­­­­­­­**CHAPTER 1**

**1.1 INTRODUCTION**In most computer security contexts, user authentication is the fundamental building block and the first line of defense. User authentication is the basis of most types of access control and for user accountability. RFC 2828 defines user authentication as:

“The process of verifying an identity claimed by or for a system entity. An authentication process consists of two steps:

* **Identification step:** Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)
* **Verification step:** Presenting or generating authentication information that corroborates the binding between the entity and the identifier.”

In essence, identification is the means by which a user provides a claimed identity to the system; user authentication is the means of establishing the validity of the claim. Note that user authentication is distinct from message authentication.

**Means Of User Authentication**

There are four general means of authenticating a user's identity, which can be used alone or in combination:

• **Something the individual knows:** Examples includes a password, a personal identification number (PIN), or answers to a prearranged set of questions.

• **Something the individual possesses:** Examples include electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a *token*.

• **Something the individual is (static biometrics):** Examples include recognition by fingerprint, retina, and face.

• **Something the individual does (dynamic biometrics):** Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

All of these methods, properly implemented and used, can provide secure user authentication. However, each method has problems. An adversary may be able to guess or steal a password. Similarly, an adversary may be able to forge or steal a token. A user may forget a password or lose a token. Further, there is a significant administrative overhead for managing password and token information on systems and securing such information on systems. With respect to biometric authenticators, there are a variety of problems, including dealing with false positives and false negatives, user acceptance, cost, and convenience.

**1.2 LITERATURE SURVEY**

In 1981, Lamport proposed a remote user authentication scheme with password table [1]. Several schemes and improvements using the static identity (ID) are included. [2]-[4]Since the user’s login ID is static in these verifier-free schemes, it may leak partial information about the user’s login messages so that the adversary can use it to forge the user’s login messages by some subtle means.X. H. Yang *et al.*262 One of the solutions to the problem is to employ dynamic ID in different login.In 2004, Das et al. [5] proposed a dynamic *ID-*based remote user authentication scheme, which can resistreplay,masquerade, and insider attacks.

**1.3 OBJECTIVE**

**1.3.1 PASSWORD AUTHENTICATION**

The front line of defense against intruders is the password system. Virtually all multiuser systems require that a user provide not only a name or identifier (ID) but also a password. The system compares the password to a previously stored password for that user ID, maintained in a system password file. The password serves to authenticate the ID of the individual logging on to the system. In turn, the ID determines whether the user is authorized to gain access to a system, the privileges accorded to the user, and is used to determine discretionary access controls.

