

# *Ziliant* *Systems*

## **Legally-Binding Digital Signatures**

---

and hardware authentication devices

<b>Document no:</b>	WHI-0004-110
<b>Version:</b>	1.1
<b>File name:</b>	Legally-binding digital signatures V1_1
<b>Date:</b>	26 March 2004

## Table of contents

1	Scope .....	3
2	Introduction.....	3
3	Electronic Signature .....	4
3.1	Asymmetric cryptography .....	4
3.2	Digital certificate .....	4
3.3	Data integrity .....	5
3.4	Signing control .....	5
4	Authentication device .....	6
4.1	Digital signature algorithm.....	6
4.2	Device user authentication .....	6
4.3	Manual control.....	6
4.4	User intention .....	7
5	Key Management .....	8
5.1	Key generation .....	8
5.2	Key storage .....	8
5.3	Key distribution.....	8
6	Security .....	9
7	References .....	11

Note: This document is protected by a copyright. No part of this document may be reproduced or used in any form or by any means - graphic, electronic or mechanical, including photocopying, recording, taping or information storage and retrieval systems without the express written permission of Ziliant Systems.

# 1 Scope

This paper describes how a hardware security device can be used to meet the requirements of legally binding digital signatures. It draws on the recommendations found in the South African Electronic Communications and Transactions (ECT) bill [1] and the German Signature Act (SigG[2]) and Ordinance (SigV[3]). It does not describe or specify any public key infrastructure (PKI) or any digital certificate practices. The described device can be used in a security system with or without the use of certificates.

## 2 Introduction

The credibility of an electronic or digital signature is only as strong as the signature method used and the environment in which it is generated. An electronic signature is also only useful if it is possible to prove the signer's identity and intention. It is virtually impossible to provide the above using a software application on a PC. Much has been written about the security weaknesses in operating systems particularly when connected to a network. Operating systems like Microsoft Windows are the most complex systems ever created and it is this very complexity that makes them vulnerable. The best we can do for now is to try to bypass our reliance on operating systems and networks and this can only really be done using hardware security devices that operate independently. Since most information systems are designed with security as an afterthought, adding security is often regarded as an impediment or "grudge buy". In the new connected world however, the image of security products has at least improved to the level of "a necessary evil" and hopefully in time may even reach the lofty heights of "an enabler"! The introduction of legally recognised digital signatures has added a new label however – "mandatory". In order for digital signatures to be recognised in a court of law, they must have been generated and verified in a manner that is dictated by the country's digital signature law (if it has one). South Africa's ECT bill [1] makes provision for the legal recognition of a digital signature and the accreditation of an authentication device capable of generating the signature.

## 3 Electronic Signature

The South African ECT act [1] defines an “advanced electronic signature” as “an electronic signature which results from a process which has been accredited by the Authority as provided for in section 38” of the act. The “Authority” in this case is the Director General of the Department of Communications. Section 38 of the act merely states that the authentication products and services must be accredited by the Authority and that failure to do so is an offence. The definition of an electronic or digital signature according to the ECT act may be found in section 39 (1).

“39. (1) The Authority may not accredit authentication products or services unless the Authority is satisfied that an electronic signature to which such authentication products or services relate –

- (a) is uniquely linked to the user;
- (b) is capable of identifying that user;
- (c) is created using means that can be maintained under the sole control of that user;
- and
- (d) will be linked to the data message to which it relates in such a manner that any subsequent change of the data or data message is detectable.

### 3.1 Asymmetric cryptography

ECT Section 39 (1)(a) immediately implies the use of asymmetric cryptography such as public key algorithms. Therefore the traditional banking Message Authentication Code (MAC) cannot be used since it uses a symmetric algorithm (DES) meaning that the bank uses the same key to verify and can also therefore produce the same signature. This section of the act also implies that the signature key must be generated within an environment that prevents disclosure or else it cannot be uniquely linked to the user (see 5.1 [Key generation](#)).

