Robert Müller

Fingerprint Verification with Microprocessor Security Tokens

Herbert Utz Verlag · Wissenschaft
München

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist
bei Der Deutschen Bibliothek erhältlich

Zugleich: Dissertation, München, Techn. Univ., 2001

ISBN 3-8316-0015-5

Printed in Germany

Herbert Utz Verlag GmbH, München
Tel.: 089/277791-00 – Fax: 089/277791-01

TABLE OF CONTENTS

1 Introduction and Overview

With the advent of e-commerce and the tremendous increase in communication, reliable methods for user authentication are required more than ever before. Today's information society needs ways to legally sign documents electronically and to verify their signatures again. In current security and payment systems, the knowledge of a secret or the possession of a token is the key to access. In recent years, a new method of user authentication has gained significant public interest: *biometrics*. It is the science of identifying individuals via their physical or behavioral characteristics. This technology allows user authentication without remembering passwords and PIN codes. In most civil applications it is neither desirable nor acceptable to collect a database of all legitimate users. Therefore, biometrics will not make security tokens redundant but more secure. The obvious advantage is that a user will be legally tied to his signature generating device. Biometrics can be used to enhance the overall system security and also the ease of use. In conjunction with smart card technology, it is possible to implement high security and privacy for open applications.

Digitally signed documents and transactions are a main application field, that requires reliable user authentication. The current legal proposals for digital signatures require knowledge based user verification to be performed in the signature generating device, which must be tamper proof. Therefore, it is most appreciated to perform biometric user authentication in security tokens like smart cards. The token industry is very price sensitive and uses microcontrollers with limited space and processing power. Several concepts on how to perform a fingerprint verification in a smart card are developed in this thesis and compared against each other.

A main objective is to make this thesis understandable without special skills while not overloading it with too many known facts. Only a basic understanding of computer science and mathematics should be necessary to work through this thesis. The introductory chapters may be read independently from each other and flipped over according to the reader's previous knowledge.

The thesis is organized into three main parts:

The first part covers chapters 2 through 4 and gives introductions and required background to technologies. The background information given in these chapters is necessary to understand the thesis. A brief introduction into the science of cryptography is summarized in chapter 2. An overview on smart card technology and biometrics is respectively given in chapters 3 and 4.

The second part focuses on fingerprint recognition and represents current state-of-the-art descriptions. Chapter 5 deals with fingerprint analysis. It gives the historical background, introduces the technical terms and points to some selected items of the previous work in fingerprinting. Image processing techniques for fingerprints and feature extraction are addressed in chapter 6. They are mandatory presuppositions for the development and testing of new algorithms.

The new ideas and algorithms developed within this thesis are introduced in the third part of this thesis. A flexible fingerprint verification system based on minutia, ridge counts and pores is designed and implemented in chapter 7. Side constraints are to achieve reliable matching accuracy while keeping the template size small to allow storage in an embedded token. Chapter 8 outlines the principles, ideas and original concepts on how to perform a positive identification in a microcontroller with limited memory and computational power. This section contains the main contributions of the thesis. The algorithms developed in this thesis were tested with various databases and sensor sources. Experimental results achieved in this work are summarized in chapter 9. The conclusions drawn from the implementation and experimental results are summarized in chapter 10. The thesis ends with open research aspects that arose during the work and were not considered in detail.

2 Cryptography

To get an idea of the advantage of performing biometric verification methods within security tokens, it is mandatory to understand the basic principles of cryptography and digital signatures. This chapter gives a brief introduction to cryptography as needed throughout this thesis. For a detailed study of the science of cryptography, the following material is suggested reading: [Schn1996, Baue1995, Beut1993, Diff1976, Kahn1967, Kobl1988, Rive1978].

