Request certificate

When a certificate is required to verify a signature, the party that needs the certificate sends

$$A \parallel B \parallel \text{<certificate request>},$$

where A is the requesting party, B is the party of which the certificate is required and <certificate request> is the required certificate. No signing is required as the certificate that will be sent back has already been signed by the CA. If the certificate is not received within the timeout period, the request is merely repeated.

Table 8.3: Required PKI management functions

	Function	Type	Purpose
1	Request certificate	Normal operation	Request a certificate for a particular entity in order to verify a digital signature or decode a message as the requesting entity does not have the correct current certificate.
2	Request CRL	Normal operation	Request an updated copy of the CRL so that the validity of certificates can be verified.
3	Request new signing key pair	Normal operation	When a signing key pair has been used more than the maximum recommended number of times, the risk for attack increases. This request is made when the risk is unacceptably high.
4	Notify signing key pair compromised	Minor alarm	The TM or CDU has determined that the signing key pair has been compromised. This implies that a new signing key pair must be issued and the current one revoked.
5	Notify master key pair compromised	Major alarm	The TM or CDU has determined that the master key pair has been compromised. This requests that all certificates relating to the CDU be revoked and the CDU must be issued manually with a new master key pair.

8.7.2 Request CRL

When the validity of a certificate must be verified, the party that must do the verifying requests the CRL from the CA by sending

A || <CRL request>,

to the CA where A is the requesting party. No signing is required as the CRL itself has already been signed by the CA. If the CRL is not received within the timeout period, the request is merely repeated.

8.7.3 Request new signing key pair

The request for a new signing key pair (SK and VK) must be made periodically so that a private key is not used to sign too many messages. This is necessary to limit the amount of ciphertext that an attacker could use. Another reason to request a new signing key pair is if the old one should be compromised.

The message

$$MRK_A \{ t_A \| r_A \| C \| \text{<request new signing key pair>} \| A \},$$

is sent to the CA, where A is the requesting CDU or TM, t_A is the current time, r_A is a nonce, C is the CA's identity, and A is the requesting party's identity. The CA obtains A's MUK from the KMC in order to verify the signature and thus that the request did indeed come from A. Should the request be in order, the CA will generate a new key pair and send the private key to A in the message

$$MRK_C \{ t_C \| r_A \| A \| E_{KDKAC} [MRK_A] \| \text{<expiry date>} \| C \}.$$

If the request was not valid, the CA will response with

$$MRK_C \{ t_C \| r_A \| A \| \text{<error message>} \| C \}.$$

The new private key of the signing key pair for A is encrypted with the KDK shared between A and C as this key is the next highest in the key hierarchy.

8.7.4 Notify signing key pair compromised

When a signing key pair is compromised (i.e. the SK of the relevant party is no longer secret), it must no longer be used with immediate effect. It must also be reported immediately so that the corresponding certificate can be added to the CRL to limit malicious use of the compromised private key. The message that the TM or CDU will send to the CA is

$$SK_A \{ t_A \| r_A \| C \| \text{<revoke signing key pair>} \| A \}.$$

The private key of the signing key pair of A (SK_A) may be used to sign this request because A already has the key, and if any other party has the key, the compromise has been proved. The response from the CA will be

$$MRK_C \{ t_C \| r_A \| A \| \text{<result>} \| C \},$$

where the result is the confirmation that the indicated SK is no longer valid.

As soon as the signing key pair has been compromised, no further messages sent by the specific entity can plausibly be signed. For this reason, a new signing key pair must be requested immediately.

8.7.5 Notify master key pair compromised

When a master key pair is compromised, it must no longer be used with immediate effect. Also, no keys that were derived from the master key pair may be used any longer. This means that all keys that an entity has have become meaningless. The only thing that a TM or CDU can do in such a situation is report the compromise and stop operating until a new master key pair has been issued. This can only be done manually in the proposed CVS.

To report a compromised master key pair, the TM or CDU sends the message

$$MRK_A \{ t_A \| r_A \| C \| \text{<revoke master key pair>} \| A \}.$$

The response from the CA will be

$$MRK_C \{ t_C \| r_A \| A \| <result> \| C \},$$

where the result is the confirmation that the master key pair and hence all keys belonging to A are no longer be valid and will no longer be used.

8.8 CONCLUSION

This chapter introduced all the messages required for the proposed CVS to operate: prepaid electricity function messages, CVS management function messages, and PKI function messages.

For the prepaid electricity function messages, the list of supported functions was given and all of the required parameters for each of the functions were listed and explained. Sequence diagrams were given in the case of the purchase of a token, together with an indication of the PKI messages required if the relevant certificates were not available in local caches. The manner in which all of the prepaid electricity messages are constructed was explained.

The messages used for the CVS management functions were listed and covered the requesting of new KDKs and DKs, and the revocation of these keys should it become known that they have been compromised.

The content of the PKI messages was given with all of the associated functions of requesting new signing key pairs as well as reporting the compromise of these or the compromise of the master key pair.

Nine

CRITICAL ANALYSIS OF SECURITY IN THE PROPOSED SYSTEM

9.1 INTRODUCTION

At this point, the manner in which the proposed system was implemented has been given. Now it must be shown that the proposed system meets the security requirements that were given briefly in Chapter 1 and in more detail in Chapter 2.

In this chapter, the authentication requirements are first given in detail as the core of the proposed CVS cannot be successful without fully operational authentication. These requirements are then applied to the prepaid electricity messages, the CVS management messages, and the PKI management messages.

Finally, the security of the complete proposed on-line CVS is analysed in terms of all of the required security services.

9.2 AUTHENTICATION REQUIREMENTS

If there are two parties, A and B, that send messages to one another, the following requirements must generally be met by authentication protocols:

1. each message received by A (or B) was indeed sent by B (or A) and has not changed since it was sent (origin authentication and integrity),
2. each message received by A (or B) has not been sent before (freshness),
3. each message received by A (or B) was intended for A and not for any other party,
4. each message received by A (or B) was sent in response to the previous message from B (or A) (part of current communication), and
5. the sending party wishes to know that the receiving party actually knows the plaintext key as this key is to be used later to guarantee origin of data.

These authentication requirements are going to be used to test the messages previously given for each of the prepaid electricity messages, the CVS management messages, and the PKI management messages.

9.3 PREPAID ELECTRICITY MESSAGES

The general structure of a prepaid electricity message, before being transmitted, was given in Section 8.5 as

$$sys \,||\, S_{ID} \,||\, D_{ID} \,||\, T_{ID} \,||\, stamp \,||\, E_{DK}\,[\,\langle \text{Prepaid electricity message} \rangle\,] \,||\, E_{SKCDUx}\,[\,H\,[\,sys \,||\, S_{ID} \,||\, D_{ID} \,||\, T_{ID} \,||\, stamp \,||\, \langle \text{Prepaid electricity message} \rangle]\,],$$

where the entire message is signed using the sending party's private key of the signing key pair, and the prepaid electricity part of the message is encrypted with the current DK shared between the communicating parties. A message of this form is used for the request and also for the response.