**1.3.2 SMART CARD**

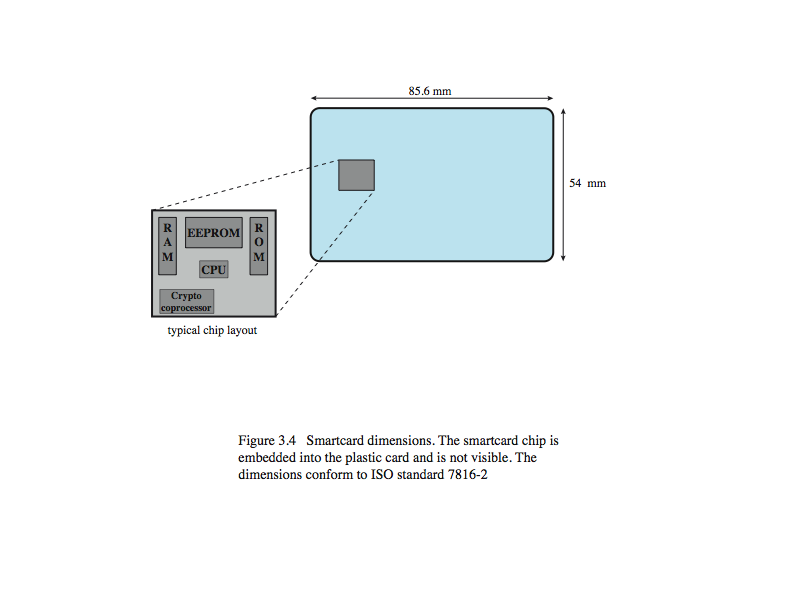


Figure 1.3.2.1

Smart Card

For user authentication to computer, the most important category of smart token is the smart card, which has the appearance of a credit card, has an electronic interface, and may use any of the possible authentication protocols. A smart card contains within it an entire microprocessor, including processor, memory, and I/O ports, as shown in figure.Some versions incorporate a special co-processing circuit for cryptographic operation to speed the task of encoding and decoding messages or generating digital signatures to validate the information transferred. In some cards, the I/O ports are directly accessible by a compatible reader by means of exposed electrical contacts. Other cards rely instead on an embedded antenna for wireless communication with the reader. A typical smart card includes three types of memory. Read-only memory (ROM) stores data that does not change during the card’s life, such as the card number and the cardholder’s name. Electrically erasable programmable ROM (EEPROM) holds application data and programs, such as the protocols that the card can execute. It also holds data that may vary with time. Random access memory (RAM) holds temporary data generated when applications are executed. Figure in the text illustrates a typical interaction between a smart card and a reader or computer system. An alternative to the smart card is a small, inexpensive flash memory device known as a USB dongle. It has the same functionality as a smart card, but connects to the existing USB port on a computer, hence it does not need a specific card reader.

**1.3.3 BIOMETRIC AUTHENTICATION**

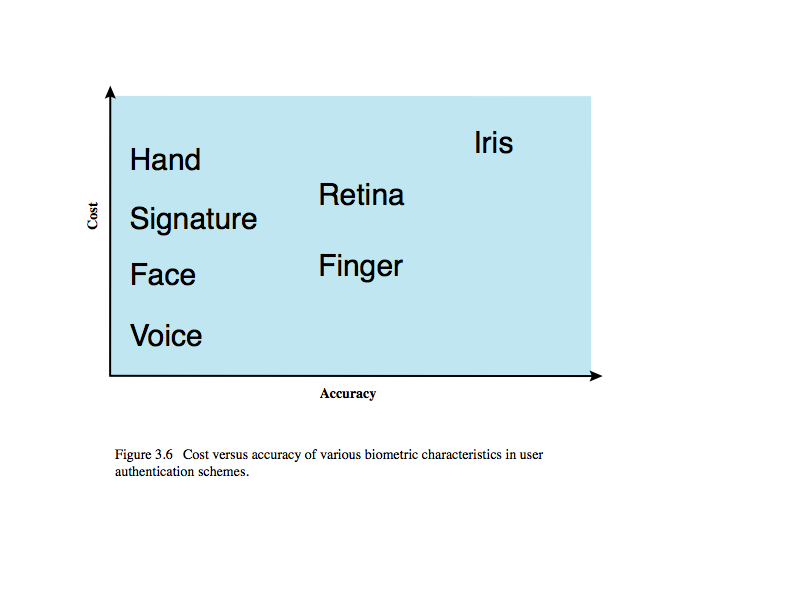


Figure 1.3.3.1

Biometric Authentication

A biometric authentication system attempts to authenticate an individual based on unique physical characteristics.These include static characteristics, such as fingerprints, hand geometry, facial characteristics, and retinal and iris patterns; and dynamic characteristics, such as voiceprint and signature. Compared to passwords and tokens, biometric authentication is both technically complex and expensive, and have yet to mature as a standard tool for user authentication to computer systems. Figure from the text gives a rough indication of the relative cost and accuracy of the most common biometric measures:

• Facial characteristics: define characteristics based on relative location and shape of key facial features, such as eyes, eyebrows, nose, lips, and chin shape.

• Fingerprints: the pattern of ridges and furrows on the surface of the fingertip, believed to be unique across the entire human population. Automated fingerprint systems extract a number of features to use as a surrogate for the full pattern.

• Hand geometry: identify features of hand,: e.g. shape, lengths & widths of fingers.

• Retinal pattern: formed by veins beneath the retinal surface is unique and therefore suitable for identification. Uses a digital image of the retinal pattern by projecting a low-intensity beam of visual or infrared light into the eye.

• Iris: Another unique physical characteristic is the detailed structure of the iris.

• Signature: each individual has a unique style of handwriting, esp in signature.

• Voice: patterns are more closely tied to physical and anatomical characteristics of the speaker, but still have a variation from sample to sample over time from the same speaker,complicating the biometric recognition task.

