CRYPTOGRAPHIC KEY MANAGEMENT PRINCIPLES

APPLIED IN SOUTH AFRICAN INTERNET BANKING

Emile Parkin

NamITech Security Solutions

NamITech (PTY) Limited

www.namitech.com

emile.parkin@namitech.com +27 (0) 11 458 0000

ABSTRACT

The convenience of Internet Banking and the breadth of functionality that it provides to its users have made it exceptionally popular, especially in countries like South Africa. Gone are the days of standing in long queues in the bank just to authorise a debit order or to get an account statement. But where accountholders in the past had to enter a secret PIN into a closed and secure system (e.g. ATM or Bank Branch system), these secrets must now be communicated through the insecure Internet. New threats and vulnerabilities within operating systems and Internet applications are published daily and the obvious question becomes apparent: Is it safe to use Internet Banking applications?

In this paper, the current architecture of Internet Banking is re-evaluated, with specific focus awarded to the cryptographic security controls implemented in such systems. Since the current sense of security is primarily based on the premise of cryptography, it is appropriate to assess if best practice principles and standards of cryptography and key management have been applied, and to what extend. Furthermore, we assess the value of applying key management principles to a PIN (or password) as if it is a cryptographic key. Through this exercise, it becomes clear that the use of a static secret value to uniquely authenticate a user is not a secure mechanism and it is not appropriate for authentication over the Internet. Possible solutions are also provided as guidelines in addressing this issue.

KEY WORDS

Internet Banking, Cryptography, Key Management, Key Life Cycle, SSL, Passwords

1 INTRODUCTION

According to recent studies, the average South African Internet Banking user is employed, between 25 and 36 years of age, has at least some tertiary education (past matric) and earns more than R15,000 per month [1]. This is a limited subset of the bank's customer base, and most of these users are educated business people. The same studies have shown that security is a major influence on usage, together with customer support [1]. However, the majority of these users are not technical, and would not be able to understand the real risks of using the Internet for banking unless it is publicly communicated. The lack of security awareness, and the fact that insecure systems and components form part of the Internet Banking application architecture, are just some of the major concerns.

Internet Banking applications have been widely used for more than five years, and much trust in these systems have been based on the use of cryptography to secure the user's PIN and/or Password from where it is entered on the user's PC. While data is secured in transit within the Secure Socket Layer (SSL), the concern lies at the two end points. It cannot be assumed that a Password is secure until it is encrypted in the SSL session, or that it will be secure once it exits the SSL session on the other side.

This paper examines the cryptographic architecture used within a typical Internet Banking application, and within the context of key management and cryptography best practices and standards. The objective is to gain an understanding of the risks involved in using Internet Banking applications.

2 OVERVIEW OF INTERNET BANKING

It is worth reviewing the components that are involved in any Internet Banking application, as this information will be recalled later in this paper. It should be noted that this is considered as the 'typical' architecture and functionality, and it can be assumed that this will vary for each bank. The purpose of this overview is to define a baseline architecture that will be studied in more detail, and to establish a common understanding of the functionality that is part of most Internet Banking applications.

2.1 Functional Overview

Typical Internet Banking services include:

1. Balance enquiries
2. Account Statements
3. Beneficiary Management (create, modify and remove)
4. Stop and Debit payments
5. Manage Investment accounts
6. Funds Transfer
7. Increase/Decrease overdraft
8. Change user settings

More advanced services that have been added to some sites include:

1. Online Applications for loans
2. Management Home loans and Vehicle Finance Accounts

2.2 Component Overview

The standard architecture for the South African Internet Banking sites typically includes the following components, which will be discussed in more detail in later paragraphs:

Table 1: Components of Internet Banking

Component	Description
User	As a registered subscriber to the Internet Banking web-site, the user is required to authenticate himself using an account number and some secret, in most cases a PIN and/or Password
Keyboard / Mouse	The input devices used to enter the user's login credentials.
Internet Browser Application	Application that loads and displays HTML code and scripts received from the Internet Banking site. This application also manages SSL sessions between the client and web server.
Network Infrastructure	A logical grouping of all the network components required to transfer data from the client to the web server hosting the Internet Banking web pages. This may include: client network card, cables, Internet Service Provider infrastructure, firewalls, routers, IDS/IPS sensors and load balancing equipment. Also, no limitation is placed on the transport medium used, for example cable or wireless connections, or the network layer protocols (e.g. GPRS, 3G, Wi-Fi).
Web Server	A server that hosts the web-pages through a web server application.
Application Server	While it is possible to incorporate all of the business intelligence onto the web server, it is considered best practice to separate the business logic from the web service logic. In this case, the application server runs the Internet Banking application, while the web server is only concerned with hosting web pages.
Mainframe	Banking customer account information is stored on mainframe systems, which manages all of the bank's core business information.