### 3.2 Digital certificate

ECT Section 39 (1)(b) implies the use of a digital certificate since the signature verifier must be able to prove that the signing key belongs to the legitimate user. It should be noted however that there are two basic models for verifying the legitimacy of a user’s public key – Certificate-based and Account-based verification. The most common is the Certificate-based model (traditional PKI approach) that uses a trusted third party to certify

user public keys by creating a digital certificate. The most common standard applied here is the X.509 standard [4]. This standard defines data fields in the certificate for user identification. A digital certificate combines the user's signature verification key with the user's Identity and is signed by an accredited Certification Authority (CA) that meets the criteria specified in ECT Section 39 (4). The alternative Account-based model does not require a trusted third party. The user's certificate is stored by service providers in a database and linked to the user's identity number or account number. The Account-based model is well suited to commercial applications where legal liability around the use of digital signatures is established using contracts. In these cases trusted third parties create commercial complications. The X9.59 [5] standard was motivated around this model. The ECT act already assumes the use of CA's, however since these CA's will be legally certified, signature liability can be established in law directly without the need for contracts therefore some of the motivations for using the Account-based model fall away. The employment of digital certificates or a public key database requires that the users public key is readily accessible (see 5.3 [Key distribution](#)).

### **3.3 Data integrity**

ECT Section 39 (1)(d) relates to the integrity of the signature i.e. that the signed data cannot be modified without detection. This implies the use of a message digest algorithm or a one-way "hash" function such as the SHA-1 algorithm [9]. The digest/hash must be calculated over the information to be signed and must be bound to the signature. In this case it is the hash that is actually signed by the signing algorithm.

### **3.4 Signing control**

ECT Section (1) implies that users must be in full control of what they are signing. This is one of the most important aspects of a digital signature and one that is the most difficult to fulfil. Electronic information is not directly tangible to humans and can therefore easily be copied, substituted or modified in a way that is not readily detectable by humans. The best that we can do right now is to provide users with a tamper-resistant device that contains the user's signing key. The device must only use the signing key when authorised to do so by the legitimate user. This can either be achieved by the use of biometric recognition of the user, and/or by the correct submission of a PIN or password that only the user knows.

## **4 Authentication device**

The requirements for an ECT compliant device must satisfy the electronic signature requirements as described above. It must comply with section 38 of act and the digital signature must comply with section 39 (1) (a, b and d) of the act. These provisions can be met through the use of hardware in the form of a security device that contains the user's signing key and signing function.

### **4.1 Digital signature algorithm**

The signature requirements can be met using the algorithm specified by the American FIPS 186 standard [8] as this is a royalty-free public standard. However this algorithm is limited to a 1024 bit key length. The RSA algorithm and the method specified in the RSA Inc. public key standard PKCS #1 can also meet the requirements and it can be used with any key length. The RSA algorithm's patent has expired and can now be used royalty-free.

### **4.2 Device user authentication**

The signing function, and hence the use of the user's signing key may only be initiated by the legitimate owner of the signing key. Therefore the security device must be capable of authenticating the legitimate user. As mentioned above, a biometric sensing mechanism such as a fingerprint scanner can be used but in practice there are problems associated with these mechanisms [1] and they are often prohibitively expensive for mass deployment. The most practical approach today is still the use of a PIN or password that is only known by the user. The PIN must however be submitted to the security device by the most direct means i.e. it should not be feasible to subversively read/record the PIN en-route. This is a problem for smart cards used with standard smart card readers. In this case the PIN is entered on the PC and submitted to the card via the reader's communications interfaces and can be intercepted on the PC by a software keystroke logger or a hardware cable/connector keystroke logger. Therefore the signing device must have its own integrated keyboard to allow direct entry.

### **4.3 Manual control**

The user must have control of the signing function or the use of the signing key. A smart card together with a conventional smart card reader cannot achieve this. Since the signature request originates from the PC, it is possible for subversive software to

continuously request signatures for any data while the card is present. Even though the request may require the submission of a PIN, the PIN can be recorded and repeatedly submitted. Therefore as required in section 4.2, the PIN must be entered directly to the signing device via an integrated keyboard. In addition, each signature request must result in a user prompt and user action to initiate the signing function. This implies the use of an integrated display.