**1.3.4 REMOTE USER AUTHENTICATION**

The simplest form of user authentication is local authentication, in which a user attempts to access a system that is locally present, such as a stand-alone office PC or an ATM machine. The more complex case is that of remote user authentication, which takes place over the Internet, a network, or a communications link. Remote user authentication raises additional security threats, such as an eavesdropper being able to capture a password, or an adversary replaying an authentication sequence that has been observed. To counter threats to remote user authentication, systems generally rely on some form of challenge-response protocol. The user first transmits his or her identity to the remote host. The host generates a random number r, often called a nonce, and returns this nonce to the user, known as the challenge. The user’s response is the result of computing a function on the random number and a hash of the password: f(r,h(P)). The host stores the hash function of each register user’s password. When the response arrives, the host compares the incoming value to that it calculates. If the quantities match, the user is authenticated. This scheme defends against several forms of attack.The host stores only a hash code of the password, which secures the password from intruders into the host system. Also only a function of the password hash is sent, so it cannot be captured during transmission. Finally, the use of a random number as one of the arguments defends against a replay attack. The text includes details of variants using tokens, static biometrics and dynamic biometrics.

**1.3.5 AUTHENTICATION SECURITY ISSUES**

As with any security service,user authentication,particularly remote user authentication, is subject to a variety of attacks, as presented in Table 3.4 in the text. Client attacks are those in which an adversary attempts to achieve user authentication without access to the remote host or to the intervening communications path. The adversary attempts to masquerade as a legitimate user. e.g. in a password-based system, the adversary may attempt to guess the likely user password. Host attacks are directed at the user file at the host where passwords,token passcodes, or biometric templates are stored. Eavesdropping refers to an adversary’s attempt to learn the password by observing the user, finding a written copy of the password, keystroke logging, etc. Replay attacks involve an adversary repeating a previously captured user response. The most common countermeasure to such attacks is the challenge- response protocol. In a Trojan horse attack, an application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric. The adversary can then use the captured information to masquerade as a legitimate user. A denial-of-service attack attempts to disable a user authentication service by flooding the service with numerous authentication attempts.

**CHAPTER 2**

**REQUIREMENTS SPECIFICATION**

**2.1 Software Requirements:**

Language : Java (JDK1.7.0)

Operating System : Microsoft Windows Xp Service Pack 3

IDE : my eclipse IDE 8.6

Front End : JAVA (Swings)