3 CRYPTOGRAPHIC KEY MANAGEMENT

Most of the security controls employed by Internet Banking applications are dependent on cryptography, and therefore also dependent on secret keys. In fact, the only way to extend the security boundary right to the user's PC is to encrypt all sensitive traffic between this PC and the bank's infrastructure. In this section we will review the standards and best practices of key management, and understand how they have been applied to Internet Banking keys.

Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties [4], or more simply; the securing of keys [5]. While cryptography has proven to be a very efficient security control, by itself it doesn't have any value unless the keys are adequately protected [6]. In essence, the security level of any cryptographic solution or architecture lies in the protection of the keys relevant to such a system.

Key management is not merely concerned with technological controls as it requires a combination of:

1. Technical controls, such as hardware and security solutions
2. Physical controls, where a safe, secure room and alarm systems may be employed, and
3. Procedural controls, which requires human interaction [4] [5]

The type of key that is to be protected defines the level of security required to protect this key. For example, a session key only lasts for the duration of a session (typically between a client and server), and therefore requires less security and protection than a master key, which may be used to encrypt various other working keys.

A framework for key management should include these controls, and must specify attributes and events associated to each key. These aspects of key management will be discussed in following sections.

3.1 Key Life Cycle Events

In order to *manage* keys, it is critical to first understand the life cycle of a key. The life cycle for symmetrical keys will differ from that of asymmetrical keys, as symmetrical schemes only use one key, opposed to asymmetrical schemes which use both a public and a private key pair [4].

The following essential events form part of the key's life cycle [4][5]:

Key Generation	This is where the key is first generated, and it signifies the creation of the key
Key Distribution	Since the key may not necessarily be used within the same system where it had been generated, it has to be distributed to other systems. For symmetric keys this is always the case, since the key must be communicated to at least one other point.
Key Loading	Once the key has been distributed to the systems that will use it, it must be loaded. In some cases this is achieved by manually loading/entering the key into a hardware security module
Key Backup	It might be a requirement to also backup the key into a secure environment
Key Usage	This is where the key will be used in cryptographic algorithms as part of a solution.
Key Storage Environment	Whenever the key is not used, it must be stored securely. A key can be stored on a normal storage medium, or on a cryptographic token, like a smart card or Hardware Security Module
Key Archive	Once the key has been decommissioned and is no longer in use, it could also be archived for future reference.
Key Destruction	A key is deleted or physically destroyed

Another event that could be added to this is *Key Renewal*, which occurs when an old key is destroyed and a new key is created to replace it.

For Asymmetric keys, the security requirements and the key life cycle controls associated with public keys opposed to private keys are much different. This is especially true for key distribution and loading, where public keys can be distributed to multiple parties without any concern for confidentiality but private keys must only be loaded into one system.

3.2 Key Attributes

Whereas the key life cycle outlines the events that are relevant to a key, the attributes of a key determines the level of security required to protect the key. A key can have the following attributes:

Table 2: Key Attributes

Key Attribute	Description
Algorithm	The algorithm that will accept this key as an input parameter, for example DES, AES, RSA, etc.
Key Length	Length of the key, which to a greater extend defines the strength of the key.
Key Usage	This attribute defines how the key will be used, and it is important that a single key is not used for too many different functions [6]. Examples include: • Encryption • Decryption • Hashing • Digital Signatures (a combination of hashing and encryption) • Key Encryption / Decryption

Key Attribute	Description
Authorization Requirements	Access control should be applied so that only authorised users can use the key, and only for its intended purpose.
Key format	While only the full clear key value can be used by the cryptographic algorithm, the key could be transported, imported or exported in other formats, like: • Encrypted: the key encrypted with another "key encryption key" • Split components: two or more components which, if they are all XOR'ed, can rebuild the original clear key. The objective of this is to ensure that when the key must be transported manually, the full key is never exposed to one party [6]. This is also referred to as the principle of *dual control.*
Life Span	The longer a key is used within a system, the more it is exposed to various types of attacks [6]. Life span can be defined by the number of transactions, seconds, minutes, hours, days, months or even years.