#### **4.4 User intention**

This is probably the most difficult criterion to meet. The user must not only be in control of the signing function but must also be sure of what is being signed i.e. the device/mechanism should prevent the information to be signed from being substituted in an undetectable way. Ideally the security device itself should be capable of displaying the information to be signed in its original form (SigV[3] § 16 (3)). This is only cost-effective for small messages or numeric information such as account numbers etc. In the case of financial transactions, this is quite easily achieved as the transaction details are usually numeric (i.e. amounts and account numbers) and can be entered directly on the integrated keyboard. In the case of document signing, there are many cases where a standard contract must be signed and the only user input required is the user's identity and signature. In these cases a unique document identity code could be entered on the integrated keyboard signifying the users intent. Any other personal details required could be automatically included from pre-stored data in the device. The document identifier together with the document itself could have been previously signed by the service provider to prevent modification. Ideally electronic documents should be designed in this way. However, there are cases where the document to be signed requires such a large amount of user input as to make it impractical to enter on a small integrated keyboard. A practical approach here is to allow entry on the PC keyboard, but at least display the hash code of the information to be signed on the device's integrated display and require that the user acknowledge this by manually initiating the signing function either by pressing a button or by entering a PIN on the device. It is at least possible to detect if the hash code was substituted before signing, since the intended document will now not have a valid signature associated with it and this can be checked at any time.

# 5 Key Management

## 5.1 Key generation

In order to guarantee that the signing key is uniquely associated with the legitimate user, it must be generated within a secure environment that prevents its disclosure. Ideally it should be generated on the user's assigned device and be initiated under the user's control (SigV[3] § 16 (1) & (2)). The authentication device should contain a highly secure environment for generating and storing the signature keys i.e. a secure tamper-resistant module (see 6 [Security](#)).

## 5.2 Key storage

During the entire signature key lifecycle, the key should be stored in a secure environment i.e. within the security module. According to the German signature law, this key's entire lifecycle must be spent within these confines i.e. from generation through to destruction. This is practical for user devices such as smart cards, but not always for server-side security devices as was painfully discovered during early attempts by vendors to comply with the German signature act. Most high-end cryptographic security devices did not comply and some of these devices represented significant investments such as the built-in security device in the IBM OS/390 mainframes. These high-end security devices are designed to handle many keys since they are mainly used on the server-side of a network. These keys are typically stored outside of the device in encrypted form. The only secret key stored in the device is the one used to encrypt the external keys. This minimises the cost of internal memory and simplifies the tamper-response mechanism used to erase sensitive information within the device. This model also has the important benefit of allowing keys to be backed up. However, this model does not comply with the German signature law, as the signature key is stored outside of the security device albeit in encrypted form. High-end devices therefore have to be designed to store and protect one or more signature keys for the entire key lifecycle.

## 5.3 Key distribution

The public key is generated together with the signing key on the authentication device and can be stored there and read out on demand. The private key can never be read out. During registration for a particular service the user initiates the transfer of the public key via a menu option on the device. If necessary a digital certificate can be generated by the

service at this stage and stored in a directory. The device should also have a public key fingerprint function allowing the user to verify that a particular service has the correct key associated with the user's identity.

## 6 Security

Electronic signatures are generated by proxy, i.e. they rely on a separate electronic device to perform the actual signature unlike a handwritten signature which is performed directly by the signer. Therefore the security around the use of the device and in fact throughout its complete lifecycle is paramount. The privacy of the signing key and its activation mechanism are the most important factors. The signing key itself should spend its full lifecycle i.e. from generation to destruction within the same secure confines. The security of the confining environment (i.e. security module) should ideally be formally verified according to FIPS-140-1 level 4 [7] and the design should be in accordance with the Common Criteria methodology. It should also ideally be formally validated to CC EAL-7 [12][13][14][15]. It is useful to use both the FIPS and CC standards as they cover different ground. CC is much like the old ITSEC standards whereby a security target is specified and the design of the system is validated using the security target objectives. This does not necessarily lead to a secure device. The FIPS 140-1 program is more oriented towards security modules and mandates the inclusion of certain characteristics. A combination of both approaches would be wise. However, achieving the above validations and subsequent certifications is an extremely onerous and expensive undertaking. A device vendor would only embark on this route if the revenue from the sales of the device justified it. This level of validation may be necessary though for a Certification Authority (CA) signing device since a security compromise of this device may affect a large number of people. In the financial world a typical user security device would at least conform to the ISO 1395-1 recommendations [11]. In this environment, the maximum damage resulting from a compromise can be measured in monetary terms and is usually limited to personal banking daily limits. However, if a signature device can be used say for signing a will, a multi-million dollar contract or even a declaration of war between two countries, the maximum damage could be extremely large. Fortunately though, in a court of law judgments are based on a balance of evidence and in any particular dispute the signature, electronic or manual only forms part of this evidence. Given that a personal signing device is not something in the public domain such as an ATM or credit card terminal and is kept