The first authentication requirement is met because the message contains the destination address (D_{ID}) and has been signed using the sending party's private key. Thus, assuming that the signature algorithm is sound, the message could not have changed (integrity is guaranteed) and it was indeed sent from the claimed source (the origin is proved by the sending party's public key being able to decode the message and the decrypted message making sense).

Each message sent has an increasing transaction ID (T_{ID}). As a transaction ID is used only once (all transactions only consist of two messages – request and response), the transaction ID makes the message unique. Since the entire message is signed with the sending party's private key, only the sending party is able to insert the transaction ID and thus freshness is guaranteed.

The third authentication requirement is that the message was indeed intended for the receiving party. The destination address (D_{ID}) is present in the message and could not have

been changed by any other party as the entire message has been signed by the sending party.

The fourth requirement that the message was sent in response to the previous message in the sequence is guaranteed by the T_{ID}.

The fifth requirement is not necessary in this case because the PKI is used to prove origin of messages.

Given that all of the authentication requirements are met, attacks from malicious sources will not succeed in interacting with the CVS as long as the keys used within the system are not compromised. The only other security threat is lack of availability. This can be enforced as well as possible with firewalls.

9.4 CVS MANAGEMENT MESSAGES

The CVS management messages are concerned with the distribution of keys for normal system operation and the revoking of keys when they are compromised. When a key needs to be distributed, the next key up in the key hierarchy (see Figure 6.2) is used for confidentiality, integrity, and authenticity. When a key is to be revoked, that same key is used for integrity and pseudo authenticity.

9.4.1 Key distribution

For the distribution of KDKs, the general format of the request is

$$MRK_A \{ t_A \| r_A \| C \| \text{<request KDK>} \| A \},$$

and the general format of the response is

$$MRK_C \{ t_C \| r_A \| A \| E_{MUKa} [KDK_{AC} \| \text{<expiry date>} \| C \},$$

if the request was valid and successfully processed.

To test the authentication in this case, we assume that the keys being used to do the signing (MRK_A and MRK_C) have not been compromised, the key used to do the encryption (E_{KUKA}) has integrity, and that the encryption and digital signing algorithms are sufficiently secure.

The first authentication requirement is that the message received by C was indeed sent by A and has not been modified. This is guaranteed as the request was signed using a private key belonging to A and hence only A has the key. Also, the response received by A could not have been modified as C signed the message using C's private key that no other party possesses.

The second requirement that the message received by C has not been sent before is proved with the included timestamp, t_A. The allowable time window can be large as the request for new keys will not occur more than once per day. The response is also unique as the nonce that A sent to C, r_A, is also included.

Thirdly, the request was indeed destined for C as it contains C's identity in it which could not have been modified, as the message has been signed. The response was also indeed destined for A as it contains A's identity in it and the response has also been signed.

The fourth requirement only applies to the response that is received by A. The fact that the message was sent in response to the previous message is guaranteed by the nonce that A used initially, r_A.

The fifth requirement, that the sending party, C, wishes to know that the receiving party, A, knows the plaintext key does not concern C, as A will again request the new key if it was not received.

In the case of the distribution of a DK, the format of the request is

$$A \parallel E_{KDKac} [\, t_A \parallel r_A \parallel C \parallel \text{<request session key>} \parallel A \parallel B \,],$$

and the format of the response is

$$C \parallel E_{KDKac} [\, t_C \parallel r_A \parallel A \parallel DK_{AB} \parallel \text{<expiry date>} \parallel C \,].$$

The authentication requirements are met in exactly the same way as explained above for the KDK case, except that here the KDK_{AC} is used to prove identity as it is shared between A and C only.

9.4.2 Key revocation

For the remainder of the CVS key management messages, the revoking of a KDK, the general format of the request is

$$A \parallel E_{KDKac} [\, t_A \parallel r_A \parallel C \parallel \text{<revoke KDK>} \parallel A \,],$$

and the general format of the response is

$$MRK_C \{\, t_C \parallel r_A \parallel A \parallel \text{<result>} \parallel C \,\},$$

and indicates whether the request was valid and successfully processed. Because a request is being made for a key to be revoked, the key itself that is being revoked can be used to encrypt the request. The only consequence that this has is that the attacking party can revoke the key and this is actually a benefit, not a security problem.

The first requirement, that the message received by C is indeed that which was sent by A (or the malicious party) is proved by the fact that the revocation request is encrypted by the compromised key. The response, however, must be signed using C's private key that has not been compromised, otherwise a malicious party could inform A that the key has been compromised although C is not aware of it.

The second requirement is that the request has not been sent before. This is irrelevant in this case because once a key has been revoked, all messages encrypted or signed with it will be ignored as they will be regarded as invalid.

Thirdly, both the request and the response are intended for the parties to which they are sent because they have the names of the relevant parties are in the part of the message that is encrypted or signed.

The fourth requirement is guaranteed as a key can only be revoked once. Thus the request for the revocation for a key can be implemented only once – additional requests will have no effect.

The last requirement is not relevant in this case.

In the case where a DK is to be revoked, the only change is that the revocation of a DK is requested in the message as

$$A \parallel E_{KDKac} [\, t_A \parallel r_A \parallel C \parallel \langle \text{revoke DK} \rangle \parallel A \,].$$

The analysis of the requirements is exactly the same as above.

Given that all of the authentication requirements are met for all of the messages, attacks from malicious sources will not succeed in compromising the security of the CVS management.

9.5 PKI MANAGEMENT MESSAGES

9.5.1 Certificate and CRL request

The request for a certificate or for a CRL is a special case as data that is publicly available from the CA is being requested. The request is either

$$A \parallel B \parallel \langle \text{certificate request} \rangle,$$

when A requests the certificate of B, or

$$A \parallel \langle \text{CRL request} \rangle,$$

when A requires the most recent CRL. These messages do not need to be signed as there is nothing confidential in them and even if the messages are modified en route to the CA, the request can merely be repeated until the correct certificate is returned.

The certificate that is returned to the requestor has no special security requirements as it is already a publicly available document. The integrity of the document must be assured, however, and this is done by the CA signing the document and each party having a public key certificate of the CA that is known to be accurate.

9.5.2 New signing key pair request

The request for a new signing key pair, where A requests the pair from C, is of the form

$$MRK_A \{ t_A \| r_A \| C \| \text{<request new signing key pair>} \| A \},$$

where the requesting party's private master key is used to sign the request. The response is of the form

$$MRK_C \{ t_C \| r_A \| A \| E_{KDKAC} [MRK_A] \| \text{<expiry date>} \| C \},$$

if a new key has been issued or

$$MRK_C \{ t_C \| r_A \| A \| \text{<error message>} \| C \},$$

if the request was invalid or there was a problem in carrying out the request.

The first authentication requirement is that the request sent from A to C was indeed sent by A and is unmodified. This is the case because the entire request is signed using A's master private key that no other party possesses.