4 CRYPTOGRAPHY IN INTERNET BANKING APPLICATIONS

In previous sections, a brief overview of the different components involved in Internet Banking applications provided a baseline understanding of such solutions. An overview of key management principles highlighted the importance of managing the keys that form part of these solutions. This section provides a closer examination of the cryptographic architecture and components.

4.1 Sensitive Data

In this context examples of sensitive data are the Account Number, PIN, Password and transaction information. Transaction information could be related to money transfers, account or accountholder information, user settings/preferences or requested information (such as balance and statement enquiries)

4.2 Cryptographic Zones

A cryptographic zone exists between two points, where a symmetric key or asymmetric public keys are shared in order to encrypt sensitive information. Once the key, or keys have been exchanged, data, and in some cases other keys, are encrypted within this zone.

For Internet Banking, the following cryptographic zones are relevant:

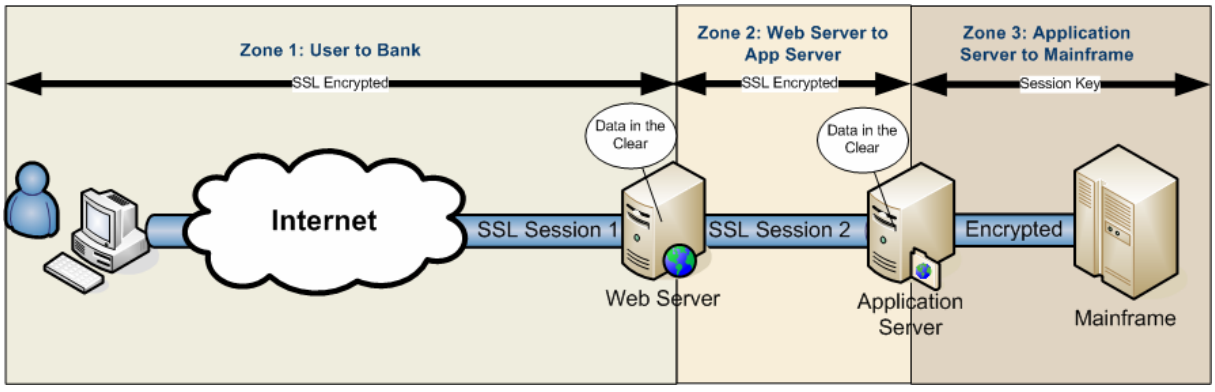


Figure 1: Internet Banking Cryptographic Zones

Zone 1: User to Bank

When the user connects to the secure (https) Internet Banking web site, the browser establishes an SSL or TLS session[1]. Sensitive information such as the account number, PIN and Password are sent through this encrypted session.

A web server typically terminates the SSL connection, but it could also be terminated on:

- A "*proxy*" server, which manages all incoming request and in some cases, routes it to different end points
- *SSL Appliance*; a hardware appliance that terminates SSL connections and sends the clear data to the next point.

The basic principle here is that there is typically a server or an appliance which terminates the SSL session and then relays the clear data to the next server.

Zone 2: Web Server to Application Server

Now that the data has progressed out of the SSL session (in Zone 1), the data is now in the clear, but must be sent to the next point, typically the application server. Some banks send this data over a clear link, while other banks send the data through a new encrypted session by either establishing a new SSL session, or transferring data through an IPSec tunnel.

Zone 3: Application Server to Mainframe

On receipt of the sensitive data, the Application Server needs to send it to the mainframe for verification. Ideally, these data elements are then encrypted with a symmetric key which has been pre-negotiated with the mainframe system.

One flaw in this approach, which is applicable to most implementations of Internet Banking web sites, is that the data is in the clear once it comes out of the SSL session. Even in cases where a Hardware Security Module is used to manage SSL connections, the clear data is always returned to the server application. This will be discussed in more detail later.

4.3 Cryptographic Keys

The following table identifies a list of keys used within an Internet Banking application, with respect to the different cryptographic zones:

Table 3: Cryptographic Keys used in Internet Banking

	Key	Usage
Zone 1: User to Bank	Asymmetric Key Pair associated with the Server's SSL Digital Certificate	Server Identification and Verification Encryption/Decryption of Session Keys within the SSL Session
	Symmetric Session Key	Used to encrypt data transferred within the same session

[1] For the purpose of this paper, Secure Socket Layer (SSL) version 3 is the selected protocol used to establish an encrypted session between the client and server, but the user or the browser may select another protocol, such as the Transmission Layer Security (TLS), or an earlier version of SSL.

