Establishing Confidence Level Measurements for Remote User Authentication in Privacy-Critical Systems

by

Matthew Robertson

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2009

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

User Authentication is the process of establishing confidence in the User identities presented to an information system. This thesis establishes a method of assigning a confidence level to the output of a user authentication process based on what attacks and threats it is vulnerable to. Additionally, this thesis describes the results of an analysis where the method was performed on several different authentication systems and the confidence level in the authentication process of these systems determined. Final conclusions found that most systems lack confidence in their ability to authenticate users as the systems were unable to operate in the face of compromised authenticating information. Final recommendations were to improve on this inadequacy, and thus improve the confidence in the output of the authentication process, through the verification of both static and dynamic attributes of authenticating information. A system that operates confidently in the face of compromised authenticating information that utilizes voice verification is described demonstrating the ability of an authentication system to have complete confidence in its ability to authenticate a user through submitted data.

Acknowledgements

I would like to thank Professor Gordon Agnew of the University of Waterloo for his advice, guidance, instruction and supervision. Professor Agnew first introduced me to the topic of cryptography and information security and possesses a wealth of information on the subject that in the short time that I have had the privilege of working with him I have been able to tap. His insights and suggestions have been invaluable in the creation of this thesis. I will remain forever in his debt for his instruction and assistance in the establishment of my future career.

Table of Contents

Appendices:

List of Figures

List of Tables

Chapter 1
User Authentication

1.1 Introduction

A fundamental concern in information systems is the authentication of local and remote entities. This concern arises from the necessity to verify that an entity attempting to gain access to the system is indeed who or what they claim to be to ensure the protection and privacy of the data inside the system. In the case of user authentication the system must be able to provide assurances that the user is who they claim to be in order to grant access privileges to the user: this process is known as the user authentication process.

User authentication supports further security services beyond access control including privacy, data authentication and non-repudiation. Privacy, similar to access control, is accomplished by ensuring that only authorized individuals can view the data. Data authentication and non-repudiation are accomplished by saving the identity of the user along with the data. These services depend on the ability of the system to ensure that a user is authenticated correctly and that there are no falsely authenticated users.

Almost all information systems today provide a means of attempting to authenticate a user before allowing him or her access to the system. This thesis examines remote user authentication as it occurs in privacy critical systems.

In Chapter 1, this thesis will first provide a brief overview of cryptography before it will explore user authentication as it is understood today by identifying the entities involved and establish what will be referred to as the Generalized User Authentication Process. Chapter 1 will continue on to describe the three known factors of authentication: “something you know”, “something you have”, and “something you are”; special emphasis will be placed on “something you are”, commonly known as biometric authentication.

Chapter 2 will define the requirements that are needed in the authentication process of a remote privacy critical system. Chapter 2 will first explore the weaknesses in the three different authentication factors before highlighting the weaknesses of the Generalized User Authentication Process and defining attacks on the process. After exploring and categorizing the attacks an understanding of the requirements for a privacy-critical system to have confidence in the output of the authentication process will be established.

Chapter 3 will contain case studies on several well known and frequently used privacy-critical systems with remote user authentication. The case studies will contain an analysis of the user authentication process highlighting any weaknesses and apply the measures assigned in Chapter 2 to establish the confidence level in the output of the authentication process. Recommendations will be made as necessary on how to improve the confidence level.

The final chapter, Chapter 4, will summarize the findings of the analysis and propose and analyze an authentication system that meets the requirements to have complete confidence in its output.

1.2 Overview of Cryptography

Cryptography is the "study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication." [1] The fundamental goal of cryptography is to address these issues and allow for the "protection and detection of cheating and other malicious activities." [1] This section will provide a very brief definition and overview of the cryptographic primitives used in user authentication and how these primitives aid the process.

