

Ziliant *Systems*

ZIPAD600 Digital Signature Generation

Non-repudiation and proof of intent through
unidirectional communications

Document no:	WHI-0005-100
Version:	1.0
File name:	Unidirectional signatures.doc
Date:	26 March 2004

Table of contents

1	Introduction.....	3
2	Legal digital signatures.....	3
3	The signature process.....	4
4	ZiPAD600 signatures.....	5
4.1	“Push-only” signature attributes	5
4.1.1	Input data capture	5
4.1.2	Unidirectional communications.....	5
4.1.3	Personal information storage	5
4.1.4	User authentication	6
4.1.5	Programmable applications.....	6
5	“Push-only” signatures.....	6
5.1	Financial transactions.....	6
5.2	Contracts and legal documents.....	6
6	“Push-pull” signatures.....	7

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 Zilant Systems.

1 Introduction

Many countries now accept digital signatures as a legal alternative to hand signatures provided that they have been generated using legally accepted methods. One of the most important factors in the generation of a digital signature is there should be evidence of the signatory's intent. The signature must be generated under the signatory's control and the signatory must be aware of exactly what is being signed. This is difficult to prove when the signature is being generated on a computer and especially if the computer is connected to a network. Even if a smart card or token is used to generate the signature, the PIN or password used to enable the signature function is typically submitted via the computer keyboard. In this case it is possible for malicious software to record the PIN/password, use it to access the device and generate any number of fraudulent digital signatures. It should not be possible for a computer to generate a signature without the user's knowledge nor should it be possible for the computer to substitute or modify the user's data before it is signed. This paper describes how the ZIPAD600 is designed to prevent data from being manipulated before signing and therefore support true non-repudiation and proof of signatory intent.

2 Legal digital signatures

Digital signatures, no matter how securely they are created, cannot always be used in a legal sense. The most common use of digital signatures today is to verify a web server's digital certificate when connecting to a secure site on the Internet. This is a verification process carried out by the client's browser using an installed Certification Authority (CA) public key. Once the server's public key has been verified, it can be used by the client browser's Secure Sockets Layer (SSL) to secure communications with the server. The most common SSL configuration used is "server authentication". In this mode only the server is required to perform a digital signature and the client machine only performs signature verification. A more advanced mode of operation commonly referred to as "client authentication" allows both client and server to be authenticated (should really be called "mutual authentication"). In this mode, the client is also required to perform a digital signature. The use of digital signatures in SSL/TLS communications however is only useful for simplifying key management between unknown entities and in most cases

cannot be used in a legal sense. Firstly, the digital signatures are usually generated in software on a computer and although some high-throughput servers use cryptographic hardware accelerators, these devices are generally only used to unburden the server and are not generally very secure. Secondly, the data being secured is transient (communications) and is not recorded. Thirdly, even if the communicated transaction data was recorded, the data that is actually signed is in fact cryptographic keying material, not the transaction data. Therefore SSL/TLS signatures will not satisfy most signature laws. In order to prove that a transaction took place in a court of law, the transaction data itself must be digitally signed and the signature itself must be generated in accordance with the digital signature legislation.

3 The signature process

The traditional method of digitally signing electronic information is to firstly create a digest or hash code of the information by using an electronic digest algorithm such as SHA-1 or MD5. The reason for this step is that the signature algorithm can only operate on a hundred or so characters at a time and it takes too long to perform multiple signatures over a large data file due to the computational effort required. The resultant hash code is unique to the data over which it was generated and can therefore be used to represent the data file in the signature. When a stand-alone hardware device is used for the signature generation, it is typically connected to the computer through some serial communications port. Data files on the computer such as spreadsheets or word processor files can get rather large and could take quite some time to be downloaded to the signature device. Therefore it is often convenient to have the computer perform the digest operation and only send the digest code to the device for signing. The resultant signature is then transmitted back to the host computer. This “push-pull” approach is problematic in that the data can be modified by the computer without the signer’s knowledge since there is no direct binding to what the user sees on the computer screen and what is submitted for signing on the user’s device. It is assumed that the secure signature device owned by the user and operated only by the user is trustworthy and the computer, which is typically an open system with several data input ports including a network connection, is not trustworthy. The bi-directional communication of the information (push-pull) during the signature process means that only one half of the process is controlled by the user i.e. the signing itself. The ideal situation is that the information being signed is viewed and/or

entered on the same secure device that signs it before communicating the signature to the computer/network i.e. the process follows a unidirectional path. In this paper, these two methods of signing will be referred to as “push-pull” signing and “push-only” signing respectively.