	Key	Usage
Zone 2: Web Server to Application Server	Asymmetric Key Pair, associated with the Server's SSL Digital Certificate	Server Identification and Verification Encryption/Decryption of Session Keys within the SSL Session
	Symmetric Session Key	Used to encrypt data transferred within the same session
Zone 3: Application Server to Mainframe	Symmetric Zone Key	Key Encryption Key Used to encrypt working keys
	Symmetric Working Keys	Key that encrypts the sensitive data between the two servers

While a PIN or Password is not normally treated as keys, there is no reason why it shouldn't carry the same security requirements for protection and usage as a key. Furthermore, it would be extremely valuable to apply the principles of the key life cycle to a PIN, and this has been done to some extend by MasterCard [7] and VISA [8].

5 SECURITY OBJECTIVES OF INTERNET BANKING APPLICATIONS

In this section we evaluate the security objectives associated with Internet Banking Applications. Moreover, the scope will be limited to the protection of the Password[2] and other sensitive information (as defined above). As mentioned in the previous section, the Password will be considered as a key, and as such it will be evaluated according to key life cycle and attribute principles.

5.1 Identification and Authentication

During the login process, the user must first identify and authenticate the web server to ensure that communication has been established with the bank's authorised and authentic server. Secondly, the bank's web server must identify and authenticate the user.

5.1.1 Identification and Authentication of User

5.1.1.1 Current Risks

The most common way of identifying the user is through an account number and password. In the following table, the risks are highlighted by applying a key management profile to the key.

[2] The various Internet Banking applications have different terminology for PINs and Passwords, for the remainder of this paper, this "group" of secrets (static text shared between the user and the bank) will simply be referred to as "Passwords"

Table 4: Internet Banking Password Key Management Profile

	Current Implementation & Risks	Risk	Best Practice Requirements
Key Attributes			
Algorithm	Symmetric, since the same password is shared between two points.	H	A working key should be randomised to prevent replay attacks [6].
Key Length	Minimum of 6 characters, and there is a restriction on the number of retries.	M	An eight character password would be the same length as a single length symmetric key, which is considered to be a weak key [7] [8]. However, the retry limit protects against brute force attacks.
Key Usage Environment	Password is used within an insecure environment.	H	A cleartext key must be entered into a secure device, unless it is an one-time key [9] [10]
Authorization Requirements	No access control is placed on the user's password, as it can be given to anyone.	H	Only authorised users and applications may use a key [7] [8] [9]
Key format	Cleartext until it enters the SSL session	H	A cleartext key must be entered into a secure device, unless it is an one-time key [9] [10]
Life Span	In most cases, indefinite. User can use a password for years.	H	A key must have a limited life span [6] [7] [8].
Life Cycle Events			
Key Generation	Initially Passwords are generated by the system, but thereafter the user can *generate* a replacement password.	H	Keys must be generated using a random number generator [6] [7] [8].
Key Distribution	In some cases, the initial bank generated password is printed in a secure envelope and mailed to the user, but other distribution mechanisms exist.	H	Keys must be distributed either encrypted or as split components [7] [8] [10]
Key Loading	Password is entered into an untrusted / insecure system (user's PC), where it can be intercepted as a cleartext value	H	A cleartext key must be entered into a secure device, unless it is an one-time key [9] [10]
Key Backup & Archive	It is up to the user to backup the Password. If it is lost, the bank branch can reset it.	M	A secure backup should be made, and to the same level of security of the original key [7][8].
Key Usage	Password is used to authenticate user, but since the user can select a password, users typically re-use passwords for different purposes (e.g. to log into email, windows domain or other web-sites	H	Key must only be used for its intended purpose [7][8]
Key Storage	User can store is Password on a document or electronically, and it is up to the user to protect his password.	H	A key must be stored and used within the boundaries of cryptographic hardware [9]
Key Destruction	Password is useless once a new password is defined	L	Not applicable

Note: H is High, M is Medium and L is Low

It is apparent from this table that the overall risk related to the usage of static passwords in the context of key management is high. Furthermore, it is also interesting to note that the banks are not prepared to carry all of the risk [12], as it is the user's responsibility to protect the PIN.