The second requirement is met due to t_A and r_A being in the request and r_A being in the response. So it is not possible for either the request or the response to have been sent previously.

The destination of the response is included in all of the messages and is signed by the party sending the message, and so the third requirement is met.

Fourthly, the nonce, r_A, is included in the response and so the response can only be an answer to the request.

Finally, party C will know that the key is not known when the key request has been repeated and so the requirement that the receiving party knows the plaintext will be met by retrying.

9.5.3 Revoke key pair

When it comes to the attention of a party that one of its private keys has been compromised, it immediately sends a request for the specific key pair to be revoked. The request from the party, A, to the CA is of the form

$$SK_A \{ t_A \| r_A \| C \| \text{<revoke signing key pair>} \| A \}.$$

The response will be of the form

$$MRK_C \{ t_C \| r_A \| A \| \text{<result>} \| C \}.$$

The first authentication requirement is that the message sent to the CA is indeed from the claimed party and has not been changed. This is proved by the fact that the sending party uses its private key (even the one that has been compromised as explained in Chapter 8) to sign the message sent.

The second and fourth requirements are met by r_A.

The third requirement is met by the identity of the destination party being included in the signed message.

The fifth requirement is not relevant. The requesting party will know that the certificate has been revoked by seeing it listed in a recent CRL.

9.6 COMPLETE CVS SECURITY

As explained in the Chapters 1 and 2, STS is functioning well in the field. The focus, therefore, is on CVS only, i.e. the issuing and delivery of STS tokens to the customer. We now consider the security aspects of the proposed on-line CVS as a whole.

The first security function that is required is confidentiality – the certainty that the keys used within the system will not be available outside the system, and the data that is sent between CDU and TM will not be available to malicious third parties. The use of a key hierarchy assures that keys can be replaced before they have been used too many times to cause a security threat. The key hierarchy allows all keys, except the master private key pair (MRK / MUK), to be replaced on demand in an on-line process. All messages that are sent within the on-line CVS have encryption of the data where necessary. All STS tokens are encrypted using the relevant DKs and so customer information is not accessible to outside parties.

Authentication is very important in a distributed system such as the on-line CVS, as financial obligation can be proved only if the origin of the messages is proved beyond reasonable doubt. In the proposed on-line CVS, all messages that have a party that must be held responsible for sending them are digitally signed using the sending party's appropriate private key. This allows the receiving party to verify that the claimed origin of the message did in fact originate the message, and this proof, together with an intact PKI, is sufficient to prove financial obligation.

Similar to the authentication described above, integrity is also assured by means of digital signatures for the prepaid electricity functions and the CVS management functions. When a digital signature agrees with the message, the integrity of the message is also proved.

Non-repudiation involves a party that sends a message not being able later to deny sending the message in question. This security service is also provided by the digital signature mechanism in this system, as only one party has the private key of the signing key pair.

Access control is provided by the combination of the firewall indicated in Figure 6.4 and each unit in the system ignoring messages that are not appropriately signed and from a recognised party in the system.

Availability operates in conjunction with access control. Because access is granted only to the entities that are allowed to have access, the units in the on-line CVS will be available to those entities that are a part of the system but not to those that are not. This method of assuring availability will not work against a denial-of-service attack where malicious parties flood the relevant parts of the network. In this case, service will not be available to anyone.

9.7 CONCLUSION

This chapter has examined the security that is provided by the proposed on-line CVS. A higher level of security is required because communication is on-line and the system is exposed to malicious attacks from the Internet.

It was shown that the given authentication requirements were met for the prepaid electricity messages, the CVS management messages, and the PKI management messages, assuming that the encryption and signature algorithms employed are sound.

The last part of this chapter examined the entire proposed on-line CVS and found that confidentiality, authentication, integrity, non-repudiation, access control, and availability are all successfully provided.

The proposed on-line CVS that has been described in this document thus meets all of the relevant security requirements and is a suitable drop-in replacement for the existing CVS from a security point of view.

Ten

SECURITY EVALUATION

10.1 INTRODUCTION

In this chapter, the implementation of the system that was created to demonstrate the operation of the proposed system is briefly described. All of the aspects that were implemented are given and it is indicated how the implementation differs from a production system. Various interfaces are provided that allow the internal operation of the system to be observed.

The proposed CVS was designed with a set of goals in mind. The manner in which the goals were met is explained on a goal-by-goal basis.

Because the current CVS is susceptible to a number of attacks, the way in which these attacks are thwarted in the proposed CVS are explained and motivated for each security service. This shows that the goals of the proposed CVS were indeed achieved.

Finally, the current and the proposed CVS are both critically analysed in terms of the X.810 series of security frameworks. The two are compared to determine which provides the required security functionality.

10.2 SYSTEM IMPLEMENTATION

A demonstration system was implemented to show the operation and the interoperation of the various components of the proposed CVS. The focus in the implementation is on the security aspects that have been described in this document and not on other aspects that will be required in real-life use. Thus, the system is merely for demonstration of security functionality, and was not intended to be a production-ready implementation.

The following CVS components were implemented: CDUs, TMs and a KMC / CA. Each of these has the ability to operate independently with its own database for recording

relevant data and messages sent and received. There is also a debugging interface for each of these components so that the internal working of the component can be observed for analysis and demonstration purposes.

10.2.1 Software architecture

The software architecture of the implementation is as shown in Figure 10.1. One of each of CDU, TM, KMC and CA are indicated. For each type of entity, the functional blocks that are used are indicated. The functions that these blocks perform are described in the subsections below. The arrows between the functional blocks indicate the direction of information flow.

10.2.2 CDUs

The CDU is the component that the client, or at least the operator in the shop, will see. As such, it has a user interface where the request for a token can be entered. The functions that were implemented for the CDU are:

- the purchase of a token,
- the re-issuing of a token,
- the updating of customer details,
- the issuing of management tokens, and
- the providing of purchase history.

The result of the purchase of a token is the 20 digit token written on the screen and printed. No interface to print magnetic card tokens was implemented.

The CDU is the unit in the system that initiates all of the communication. The TM reacts to requests by the CDU. The KMC and the CA react to requests from the CDU and TM.

Each of the messages in Chapter 8 required to implement these functions was implemented in the manner described in Chapter 8. Sample messages of the messages used in the purchase of a token are included in Addendum A for illustration purposes.