4 ZiPAD600 signatures

The ZiPAD600 is a compact standalone terminal that connects to a computer and is used for the purpose of generating electronic signatures. The ZiPAD600 can be used to sign any electronic information residing on a computer. It can be used in the traditional “push-pull” way described above or it can be used in a more secure “push-only” method where data is captured and signed directly on the device before being sent to the host computer.

4.1 “Push-only” signature attributes

4.1.1 Input data capture

In order to force the signature process to follow a unidirectional path, the signing device must provide the ability to enter and view data. The ZiPAD600 provides a 2 x 16 character LCD display and a keypad with 16 keys. This allows data to be entered, viewed and signed on the ZiPAD600 directly. Although mainly suited to financial transactions, and other applications requiring minimal user input, this covers quite a wide variety of applications (see following sections).

4.1.2 Unidirectional communications

The ZiPAD600 provides a standard keyboard interface to the computer (PS/2 or USB) thereby enhancing the unidirectional nature of the signature process i.e. the data cannot be manipulated since there is no back channel.

4.1.3 Personal information storage

The ZiPAD600 provides 64Kbytes of data storage space that can be used to store the user’s personal information such as name, address, ID number, etc. The user need only enter this personal information once. Thereafter it can be automatically used in a signature or transmitted to the computer. In financial applications, the user can store frequently used account numbers in the same way. Since there is way more than enough storage space

available, the additional space can be used by a ZiPAD600 application to store an audit trail of all transactions completed (even entire signatures).

4.1.4 User authentication

The ZiPAD600 requires the user to enter a PIN before any device function can be performed.

4.1.5 Programmable applications

The ZiPAD600 supports customisable applications. The device has 64Kbytes of Flash EEPROM memory for application storage. This makes it easy for applications to be tailored for “push-only” signatures since the user screens and prompts can be customised.

5 “Push-only” signatures

5.1 Financial transactions

Most financial transactions only require the user to enter an amount and one or two account numbers. Since the user’s account numbers can be stored on the device, an account number only needs to be entered during a transaction in a once-off beneficiary payment. In most cases, the user will only need to enter an amount and use a menu to select the account number. Therefore the entire transaction can be completed on the ZiPAD600.

5.2 Contracts and legal documents

Most contracts and other legal documents are standard forms and can often be referenced by a unique identifier (i.e. “a KK354 form”). Therefore it is possible to merely include the form’s unique identifier in the signature and not necessarily the entire contents of the form. This assumes that the form is kept under strict configuration control and cannot be modified at whim. Of course the situation can be vastly improved if the electronic document is designed to be used in this way i.e. the document’s unique reference is numeric-only and the standard form is itself digitally pre-signed to prevent any modification. The signatory merely enter the form’s unique identifier on the ZiPAD600 during a signature operation. Any personal information required by the form (pre-stored in the device) can be automatically transmitted as keystrokes (keyboard interface) for inclusion into the form as if the user had typed them in on the computer keyboard. The

same pre-stored personal data is automatically included in the signature on the ZiPAD600. In this way an electronic document can be signed using the “push-only” method completely under the users control.

6 “Push-pull” signatures

Of course there are applications where the method described in 5.2 above is not practical. In cases where the entire contents of a document is unique and requires signing, there is currently no practical alternative but to use the “push-pull” method. ZiPAD600 also supports this method but adds an additional security feature to improve the situation. When a document hash code is presented to the device for signing by the host computer, the user is informed of this event and is required to take action before the data is signed. The ZiPAD600 displays the hash code on the LCD and requires that the user press the “Auth” key before a signature is generated. Although the hash code is in most cases meaningless to a user, the important point is that the computer cannot autonomously force the ZiPAD600 to generate a signature. This method is fairly secure in that a fraudulent signature attempt can be detected. If some malicious software on the computer tries to substitute the legitimate hash code for another one, the user can detect this afterwards since the intended information will now not have a legitimate signature and this can be verified at any time by the user or any other third party.