1.2.1 Cryptographic Goals

The information security objectives contained within the definition of cryptography form a framework upon which other information security objectives can be derived. The following definitions of the four objectives were obtained from the Handbook of Applied Cryptography. [1]

1. **Confidentiality** is a service used to keep the content of information from all but those authorized to have it. Secrecy is a term synonymous with confidentiality and privacy. There are numerous approaches to providing confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.

2. **Data integrity** is a service which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution.

3. **Authentication** is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. For these reasons this aspect of cryptography is usually subdivided into two major classes: entity authentication and data origin authentication. Data origin authentication implicitly provides data integrity (for if a message is modified, the source has changed).

4. **Non-repudiation** is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary. For example, one entity may authorize the purchase of property by another

entity and later deny such authorization was granted. A procedure involving a trusted third party is needed to resolve the dispute.

1.2.2 Encryption

Figure 1.1 illustrates two parties, Alice and Bob, who are communicating over an unsecured channel. Alice and Bob however require that their communication be confidential; to accomplish this goal encryption is used.

In Figure 1.1 Alice has a plaintext message (*m*) she wishes to confidentially send to Bob, so before transmission it is passed through an encryption algorithm (E_e) to produce a ciphertext (*c*) which is then transmitted to Bob. Bob then passes the ciphertext through a decryption algorithm (D_d) to recover the original plaintext message.

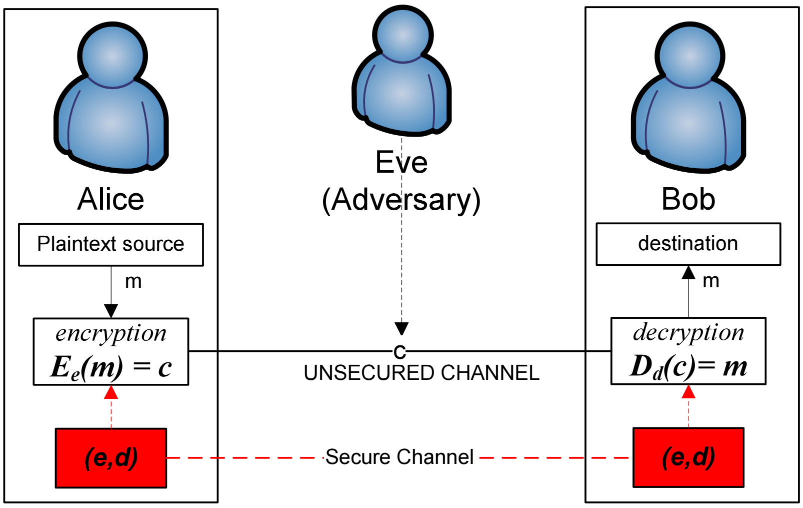


Figure 1.1 - Two-party Communication using Encryption

1.2.2.1 Symmetric-Key Cryptography

> **Definition:** Consider an encryption scheme consisting of the sets of encryption and decryption transformations { $E_e : e \in \mathrm{K}$ } and { $D_d : d \in \mathrm{K}$ }, respectively, where K is the key space. The encryption scheme is said to be symmetric-key if for each associated encryption/decryption key pair (*e, d*), it is computationally "easy" to determine *d* knowing only *e*, and to determine *e* from *d*. Since *e=d* in most practical symmetric-key encryption schemes, the term symmetric-key becomes appropriate. [1]

Figure 1.1 is an illustration of a symmetric key encryption scheme. In Figure 1.1 Alice and Bob first agree upon a symmetric key through an offline key exchange protocol. Alice and Bob then use that key to encrypt and decrypt messages passed between each other. A problem that is immediately obvious in symmetric key cryptography is how Alice and Bob can efficiently agree upon the symmetric key: this is known as the key distribution problem.

Symmetric key encryption schemes are divided into two classes: block ciphers and stream ciphers. A block cipher is an encryption scheme which divides the plaintext message into multiple blocks of fixed length and encrypts one block at a time. A stream cipher is an encryption scheme that converts a plaintext message to ciphertext one bit at a time.