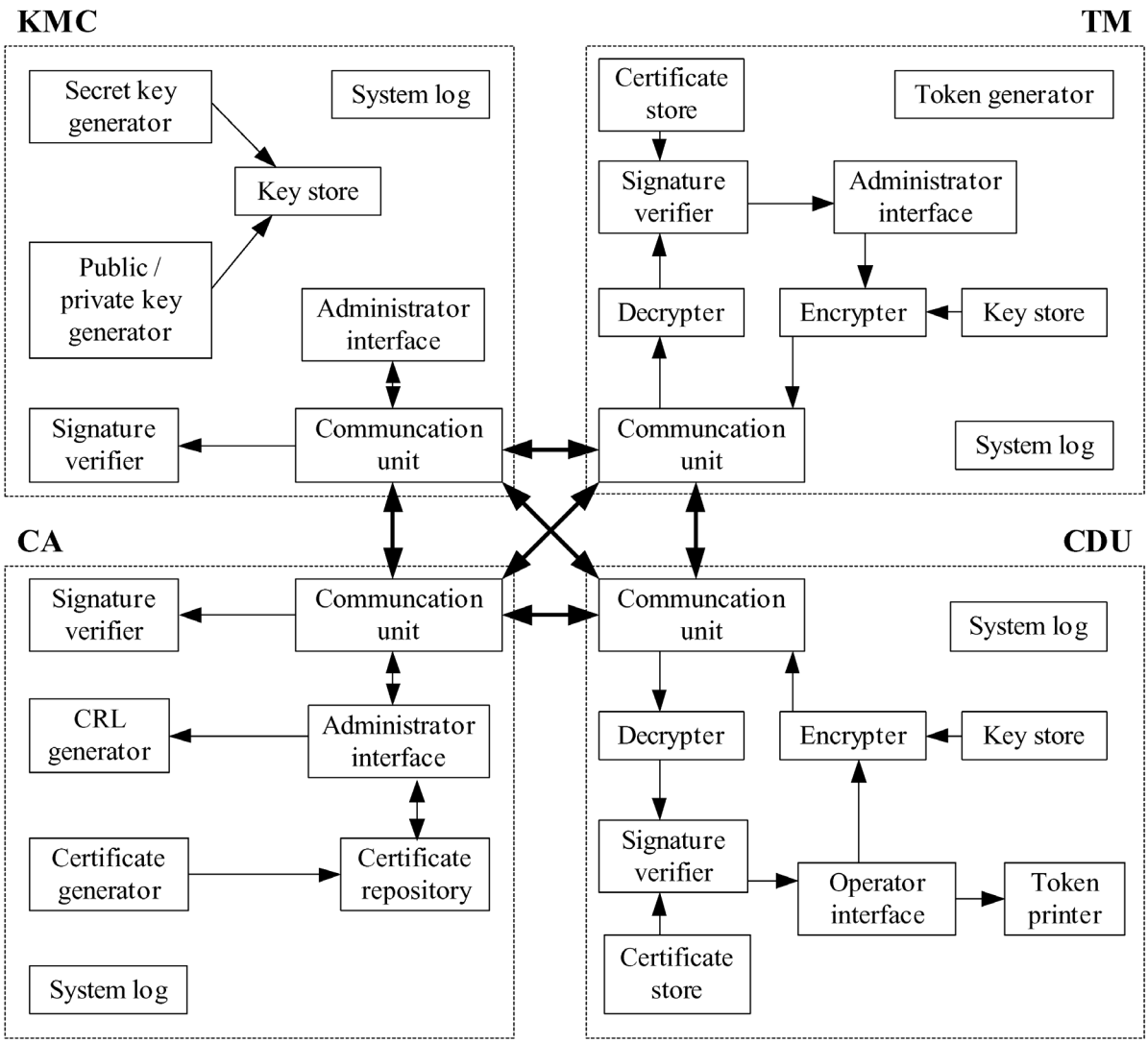

Figure 10.1: Software architecture used for the demonstration system

In Figure 10.1, the functional units of the CDU are indicated. The communication unit facilitates access to the TMs, KMC and CA. The decrypter decrypts received information, and the signature verifier ensures that signatures are valid. This is done in conjunction with the certificate store that provides the relevant certificates. The operator interface allows for requests and data from the customer to be input. The key store is where the CDU stores all of its private keys that are used when data is encrypted before being transmitted by the communication unit. The token printer is used to print the token for the customer. The system log keeps a history of the operation of the system for fault finding purposes.

10.2.3 TMs

The TM is the recipient of every prepaid electricity message sent by the CDUs and as such gives a valid response to all of the functions that the CDU can request of it. In a production implementation, the TM would communicate with higher management systems to obtain and verify customer details. The TM would also communicate with systems to obtain the actual tokens sent to the CDUs. In the demonstration system, the TM does not communicate with such systems and generates dummy tokens that are sent to the CDUs.

When a TM requires a new KDK shared with a specific CDU, SK / VK pair, or a new DK shared with a specific CDU, it requests this from the KMC. When it requires a CDU's VK certificate to authenticate a request, it obtains this from the CA, should it not already have an up-to-date copy.

The TM has an interface where one can view the prepaid electricity transactions that have passed through it, sorted by ED or CDU. This will correspond to the transaction history of each CDU.

In Figure 10.1, the functional units of the TM are indicated. Most of the components operate in the same way as in a CDU. The communication unit facilitates access to the CDUs, KMC, and CA. The decrypter decrypts received information and the signature verifier ensures that the signature is valid. This is done in conjunction with the certificate store. The token generator provides tokens once the requests have been validated. The administrator interface allows for the monitoring of the operation of the TM. The key store is where the TM stores all of its private keys that are used when data is encrypted before being transmitted by the communication unit. The system log keeps a history of the operation of the system for fault finding purposes.

10.2.4 Key Management Centre / Certificate Authority

The implemented KMC has the ability to generate all of the key types shown in the key hierarchy of Figure 6.2, i.e. MRK / MUK pairs, KDKs, SK / VK pairs, and DKs on demand as required by the CDUs and TMs. These keys are transported to the relevant

CDU or TM using the next higher key in the hierarchy. In the case of a public key pair, the public key is passed on to the CA so that a certificate indicating the relevant details of the entity to which the indicated public key has been allocated can be generated.

The implemented CA generates certificates requested by the KMC, maintains a CRL and adds keys to the CRL in response to the proper requests. The CA has an administrative interface where the certificates currently available, and the status of the certificates (valid, expired, or revoked), are indicated. It also has an interface where the most recent CRL can be obtained.

To facilitate demonstration of the system, it was decided to give KDKs and DKs an extremely short lifetime of only 10 minutes, and SK / VK pairs a lifetime of 20 minutes so that it is easy to demonstrate the process that occurs when keys expire and have to be renewed.

In Figure 10.1, the functional units of the KMC and CA are indicated. The communication units facilitate access to the TMs and CDUs. The secret key generator and public / private key generator manufacture keys as required. The signature verifiers are used to check that received messages are indeed valid. The CRL generator keeps the CRL up to date, and the certificate generator creates certificates as the KMC creates new keys. In both the KMC and the CA, the administrator interface allows observation of the current operation and changes to the key store and certificate repository. The system log records the operation of the KMC and CA systems for fault finding purposes.

10.2.5 Environment

Linux was chosen as the operating system as open source implementations of cryptographic libraries and X.509 PKI are readily available. PHP v 5.0.5 was selected as the scripting language as it has built-in support for XML/SOAP, provides access to the required cryptographic functions, and readily allows access to networking functionality, including HTML – all of which are required to make the demonstration system functional.