5.1.1.2 Possible Solutions

When reviewing the table above and keeping in mind that the 'system' into which the user will enter the password will always be insecure[3], the following key points become apparent:

1. Static passwords are not good enough, and randomised or one-time passwords should be used.
2. A trusted device must be used to create a random one-time password.
3. Only the authorised user must be able to use the password, and it should not be possible for the user to distribute the password to other users.

The following technologies currently exist and could be used to meet the requirements defined above:

One-Time Passwords

One-time password generating tokens	Proprietary tokens are available from vendors such as ActivCard and RSA Security. These tokens are able to generate a one time randomised password that is verified by an authentication server.
Mobile	A few vendors have also developed one-time password generation applications that will run on mobile devices, such as mobile phones and PDAs
OATH	An Open Authentication standard which will allow any OATH compliant one-time password generating token to work with any authentication server.
Symmetric Key Infrastructure (SKI)	Similarly, proprietary tokens can use SKI to authenticate the user, where the Internet Banking application displays a generated number on the web site and the user responds with the correct response.

Public Key Infrastructure Solutions

In this case, each Internet Banking user is provided with a unique digital certificate, which can be stored on storage media or on a cryptographic token. In addition this could be combined with biometric information and a password as a three-factor authentication solution.

MasterCard Chip Authentication Program (CAP)

Major South African Banks are already in final testing phases of EMV smart card solutions, and will soon distribute these EMV cards to their customer base. By inserting this smart card into a CAP compliant smart card reader (not connected to the PC), the user is able to enter his bank PIN and generate a one-time password, which could be used as an authentication mechanism. Version 2 of the CAP standard will include an SKI implementation.

[3] The user's system is considered insecure and untrusted based on the guidelines set forth in [9]

5.2 Identification and Authentication of Web Server

There are several risks associated with the identification and authentication of the web server, but it is interesting to note that one of the most significant risks is security awareness of users. For the user to identify and authenticate the web server, he is required to locate the SSL security seal within the browser, verify the subject information (company name, server name etc), validate the trust hierarchy and ensure that the certificate is valid (i.e. within the validity period, and it hasn't been revoked). Although 86% of South African Internets banking users have post-matric qualifications [1], they have not been educated to properly authenticate a web server.

Phishing attacks use both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials [13] and such attacks rely on the assumptions that user would not be able to differentiate between the real site and a spoofed site.

From a technical perspective, and assuming the user is able to verify the SSL digital certificate of the web server, there is still the risk that someone might be able to extract or copy the associated private key from the server. Either an internal IT administrator within the bank can export the certificate, or a hacker can compromise the web-server and run a key finding attack [2]. It is considered best practice to protect private keys using a Hardware Security Module [9] as this would make in infeasible to try to extract the private keys without authorization.

5.3 Confidentiality of Password and Sensitive Information

5.3.1 Current Risks

Although the SSL version 3 protocol is believed to be secure [3], there are still various risks of exposing confidential sensitive data in cleartext. Data are protected during transit within the SSL session, but that is not to say that it is confidential before it enters the session, or after it leaves the session.

Client: Sensitive Data before entering the SSL Session

- *Keystroke logging:* In this case, there is a small application running in the background that captures all keystrokes, as the user enters his Password and account information. This application can then store, e-mail or replay this information at a later stage.
- *SSL Proxy*: Numerous SSL proxies are freely available on the Internet, and a hacker could easily install such a proxy on the user's system. The SSL session is established between the proxy application and the web-server, and optionally, another SSL session is setup between the proxy application and the browser. In this case the user will be prompted with a message that the certificate is not valid, but as discussed above, the user may not even know that this is a security risk.
- *Phishing*: As discussed, an attacker could also obtain the user's credentials through an out-of-band mechanism, such as requesting the user to enter it into a spoofed web-site or via e-mail.

Since the user's system is considered to be insecure and untrusted, it is possible that other forms of *spyware* and *malware* software may be installed on the system and could intercept information before it is encrypted in the SSL session.

Server: Sensitive Data after it exits the SSL Session

Once the SSL session is terminated on the Web Server, the data will be returned to the server application in the clear. This implies that any of the authorised administrators of that web server are

able to view the clear credentials. In addition, if a hacker is able to compromise the web server, he would have similar rights to view clear credentials.