**2.2 Hardware Requirements:**

Processor : Intel Pentium 4

**CHAPTER 3**

**DESIGN**

**BASIC CHALLENGE-RESPONSE PROTOCOLS FOR REMOTE USER AUTHENTICATION**

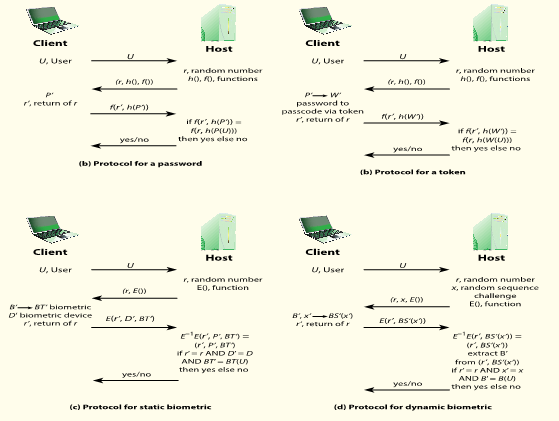


Figure 3.1

Basic Challenge-Response Protocols for remote user authentication

* Figure a provides a simple example of a challenge-response protocol for authentication via password. Actual protocols are more complex. In this example, a user first transmits his or her identity to the remote host. The host generates a random number r, often called a **nonce,** and returns this nonce to the user. In addition, the host specifies two functions, h() and f(), to be used in the response. This transmission from host to user is the challenge.The user’s response is the quantity f(r’, h(P’)), where r’ = r and P’ is the user’s password. The function h is a hash function, so that the response consists of the hash function of the user’s password combined with the random number using the function f.The host stores the hash function of each register user’s password, depicted as h(P(U)) for user U. When the response arrives, the host compares the incoming f(r’, h(P’)) to the calculated f(r, h(P(U))). If the quantities match, the user is authenticated.This scheme defends against several forms of attack. The host stores not the password but a hash code of the password.This secures the password from intruders into the host system. In addition, not even the hash of the password is transmitted directly, but rather a function in which the password hash is one of the arguments. Thus, for a suitable function f, the password hash cannot be captured during transmission. Finally, the use of a random number as one of the arguments of defends against a replay attack, in which an adversary captures the user’s transmission and attempts to log on to a system by retransmitting the user’s messages.
* Figure b provides a simple example of a token protocol for authentication.As before, a user first transmits his or her identity to the remote host. The host returns a random number and the identifiers of functions f() and h() to be used in the response. At the user end, the token provides a passcode **W’.**  The token either stores a static passcode or generates a one-time random passcode. For a one-time random passcode, the token must be synchronized in some fashion with the host. In either case, the user activates the passcode by entering a password **P’.**  This password is shared only between the user and the token and does not involve the remote host.The token responds to the host with the quantity f(**r’**, h(**W’** )). For a static passcode, the host stores the hashed value h(**W** (**U** )); for a dynamic passcode, the host generates a one-time passcode (synchronized to that generated by the token) and takes its hash.Authentication then proceeds in the same fashion as for the password protocol.
* Figure c is an example of a user authentication protocol using a static biometric.As before, the user transmits an ID to the host, which responds with a random number and, in this case, the identifier for an encryption E(). On the user side is a client system that controls a biometric device. The system generates a biometric template BT from the user’s biometric B’ and returns the ciphertext E(r’, D’, BT’) , where D’ identifies this particular biometric device. The host decrypts the incoming message to recover the three transmitted parameters and compares these to locally stored values.For a match, the host must find r’ = r . Also, the matching score between BT’ andthe stored template must exceed a predefined threshold. Finally, the host provides a simple authentication of the biometric capture device by comparing the incoming device ID to a list of registered devices at the host database.
* Figure d is an example of a user authentication protocol using a dynamic biometric.The principal difference from the case of a stable biometric is that the host provides a random sequence as well as a random number as a challenge. The sequence challenge is a sequence of numbers, characters, or words. The human user at the client end must then vocalize (speaker verification), type (keyboard dynamics verification), or write (handwriting verification) the sequence to generate a biometric signal BS’ (x’) . The client side encrypts the biometric signal and the random number. At the host side, the incoming message is decrypted. The incoming random number r’must be an exact match to the random number that was originally used as a challenge (r ). In addition, the host generates a comparison based on the incoming biometric signal BS’ (x’) , the stored template BT (U ) for this user and the original signal x . If the comparison value exceeds a predefined threshold, the user is authenticated.