10.2.6 Result of demonstration implementation

The operation of the implementation was exactly as expected – the functions that were implemented operated in the intended manner. The messages that were used have the formats indicated in Chapter 8 and are secure as shown in the previous chapter.

10.3 HOW THE GOALS OF THE PROPOSED SYSTEM WERE MET

The proposed CVS was designed to implement the functionality required. The way in which each goal was met in the design is now indicated.

10.3.1 Security Modules must be stored securely

In the off-line vending system, every CDU had to have an SM in order to vend in a timely manner. By making the system on-line, the SMs can be centrally located in a secure building. The feasibility of guarding all of the SMs effectively was thus greatly increased. This became possible because the CDU must communicate with a TM for every transaction.

10.3.2 Transaction data must be kept synchronised

Because the vending system was off-line, it the time from the occurrence of a transaction until the applicable TM had knowledge of the transaction could vary considerably. By making the system on-line, the TM is informed of the transaction *before* it is completed and thus the information that the TM has cannot be outdated.

10.3.3 Financial exposure must be limited

Due to the implementation of the PKI, the ability of a CDU to vend once it has been reported as being stolen is limited to a maximum amount of time that is equal to the regularity with which CRL updates are issued. This is in contrast to the previous system of off-line vending where a stolen CDU had the ability to vend indefinitely.

10.3.4 Financial intelligence must be centralised

The relation between money and the amount of electricity that can be bought for it has to be programmed into each CDU individually when the system is off-line. By making CVS on-line, the relation is limited to just the TMs. This allows changes in the pricing structure to be implemented much more easily.

10.3.5 Tokens must be rapidly provided

Changing the system to on-line increases the difficulty with which tokens are provided to customers. Now a CDU, a TM, possibly a CA, and the communication between them must be functioning properly. Given the ubiquity and reliability of IP today, this goal will be achievable for a very high percentage of the time.

10.3.6 Parties must be accountable

In off-line vending, because the CDUs could generate tokens independently, the onus was on the CDU to report that a token had been sold and to be accountable for paying the corresponding fee. In on-line vending, through the use of digital signatures, a party can be held legally accountable when something is signed by it. Because of this, in combination with the CDU not being able to generate a token independently, a third party will be able to arbitrate, given the evidence provided.

10.3.7 The Internet must be used as the communication mechanism

The communication was specifically designed to operate using TCP/IP with all of the necessary security functions.

10.4 ATTACKS ON THE PROPOSED CVS

A list of attacks that are possible on the currently used CVS was given in Table 3.1. All of these attacks are no longer possible in the proposed CVS as explained below.

Since none of these attacks is now possible, the security provided by the proposed CVS meets the requirements for successful deployment that the existing CVS lacks.

10.4.1 Confidentiality

All sensitive user information transmitted within the on-line CVS is encrypted using session keys and so is not available to be analysed for attack.

Transaction logs are no longer transmitted between CDU and TM as the communication for each individual transaction occurs separately. Thus it is no longer possible to guess which CDU has the largest amount of cash on the premises by analysing transaction logs. However, the number of transactions that occur will be visible by observing the traffic between CDU and TM.

10.4.2 Authentication

Each entity within the proposed CVS is assigned private keys for authentication purposes. These private keys are used digitally to sign data that the entity transmits and so every entity can be held accountable for the data that it generates.

10.4.3 Integrity

Messages sent between TM and CDU can no longer be modified without detection. Any alteration of the contents of messages will be detected by the digital signature no longer being valid. Any party that has access to the CA can check the integrity of messages.

It is no longer possible for the same prepaid electricity credit to be sold multiple times. This is because the CDUs no longer need to have SMs on the premises.

10.4.4 Non-repudiation

When a token is requested by a CDU, the request must be digitally signed. This makes it impossible for a CDU to generate a fake request for which it cannot be held financially accountable.

10.4.5 Access control

The CDU can no longer generate tokens independently and access control is performed by checking the authentication of parties that request tokens. Thus, invalid parties will not be able to request tokens successfully.

10.4.6 Availability

The only weak point in the proposed CVS is the lack of on-line communication when a transaction must occur. This is addressed by ensuring that sufficient reliable bandwidth between all of the entities in the proposed CVS is available.

10.5 CURRENT SYSTEM EVALUATED IN TERMS OF X.810

The currently deployed system was compared against the X.810 framework to determine exactly where the system lacked security functionality. This was later used to compare the currently deployed CVS with the proposed CVS.

10.5.1 Authentication framework

The current CVS is susceptible to both replay and relay attacks. In the current CVS, the same tokens can be sold multiple times, but only be paid for once. This is replay. Also, because the communication between CDU, SMS and TM is off-line, it is relatively easy to interrupt communication.

10.5.2 Access control framework

The existing system is vulnerable to the access control threat of unauthorised use. Because the CDUs have the ability to generate tokens independently using the SMs, unauthorised use of a part of the system can occur.

10.5.3 Non-repudiation framework

In the current CVS, attacks on non-repudiation of communication between CDU, SMS and TM are unlikely. The focus of an attack makes sense only when a third party benefits by receiving generated tokens. There is thus no reason to fabricate data that is sent from a

CDU to an SMS. Because the process of communicating sales data is essentially manually driven, it is possible that evidence will be lost.

10.5.4 Confidentiality framework

Information in the existing system is not hidden or encrypted when in transit. It is thus possible, should this information be intercepted, to interpret it. However, because the information is essentially transmitted manually, the effort that would be expended in obtaining the data would probably outweigh the benefit that a third party would derive from it.

10.5.5 Integrity framework

Data that is uploaded from CDU to SMS to TM has appropriate checksums to ensure that unauthorised creation, modification, deletion, insertion, and replay do not occur.

10.5.6 Security audit and alarms framework

The architecture of the existing CVS does not lend itself to being audited. Information is not available in real-time. Also, because the authentication mechanisms are not sufficient, the reliability of the data that is available cannot be assumed to be very high. For this reason, analysis of data in the system to any level of certainty is difficult.

10.6 PROPOSED SYSTEM EVALUATED IN TERMS OF X.810

The proposed CVS was compared against the X.810 framework in order to ensure that all of the required security functionality had been provided for. Each part of the framework is briefly discussed below.

10.6.1 Authentication framework

The proposed CVS is immune to replay and relay attacks due to the digital signatures and supporting PKI that are used. The prepaid electricity transactions contain transaction IDs which make replay impossible.

10.6.2 Access control framework

Unauthorised operations cannot occur in the proposed CVS as every transaction request and response are digitally signed to ensure that the party that transmits the message can be held accountable for the message. Unauthorised use is prevented by incorrectly signed messages being ignored. No sensitive customer information is kept on the CDUs and thus the only data that must be protected on a CDU is the CDU's keys. Data cannot be modified in transit as the digital signatures will no longer be valid. Denial of service is prevented by the firewall between the TM cluster and the Internet that prevents unsolicited traffic from reaching the TMs.