2.1 Historical Background

Cryptography stems from the Greek words “Kryptos” (= hidden) and “Graphein” (= write). The science of cryptography is as old as written communication between individuals. First trials to dissimulate messages are even known from ancient tribes, who swaddled papyrus rolls round a bamboo cane and then wrote down their message to be delivered by a messenger. Reading the symbols was only possible for the legitimate recipient holding a cane with the same diameter. The historic Julius Cesar also used an own cryptograph by shifting any character of a message by three positions in alphabetical order[1]. After these basic approaches to hide the content of a message from spies, it was primarily the work of intelligence services during the world wars and even in the era of the cold war, that pushed ahead the development of cryptography and *cryptanalysis*[2]. More sophisticated character- and word-based cipher methods are listed in [Baue1995] and [Kahn1967]. Of course, the algorithms that were designed for manual encryption on paper or by means of mechanical machines, can all be broken in milliseconds with current computers. Since the dawn of digital technology, sophisticated cipher methods have been developed, that require extensive computational effort to be broken. Modern cryptography is a pure application of mathematics, in particular of number theory and complexity theory [Kobl1988, Schn1996, Kran1986, Ries1985, Papa1994].

The competing requirements of efficiency and security represent a general dilemma of the information society. On the one hand, individuals want to communicate in real-time and share their points of view and information globally. The tremendous need for communication calls for open, flexible and standardized systems. On the other hand, everybody’s right of privacy and intellectual property must be respected. Complex cryptographic algorithms are required to ensure, that private communication cannot be illegally accessed or intercepted even with main frame computers.

In our days, a large portion of the knowledge on cryptography is accessible to the public. With the increase in email-traffic, mobile phones and e-commerce, also civil applications

[1] This much reported on historic cipher method is known as Cesar-cipher today.

[2] The science of keeping messages secure and authentic is called cryptography. The art and practice of breaking ciphers is referred to as cryptanalysis. *Cryptology* is the branch of number theory and computational complexity, that encompasses both.

need concealed or, even more importantly, authentic communication. Digital communication without cryptographic protection is equivalent to sending letters without a sealed envelope. Cryptography is a suitable technology to protect the privacy of individual communication. The products associated with cryptography are no longer the mainframe computers and high speed networks from intelligence services. Today it is the tiny and inexpensive embedded system or cryptographic token, that carries out the operations to protect our communication and transactions – the modern medium *smart card* (see section 3) being in the vanguard.

2.2 Symmetric Cryptography

Let us consider a communication line with a source, that generates messages and a sink to receive them. Since others could listen to what is sent, the author applies some algorithm to his messages with a key as parameter. This operation is called *encryption* and produces the *ciphertext*. The recipient must apply an inverse operation, called *decryption*, with the same key to extract the original *plaintext* again. Sender and receiver must on one occasion have agreed on the cipher-method and the key. Methods utilizing this scheme are called *symmetric algorithms* because they use one and the same key for enciphering and deciphering messages.

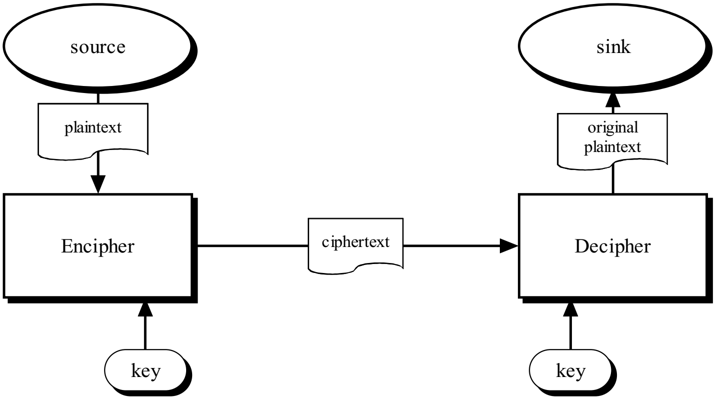

Figure 2.1: Communication scheme for symmetric encryption

If we denote the plaintext as *m* (for message), the ciphertext as *c*, the encipher operation as *E*, the decryption procedure as *D* and the key as *k*, the following conjectures are valid:

$c = E_k(m)$ The encipher function produces the ciphertext.

$m = D_k(c)$ The decipher function delivers the plaintext again.

$D_k(E_k(m)) = m$ This identity must hold true in order to recover the plaintext, when encrypted and decrypted.