**CHAPTER 4**

**IMPLEMENTATION**

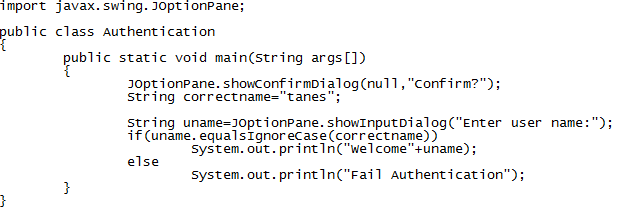


Figure 4.1

Source code for user authentication

**CHAPTER 5**

**RESULTS AND ANALYSIS**

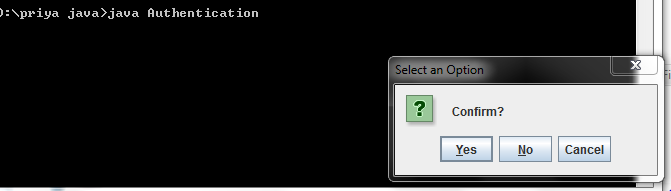
****

Figure 5.1

Dialog Box to choose an option to enter a username

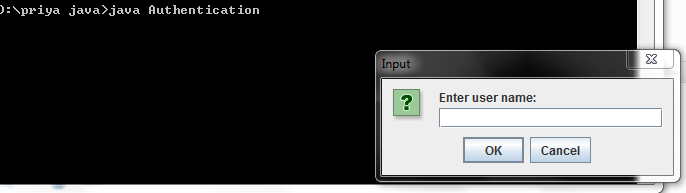


Figure 5.2

Dialog Box to enter a user name

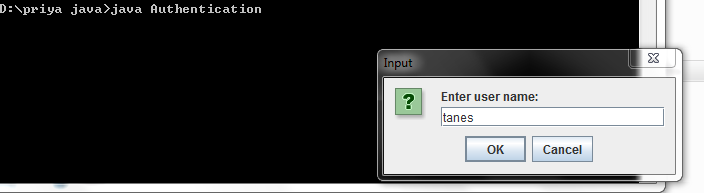


Figure 5.3

Dialog Box showing authenticated username entered

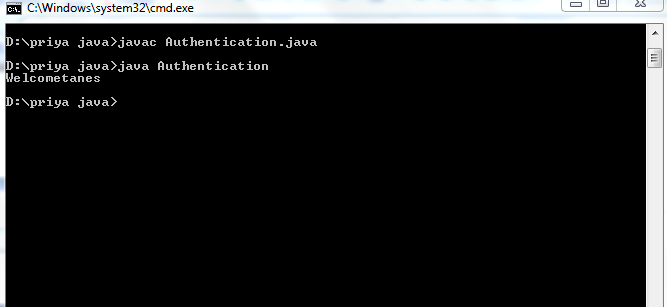


Figure 5.4

Output showing that username entered is an authenticated user

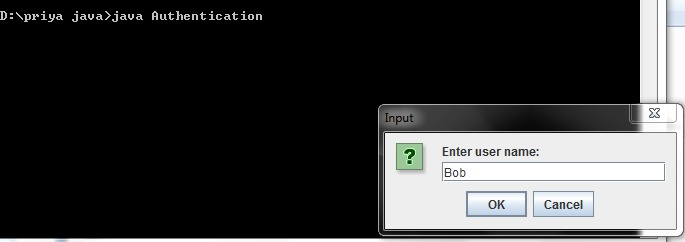


Figure 5.5

Dialog Box to enter an invalid username

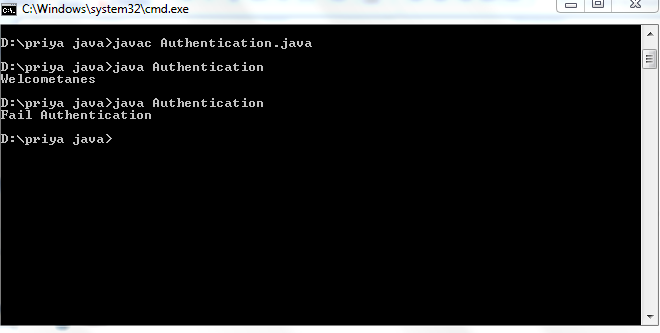


Figure 5.6

Output showing that username entered is an invalid user

**CONCLUSION**

Several dynamic user authentication schemes have weaknesses and attacks because of using timestamps.Besides, the implement of strict and safe time synchronization is very difficult and increases network overhead. To eliminate these weaknesses,a new dynamic user authentication scheme based on nonce instead of timestamps has to be proposed. Mutual authentication is performed using a challenge-response handshake between user and remote server.

**REFERENCES**

[1] Lamport, L. (1981) Password Authentication with Insecure Communication. Communications of the ACM, 24, 770-772. http://dx.doi.org/10.1145/358790.358797

[2] Yoon, E.J., Ryu, E.K. and Yoo, K.Y. (2004) Further Improvement of an Efficient Password Based Remote User Authentication Scheme Using Smart Cards. IEEE Transactions on Consumer Electronics, 50, 612-614.http://dx.doi.org/10.1109/TCE.2004.1309437

[3] Tina, X., Zhu, R.W. and Wong, D.S. (2007) Improved Efficient Remote User Authentication Schemes. International Journal of Network Security, 4, 149-154.

[4] Yang, L. and Ma, J.F. (2011) Trusted Mutual Authentication Scheme with Smart Cards and Passwords. Journal of University of Electronic Science and Technology of China, 4, 128-133.

[5] Das, M.L., Saxena, A. and Gulati, P. (2004) A Dynamic Id-Based Remote User Authentication Scheme. IEEE Transactions on Consumer Electronics, 50, 629-631. http://dx.doi.org/10.1109/TCE.2004.1309441