10.6.3 Non-repudiation framework

Non-repudiation is provided by means of digital signatures. Because the reliability of the digital signatures depends on the safekeeping of the keys used to create them, suitable key hierarchies and distribution techniques have been deployed so that the non-repudiation in the system is of sufficient quality.

10.6.4 Confidentiality framework

Access to data is prevented by encrypting sensitive data. The keys that are used for the encryption of data are protected in a manner similar to the keys used for digital signatures. Suitable encryption algorithms were selected to ensure that the encrypted data remains confidential.

10.6.5 Integrity framework

Again, digital signatures are used to ensure the integrity of data that has been signed by ensuring data integrity through detection. This prevents unauthorised creation, modification, deletion, insertion, and replay. Should data be modified, the digital signature will no longer be valid and the data will be ignored.

10.6.6 Security audit and alarms framework

The architecture of the system facilitates the on-line gathering of information in the system as all data is collected at a centralised location. This allows the analysis of trends over the entire system to be performed easily and very soon after the relevant transactions have occurred.

10.6.7 Comparison of the current CVS and the proposed CVS

In comparing the current and the proposed CVS to the X.810 frameworks, the following was noted:

- The current CVS is susceptible to both replay and relay attacks, whereas the proposed CVS has measures in place that prevent such attacks from occurring without being detected and consequently ignored.
- Because the existing CVS has SMs deployed in the field, it is very susceptible to unauthorised access to these SMs should they be stolen. The architecture of the proposed CVS, however, makes provision that no SMs need be present in the field, and thus it is significantly more robust to attacks in terms of unauthorised access.
- Both the current and the proposed CVS are resistant to non-repudiation attacks.
- The current CVS is susceptible to confidentiality attacks, although such attacks are difficult as the system does not use on-line communication. The proposed CVS has the necessary measures in place to prevent confidential information from being obtained.
- Both the current and the proposed CVS have measures in place to ensure integrity of data.
- The current CVS, primarily because it is off-line, does not lend itself to being audited. The proposed CVS, on the other hand, is structured in such a way that auditing at the centralised database can be implemented with little difficulty.

The current CVS is susceptible to authentication, unauthorised access, and confidentiality attacks. Effective auditing in the current CVS is difficult. All of these shortcomings have been removed in the proposed CVS, making the proposed CVS superior in terms of security.

10.7 CONCLUSION

This chapter has described the demonstration system, the manner in which the goals in the proposed system were met and the manner in which attacks on the proposed CVS are thwarted.

The demonstration system, although not a production-ready implementation, provides enough functionality so that the operation of the system can be viewed. The implementation has two aspects that allow its operation to be examined – detailed log files and interfaces to view various pieces of information on each of the components.

The manner in which all of the goals of the proposed CVS that were described in Chapter 3 are met were individually explained and motivated. It was then shown that all of the required security functions have been provided. The proposed CVS provided confidentiality, authentication, integrity, non-repudiation, access control, and availability.

The current CVS and the proposed CVS were both compared to the X.810 security framework series. It was found that the current CVS lacked functionality in every framework except the integrity framework. In comparison, the proposed CVS provided all of the required functionality in every framework, and was immune to the threats that were analysed. Thus, it has been shown that the security shortcomings that the current CVS contains, have been alleviated in the development of the proposed CVS.

Eleven

CONCLUSION

11.1 INTRODUCTION

This final chapter provides a brief overview of the reasons why the current CVS is not sufficient and a proposal for an improved version that provides the required functionality. The security that is provided by the proposed system is shown. Finally, ideas for future extensions to this work are given.

11.2 SYSTEM REQUIREMENTS

There are a number of a problems in the off-line implementation of prepaid electricity as currently implemented using STS and CVS. These problems include the vulnerabilities of illegitimate tokens being generated should a CDU be stolen, up to difficulties in implementing new cost structures.

A system was required that could solve these problems and also hold vending parties accountable for tokens that have been sold in a legally enforceable manner. The requirements of such a system were described in terms of seven goals at in Chapter 3 and were briefly:

1. Security Modules must not be exposed in the field.
2. Transaction data must be synchronised with the management system.
3. Financial exposure must be limited.
4. Intelligence of the system must not be distributed.
5. Customers must perceive performance as being instantaneous.
6. Proof must exist for financial obligation.
7. The Internet should be used as the communication medium.

11.3 PROPOSED SYSTEM

The proposed CVS is an on-line system (as opposed to the currently used off-line system) where the intelligence of issuing tokens is no longer in the field, but at a centralised location. Every time a transaction occurs, the CDU in the field must communicate with the TM in order to obtain the requested token. By doing this, the independence is removed from the CDU, and the system is centralised at the TM.

In terms of security, the proposed CVS has a key hierarchy that was specifically designed to meet the confidentiality requirements of the various types of communication. Each CDU and TM has a master key pair that is used to prove its identity. The key encrypting key, the DK, and the signing key pair are dependent on this master key pair.

A PKI is needed to support the public and private keys indicated above. A suitable PKI was developed that met the authentication requirements. The procedures involved in setting up each component of the proposed CVS, so that the security functionality can be implemented, were discussed.

Finally, the messages themselves were devised to meet the functions required for the prepaid electricity, the key management, and the PKI. The structure and content of these messages were analysed to show that the required functions were performed but that the security of the system was not compromised.

11.4 SECURITY PROVIDED BY THE PROPOSED SYSTEM

As has been shown in the previous two chapters, the proposed CVS meets the goals listed in Section 3.7 and provides all of the required security functionality.

The proposed CVS provides confidentiality and integrity. The data internal to the system is protected from disclosure as a measure to ensure that the system is not vulnerable to security attacks. Customer data is also protected to ensure that it is not available to

malicious parties. The integrity ensures that the tokens that customers receive are valid and will work as expected.

The system provides authentication and non-repudiation. These are crucial to the successful operation of the system as without them, financial accountability cannot be maintained.

The proposed CVS also provides access control and ensures availability. These measures ensure that the proposed CVS will give customers an acceptable level of service.

11.5 CRITICAL ASSESSMENT OF RESEARCH

Given the problem that has been described in the shortcomings of the existing system used for the vending of prepaid electricity, a solution was found. This solution meets all of the security requirements so that the proposed system can replace the existing system.

The contribution to the knowledge in the field of vending prepaid electricity is:

- the analysis of shortcomings in the existing system,
- the development of a new system that meets the necessary security requirements,
- the development of the necessary supporting infrastructure in terms of PKI for the new system,
- the development of the messages required for communication in the new system, and
- the analysis of the new system showing that the security requirements were met with reference to the X.810 security framework.

The proposed system that was developed is sufficient in terms of security to replace the entire system currently in use. However, the system that was developed is not sufficient to be used as a replacement as described, but requires additional work to make it an operational production system (see the next section).