securely by the user, the exposure to potential compromise is already limited. Device vendors can however use an accredited smart card to store the signature keys. There are a few smart cards that have already been certified in accordance with the Common Criteria (CC) to at least EAL-4 level. These are highly secure devices that provide protection against side-channel attacks such as DPA and SPA attacks [16] as well as DFA attacks [17]. In addition they are designed to resist penetration attacks aimed at accessing the signature keys.

## 7 References

- [1] Republic of South Africa, Electronic Communications and Transactions Bill 2002.
- [2] Gesetz zur digitalen Signatur (Signaturgesetz - SigG)<sup>1</sup>, Artikel 3 in „Gesetz zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste (Informations- und Kommunikationsdienste-Gesetz - IuKDG)“; 22. Juli 1997, Bundesgesetzblatt vom 28. Juli 1997
- [3] Verordnung zur digitalen Signatur (Signaturverordnung - SigV)<sup>2</sup> in der Fassung des Beschlusses der Bundesregierung vom 8. Oktober 1997
- [4] ITU-T Recommendation X.509, Information Technology, Open systems Interconnection, The directory authentication framework, 03/00, Approved 2000-03
- [5] DSTU X9.59 – Electronic Commerce for the Financial Services Industry: Account-Based Secure Payment Objects (Draft Standard)
- [6] The RSA Encryption Standard PKCS #1 – RSA Data Security Inc.
- [7] FIPS PUB 140-1 Security Requirements for Cryptographic Modules. National Institute of Standards and Technology (NIST) publication 1994 January 1
- [8] FIPS 186, “Digital signature standard” Federal Information Processing Standards Publication 186, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.
- [9] FIPS 180, “Secure hash standard” Federal Information Processing Standards Publication 180, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, May 11 1993.
- [10] ISO 9564-1:2002, Second Edition, Banking – Personal Identification Number management and security, Part 1: PIN protection principles and techniques
- [11] ISO 13491-1, First Edition 1998-06-15, Banking – Secure cryptographic devices (retail), Part 1: Concepts, requirements and evaluation methods

---

<sup>1</sup> German Digital Signature Act

<sup>2</sup> German Digital Signature Ordinance

- [12] Common Criteria for Information Technology Security Evaluation, Part 1 – Introduction and general model, August 1999, Version 2.1, CCIMB-99-031.
- [13] Common Criteria for Information Technology Security Evaluation, Part 2 – Security functional requirements, August 1999, Version 2.1, CCIMB-99-032.
- [14] Common Criteria for Information Technology Security Evaluation, Part 3 – Security assurance requirements, August 1999, Version 2.1, CC-IMB-99-033.
- [15] Common Criteria for Information Technology Security Evaluation, EMV Integrated Circuit Card Credit & Debit Application Protection Profile (EMV-App PP) Draft, Draft Version 0.4, 14 December 2001
- [16] Differential Power Analysis (DPA) by Paul Kocher, Joshua Jaffe and Benjamin Jun. Paper available in PDF format from [www.cryptography.com](http://www.cryptography.com)
- [17] Differential Fault Analysis (DFA) of Secret Key Cryptosystems. – Eli Biham & Adi Shamir – Technion Israel Institute of Technology, Computer Science Department Technical Report – CS0910-1997 ([www.cs.technion.ac.il/~biham/](http://www.cs.technion.ac.il/~biham/))
- [18] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino, "Impact of Artificial Gummy Fingers on Fingerprint Systems," Proceedings of SPIE Vol. #4677, Optical Security and Counterfeit Deterrence Techniques IV, 2002.