Similar to the risk highlighted in the *Identification and Authentication* section, an attacker would also be able to find the private key(s) on the hard drive [2] and compromise the confidentiality of all SSL communication to the server.

5.3.2 Possible Solutions

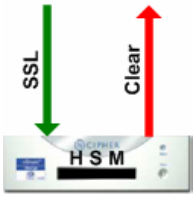
Client: Protecting Sensitive Data before entering the SSL Session

Since the confidentiality of the Password seems to carry the most risk, it has become clear that the password must be different every time, thus avoiding replay attacks [6]. The attacker would therefore find no value in capturing the current password, as it will be rendered useless after the first use.

Server: Protecting Sensitive Data after it exits the SSL Session

It has been proved that the private key of a SSL certificate can be extracted once access is gained to a server [2]. Furthermore, once the data exits the SSL Session, it is in the clear. Solutions to these problems include:

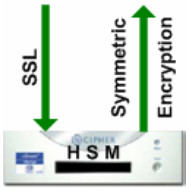
Hardware SSL



By protecting the SSL Digital Certificate with a Hardware Security Module, it is impossible to gain access to the private key without explicit authorisation. The HSM will also offload the asymmetric processing involved in the SSL handshake, thus acting as an SSL accelerator.

Note that in this case the data will still be in clear format once it exits the HSM.

Hardware SSL, and password re-encryption



The SSL Digital Certificate is protected by the HSM, as in the previous solution, but in this case the data is immediately re-encrypted with a symmetric key (usually shared with the mainframe). This SSL-to-Symmetric re-encryption is executed within the secure boundary of the HSM.

Neither attackers nor administrators of this server are therefore capable of viewing the clear password values.

6 CONCLUSION

We have seen that if we think of a PIN or Password as a cryptographic key, and therefore apply the principles of key management to this *key*, it highlights serious risks within the *Identification and Authentication* mechanisms employed by Internet Banking applications. Furthermore, even though the data is confidential between the client and the server, the same assertion is not true before or after the SSL connection. Although the banks are continually adding new security features to these websites, such as *randomised PIN Pads*, and notification methods, they have not yet addressed the real risk.

7 REFERENCES

[1] Buys, M, Brown, I. *Customer Satisfaction with Internet Banking Web Sites: An Empirical Test and Validation of a Measuring Instrument*. Proceedings of SAICSIT 2004, Pages 44 – 52

[2] Shamir, A., Van Someren, N., *Playing hide and seek with stored keys.* Financial Cryptography '99, LNCS 1648, Springer-Verlag, 1999, pp. 118–124

[3] Wagner, D., Schneier, B. *Analysis of the SSL 3.0 protocol.* The Second USENIX Workshop on Electronic Commerce Proceedings, USENIX Press, November 1996, pp. 29-40

[4] Menezes, A., Van Oorschot, P., and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996. (Chapters 13 & 15)

[5] KPMG Risk and Advisory Services, *Key Management Policy and Practice Framework*, KPMG LLP, January 2002

[6] Ferguson, N., Schneier, B., *Practical Cryptography*, Wiley Publishing, 2003

[7] MasterCard International, *Payment Card Industry PIN Security Requirements*, MasterCard International Incorporated, July 2004

[8] VISA International, *Payment Card Industry PIN Security Requirements*, VISA International, July 2004

[9] National Institute of Standards and Technology, *FIPS 140-2: Security Requirements for Cryptographic Modules,* Federal Information Processing Standards Publication, May 2001

[10] American National Standards Institute, ANSI X9.17: Financial Institution Key Management (Wholesale) standard

[11] Freier, A.O., Karlton, P., Kockhe, P.C., *The SSL Protocol, Version 3.0. Internet Draft*, Transport Layer Security Working Group, November 1996 (http://home.netscape.com/eng/ssl3)

[12] Granova, A., Eloff, J.H.P., South African Online Banking: Who carries the risk?, Proceedings of the ISSA 2004 Enabling Tomorrow Conference, July 2004

[13] Anti-Phishing Work Group (APWG), [online] http://www.antiphishing.org/

Vendor References:

ActivCard	www.activcard.com
RSA	www.rsasecurity.com
OATH	www.openauthentication.com
nCipher	www.ncipher.com
NamITech	www.namitech.com