11.6 FUTURE EXTENSIONS TO THIS WORK

The work that has been described in this document in only the initial work in providing the necessary security for an on-line CVS. Additional work that should be done to make this into a deployable system is as follows:

- The implementation of a production version of the system that has been described. This is necessary to have an implementation of the system that can be used to replace the existing CVS.

- The development of alternatives to replace CDUs in the on-line CVS as an access point for customers to use. Possibilities for this include ATMs, cellular telephones used by agents, and direct access via the Internet. The latter two options will have to provide some means of receiving and handling money.

- The detailed examination of last mile access to the Internet. In this document, it has been assumed that an IP pipe is available at the site of every CDU and TM. Various means of providing this could be examined such as VSAT Internet access, GRPS access, cost effective dial-up access, etc.

- The expansion of the design of the system to work on a world wide basis. The design in this document was only aimed at a single country on the scale of South Africa. International operation will have several challenges associated with it, similar to GSM roaming.

11.7 CONCLUSION

The purpose of designing and implementing a prepaid electricity system that makes use of the Internet and meets the goals stated in Chapter 3 has been successfully achieved. All of the appropriate security aspects have been taken into account. The result is the system that has been described and analysed in this document.

REFERENCES

[1] B Mokgele, *Eskom prepayment strategy and future direction*, South African Prepayment Week, Johannesburg, South Africa, June 2003.

[2] R Kaplan, *Standard Transfer Specification guidelines*, Measurement and Control Department, National PTM&C, Transmission Group, Eskom, 1995.

[3] STS Association, *Electricity Sales Systems – Part 6: Interface standards – Section 6: Standard transfer specification / credit dispensing unit – Electricity dispenser – categories of tokens and transaction data fields*, Rationalized User Specification, Edition 1.0, NRS 009-6-6:1998, 1998. Available: http://www.nrs.eskom.co.za/nrs/specifications/nrs%20009-6-6%20(1.1).pdf, last accessed 2005/10/20.

[4] STS Association, *Electricity Sales Systems – Part 6: Interface standards – Section 7: Standard transfer specification / Credit dispensing unit – Electricity dispenser – Token encoding and data encryption and decryption*, Rationalized User Specification, Edition 2.2, NRS 009-6-7:2002, 2002. Available: http://www.nrs.eskom.co.za/nrs/specifications/nrs%20009-6-7%20(2.2).pdf, last accessed 2005/10/20.

[5] STS Association, *Electricity Sales Systems – Part 6: Interface standards – Section 8: Standard transfer specification / Disposable magnetic token technology – Token encoding format and physical token definition*, Rationalized User Specification, Edition 2.0, NRS 009-6-8:2004, 2004. Available: http://www.nrs.eskom.co.za/nrs/specifications/nrs%20009-6-8.pdf, last accessed 2005/10/20.

[6] STS Association, *Electricity Sales Systems – Part 6: Interface standards – Section 9: Standard transfer specification / Numeric token technology – Token encoding format and physical token definition*, Rationalized User Specification, Edition 1.0, NRS 009-6-9:1997, 2002. Available: http://www.nrs.eskom.co.za/nrs/specifications/nrs%20009-6-9%20(1.0rec).pdf, last accessed 2005/10/20.

[7] STS Association, *Electricity Sales Systems – Part 7: Standard transfer specification / The management of cryptographic keys*, Rationalized User Specification, Edition 1.0, NRS 009-7:1999, 1999. Available: http://www.nrs.eskom.co.za/nrs/specifications/nrs009-7.pdf, last accessed 2005/10/20.

[8] Syntell Networks (Pty) Ltd, *Synapse Vending System*, 2002. Available: http://www.syntell.co.za/document/productpdf/Synapse.pdf, last accessed 2006/02/28.

[9] EasyPay web site: http://www.easypay.co.za, last accessed 2006/02/28.

[10] Prism TranSwitch Services web site: http://www.prism.co.za, last accessed 2006/02/28.

[11] W Ford and MS Baum, *Secure Electronic Commerce – Building the Infrastructure for Digital Signatures and Encryption*, Prentice Hall, 2001.

[12] W Stallings, *Cryptography and Network Security – Principles and Practice*, Second edition, Prentice Hall, 1999.

[13] Thawte web site: http://www.thawte.com/, last accessed 2005/10/21.

[14] VeriSign web site: http://www.verisign.com/, last accessed 2005/10/21.

[15] ITU-T Recommendation X.509 version 3 (1997), *Information Technology - Open Systems Interconnection - The Directory Authentication Framework*, ISO/IEC 9594-8:1997.

[16] ITU-T Recommendation X.810 (1995), *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Overview*, ISO/IEC10181-1:1996.

[17] ITU-T Recommendation X.811 (1995), *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Authentication Framework*, ISO/IEC10181-1:1996.

[18] ITU-T Recommendation X.812 (1995), *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Access Control Framework*, ISO/IEC10181-3:1996.

[19] ITU-T Recommendation X.813 (1996), *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Non-repudiation Framework*, ISO/IEC10181-4:1997.

[20] ITU-T Recommendation X.814 (1995), *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Confidentiality Framework*, ISO/IEC10181-5:1996.

[21] ITU-T Recommendation X.815 (1995), *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Integrity Framework*, ISO/IEC10181-6:1996.

[22] ITU-T Recommendation X.816 (1995), *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Security Audit and Alarms Framework*, ISO/IEC10181-7:1996.

[23] AO Freier, P Karlton, PC Kocher, *The SSL Protocol, Version 3.0*, http://wp.netscape.com/eng/ssl3/, Nov. 1996, last accessed 2005/10/21.

[24] G Apostolopoulos, V Peris, P Pradhan and D Saha, *Securing Electronic Commerce: Reducing the SSL Overhead*, IEEE Network, vol. 14, no. 4, pp. 11 – 13, Jul. / Aug. 2000.

[25] T Bray, J Paoli, CM Sperberg-McQueen and E Maler, *Extensible Markup Language (XML) 1.0 (Second Edition)*, World Wide Web Consortium, http://www.w3.org/TR/2000/REC-xml-20001006, Oct. 2000, last accessed 2005/10/21.

[26] S Widergren, A deVos, J Zhu, *XML for Data Exchange*, IEEE Power Engineering Society Summer Meeting, vol. 2, pp. 840 – 842, Jul. 1999.

[27] N Mitra (editor), *SOAP Version 1.2 Part 0: Primer*, World Wide Web Consortium, http://www.w3.org/TR/2003/REC-soap12-part0-20030624, Jun. 2003, last accessed 2005/10/21.

[28] T Imamura, B Dillaway and E Simon, *XML Encryption Syntax and Processing*, World Wide Web Consortium, http://www.w3.org/TR/xmlenc-core/, Dec. 2002, last accessed 2005/10/21.

[29] DE Eastlake, JM Reagle and D Solo, *XML-Signature Syntax and Processing*, World Wide Web Consortium, http://www.w3.org/TR/2001/PR-xmldsig-core-20010820, Aug. 2001, last accessed 2005/10/21.