The most popular symmetric cipher today is the *Data Encryption Standard* (DES), that uses a key with 56 information bits and operates in 16 rounds on a 64-bit plaintext. DES was developed in collaboration with the IBM laboratories and the U.S. National Security Agency in the early 1970s, published as a standard in 1977 [NBS1977] and adopted by ANSI and ISO later [ANSI1981, Schn1996]. The restricted key length should allow governments and institutions, which can afford enough computational power, to break the cipher by exhaustive search. Other important symmetric *block cipher algorithms*[3] and derivatives are IDEA, Triple-DES, SAFER, DES-X, FEAL, Blowfish and AES. Many more are listed and described in [Schn1996].

2.3 Public Key Cryptography

The concept of *public key cryptography* was invented independently by two teams of scientists. Their popular publications [Diff1976] and [Merk1978] mark an important milestone in the history of cryptography and gave this science a new direction. Keys come in pairs, where the first part is used to encipher and the second part to decipher. Therefore, public key algorithms sometimes are also referred to as *asymmetric ciphers*. Suppose an individual B possesses a pair of keys (k_P, k_S) that fits together. He discloses his encipher key k_P, the public key, to a specific sender A[4] who enciphers his message m to B:

$$c = E_{k_P}(m).$$

The ciphertext c is sent to B, who uses his secret key k_S to decipher it:

$$m' = D_{k_S}(c).$$

The cipher system was designed and an appropriate key pair generated so that the equation

$$m = m' = D_{k_S}(E_{k_P}(m)) \text{ holds.}$$

Many public key cryptosystems have been proposed since the pioneer work, but most of them were vulnerable to attacks or impractical to use. The most prominent public key algorithm is named RSA, after its inventors Ronald Rivest, Adi Shamir and Leonard Adleman [Rive1978]. It is by far the easiest public key cipher method to understand, implement and use and has withstood years of extensive cryptanalysis. RSA's security is based on the difficulty of factoring large integers. Here is how it works:

First of all, two random large prime numbers p and q are chosen and the product is computed:

$$n = pq.$$

[3] A block algorithm always divides the message to encipher in blocks of fixed length, while the so-called stream ciphers work on every character, byte or bit of a data stream [Baue1996].

[4] In practice, B would publish his public key to anybody like an address or a phone number.

The encryption key e is randomly chosen such that e is relatively prime to $(p\text{-}1)(q\text{-}1)$.

The extended Euclidian algorithm [Kobl1988] is used to compute d such that

$$ed \equiv 1 \bmod (p\text{-}1)(q\text{-}1).$$

This means, that d is the multiplicative inverse of e modulo $(p\text{-}1)(q\text{-}1)$.

Finally, the primes are normally discarded. The pair $\{e, n\}$ is the public key. d is the private key. To encipher, a message m_{plain} first has to be coded into a block stream of integer numbers m_i smaller than n. The encipher function E and the decipher function D are defined as follows:

$$E: \quad Z_n \rightarrow Z_n$$ [5]

$$m \rightarrow m^e \bmod n$$

$$D: \quad Z_n \rightarrow Z_n$$

$$c \rightarrow c^d \bmod n$$

A specific m_i is encrypted to

$$c_i = m_i^{\,e} \bmod n.$$

Applying the decipher operation to c_i yields

$$\begin{aligned} c_i^{\,d} \bmod n &= (m_i^{\,e} \bmod n)^{\,d} \bmod n \\ &= m_i^{\,ed} \bmod n \\ &= m_i^{\,k(p\text{-}1)\,(q\text{-}1)\,+1} \bmod n \\ &= m_i\,(m_i^{\,k(p\text{-}1)\,(q\text{-}1)} \bmod n\,) \bmod n \\ &= m_i \cdot 1 \bmod n \\ &= m_i \end{aligned}$$

That means, the message is recovered. The detailed number-theoretic conjectures to prove the above equations and identities are found in any of the following textbooks: [Kobl1988, Baue1994, Ries1985, Kran1986] and obviously in the original publication [Rive1978]. Good resources for multiple precision arithmetic as needed to implement RSA and other systems are [Saue1992, Knut1969, Wink4/1989]. The challenging task of key generation is addressed in [Damg1992, Maur1989, Maur1992, Müll1995].

[5] Z_n represents the integer numbers from 0 to n-1.