[30] M Gudgin, M Hadley, N Mendelsohn, JJ Moreau, H. F. Nielsen (editors), *SOAP Version 1.2 Part 1: Messaging Framework*, World Wide Web Consortium, http://www.w3.org/TR/2003/REC-soap12-part1-20030624, Jun. 2003, last accessed 2005/10/21.

[31] M Naedele, *Standards for XML and Web Services Security*, IEEE Computer, vol. 36, no. 4, pp. 97 – 98, Apr. 2003.

[32] Sun Microsystems, *Security and the Java Platform*, Sun Developer Network. Available: http://java.sun.com/security/, last accessed 2005/11/15.

[33] B. Kaliski, *TWIRL and RSA Key Size*, RSA laboratories. Available: http://www.rsasecurity.com/rsalabs/node.asp?id=2004, May 2003, last accessed 2005/10/21.

[34] W. Diffie, *The First Ten Years of Public-Key Cryptography*, Proceedings of the IEEE, vol. 76, pp. 560 – 577, May 1988.

Addendum A: SAMPLE MESSAGES

This addendum contains sample SOAP messages as used in the "verify customer details" and "purchase token transactions". In the sample messages, encryption is not used so that the contents of the messages can be understood easily. In normal operation, the latter part of the messages will be encrypted, as indicated in Section 8.5.

The first message shown, in Figure A.1, is the message used when the CDU requests the customer details to confirm that a token is to be bought for the correct customer. The second message, in Figure A.2, is the message returned from the TM that indicates the customer's details associated with the particular ED. The contents of these messages are taken from Section 8.4.3.

The third message, in Figure A.3, request a token from the TM for a specific ED for the indicated amount. The fourth message, in Figure A.4, provides the token that the customer (and hence the CDU) requested. The contents of these messages are taken from Section 8.4.1.

(The indentation is added to the SOAP messages for readability only – it is not a requirement for successful operation and is removed before processing.)

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:vending xmlns:m="http://sts.org.za/vending/cvs2.0"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2005-11-29T13:20:10.320+02:00</m:dateAndTime>
    </m:vending>
    <n:source>
      <n:type>cdu</n:type>
      <n:id>4322-6453-6234-1234</n:id>
    </n:source>
    <n:destination>
      <n:type>tm</n:type>
      <n:id>1231-2342</n:id>
    </n:destination>
    <n:customer xmlns:n="http://sts.org.za/customer"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <n:edid>1234-4312-2345-4231</n:edid>
    </n:customer>
  </env:Header>
  <env:Body>
    <p:transaction
        xmlns:p="http://sts.org.za/transaction">
      <p:requestcustomerdetails>
        <p:status>confirm_customer</p:status>
      </p:purchaserequest>
    </p:transaction>
    <p:signature xmlns:p="http://sts.org.za/rsa_sha1_sig"
      <p:value>42342b2a…23ef8b</p:value>
    </p:signature>
  </env:Body>
</env:Envelope>
```

Figure A.1: Message requesting customer details

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:vending xmlns:m="http://sts.org.za/vending/cvs2.0"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2005-11-29T13:20:10.832+02:00</m:dateAndTime>
    </m:vending>
    <n:source>
      <n:type>tm</n:type>
      <n:id>1231-2342</n:id>
    </n:source>
    <n:destination>
      <n:type>cdu</n:type>
      <n:id>4322-6453-6234-1234</n:id>
    </n:destination>
    <n:customer xmlns:n="http://sts.org.za/customer"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <n:customerid>AB43-23D1-0987-DE23</n:customerid>
      <n:name>Koos van der Merwe</n:name>
      <n:address>Plot 91, Hammanskraal</n:address>
      <n:system>STS on-line</n:system>
      <n:edid>1234-4312-2345-4231</n:edid>
    </n:customer>
  </env:Header>
  <env:Body>
    <p:transaction
        xmlns:p="http://sts.org.za/transaction">
      <p:confirmdetails>
        <p:status>confirm_customer</p:status>
      </p:confirmdetails>
    </p:transaction>
    <p:signature xmlns:p="http://sts.org.za/rsa_sha1_sig"
      <p:value>c2e12b2b…12eaab</p:value>
    </p:signature>
  </env:Body>
</env:Envelope>
```

Figure A.2: Message providing customer details

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:vending xmlns:m="http://sts.org.za/vending/cvs2.0"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j82</m:reference>
      <m:dateAndTime>2005-11-29T13:21:11.145+02:00</m:dateAndTime>
    </m:vending>
    <n:source>
      <n:type>cdu</n:type>
      <n:id>4322-6453-6234-1234</n:id>
    </n:source>
    <n:destination>
      <n:type>tm</n:type>
      <n:id>1231-2342</n:id>
    </n:destination>
    <n:customer xmlns:n="http://sts.org.za/customer"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <n:edid>1234-4312-2345-4231</n:edid>
    </n:customer>
  </env:Header>
  <env:Body>
    <p:transaction
        xmlns:p="http://sts.org.za/transaction">
      <p:confirmation>
        <p:value>20.00</p:value>
        <p:currency>ZAR</p:currency>
        <p:status>purchase_token</p:status>
      </p:confirmation>
    </p:transaction>
    <p:signature xmlns:p="http://sts.org.za/rsa_sha1_sig"
      <p:value>2e4d98cc…a2cc21</p:value>
    </p:signature>
  </env:Body>
</env:Envelope>
```

Figure A. 3: Message containing purchase request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:vending xmlns:m="http://sts.org.za/vending/cvs2.0"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j82</m:reference>
      <m:dateAndTime>2005-11-29T13:21:11.732+02:00</m:dateAndTime>
    </m:vending>
    <n:source>
      <n:type>tm</n:type>
      <n:id>1231-2342</n:id>
    </n:source>
    <n:destination>
      <n:type>cdu</n:type>
      <n:id>4322-6453-6234-1234</n:id>
    </n:destination>
    <n:customer xmlns:n="http://sts.org.za/customer"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
      <n:customerid>AB43-23D1-0987-DE23</n:customerid>
      <n:name>Koos van der Merwe</n:name>
      <n:address>Plot 91, Hammanskraal</n:address>
      <n:system>STS on-line</n:system>
      <n:edid>1234-4312-2345-4231</n:edid>
    </n:customer>
  </env:Header>
  <env:Body>
    <p:transaction
        xmlns:p="http://sts.org.za/transaction">
      <p:token>
        <p:value>20.00</p:value>
        <p:currency>ZAR</p:currency>
        <p:status>confirm_customer</p:status>
        <p:token>4321543480192543234</p:token>
      </p:token>
    </p:transaction>
    <p:signature xmlns:p="http://sts.org.za/rsa_sha1_sig"
      <p:value>10dd092e…17edac</p:value>
    </p:signature>
  </env:Body>
</env:Envelope>
```

Figure A.4: Message providing purchased token