Stream ciphers operate through the use of a keystream generator, which is delivered to Alice and Bob before the communication. The keystream generator generates the next bit of the key which is combined with the next bit of the plaintext in an *exclusive-OR* function. The security of the scheme depends entirely on the implementation of the keystream generator: if the next bit of the keystream is predictable with probability greater than 50% then the encryption is weak.

Most well-known symmetric-key encryption techniques are block ciphers. [1] Claude Shannon in his landmark paper Communication Theory of Secrecy Systems defined two characteristics of block ciphers: confusion and diffusion. [2]

Confusion, also known as substitution, is the process of making the relationship between the ciphertext and key as complex as possible by replacing symbols in the plaintext with other symbols to form the cipher text. Diffusion, also known as transposition, is the process of spreading the effect of the plaintext or key as widely as possible over the ciphertext by permutating the symbols in a block of ciphertext. A good block cipher should have both confusion and diffusion properties, as a result most block ciphers are compositions of several rounds of confusion and diffusion techniques. Common block ciphers used in practice are: the Data Encryption Standard (DES), Triple-DES, and the Advanced Encryption Standard (AES).

The security of symmetric key encryption is based on the idea of making the ciphertext appear as random as possible. Essentially this means that ideally the only method for the adversary (Eve) to recover the plaintext from the ciphertext is to perform an exhaustive search on the keyspace.

Therefore "a necessary, but usually not sufficient, condition of the encryption scheme to be secure is that the key space be large enough to preclude an exhaustive search." [1]

In an authentication system, "cryptography is used to guarantee the authenticity of the message to the receiver." [3] This means that an adversary must be prevented from adding, deleting or modifying messages between the sender and receiver.

1.2.2.2 Public-Key Cryptography

The most significant limitation in symmetric-key cryptography is the requirement that both Alice and Bob must possess a shared key before communication begins. In many situations this is unfeasible: Alice may not know Bob before the communication and/or there is no secure method of exchanging keys. Additionally in a system of *n* users each user must maintain a set of $(n\text{-}1)$ keys, leading to a total of $\frac{n(n-1)}{2}$ keys in the system. These limitations in symmetric-key cryptography can be addressed through public key cryptography.

In a public-key cryptosystem, first described by Diffie and Hellman [3] in 1976, two communicating users exchange a key by communicating back and forth across an unsecured channel until arrive at a key in common; a third party eavesdropping on this exchange must find it computationally infeasible to compute the key from the information overheard. Currently there are two problems that are considered viable to provide the necessary computational infeasibility of public-key cryptosystems:

1. the factoring problem
2. the discrete logarithm problem

The factoring problem is "based on the believed difficulty of factoring the product of two large primes" [4] while the discrete logarithm problem is "based on the difficulty of finding logarithms in a finite field." [4]

Figure 1.2 illustrates encryption using public-key techniques. Figure 1.2 differs noticeably from Figure 1.1 through the absence of the secure channel where the symmetric key was previously exchanged. In Figure 1.2 the encryption key (e) is sent by Bob to Alice through an unsecured channel, Alice then uses that key to encrypt the message ($E_e(m)$) and Bob decrypts the ciphertext ($D_d(c)$) using the decryption key (d). The public-key cryptosystem illustrated in Figure 1.2 gives rise to the below definition of public-key encryption:

Definition: Consider an encryption scheme consisting of the sets of encryption and decryption transformations $\{ E_e : e \in \mathrm{K} \}$ and $\{ D_d : d \in \mathrm{K} \}$, respectively. The encryption method is said to be a *public-key encryption scheme* if for each associated encryption/decryption pair (*e, d*), one key *e* (*the public key*) is made publically available, while the other *d* (*the private key*) is kept secret. For the scheme to be *secure*, it must be computationally infeasible to compute *d* from *e*. [1]

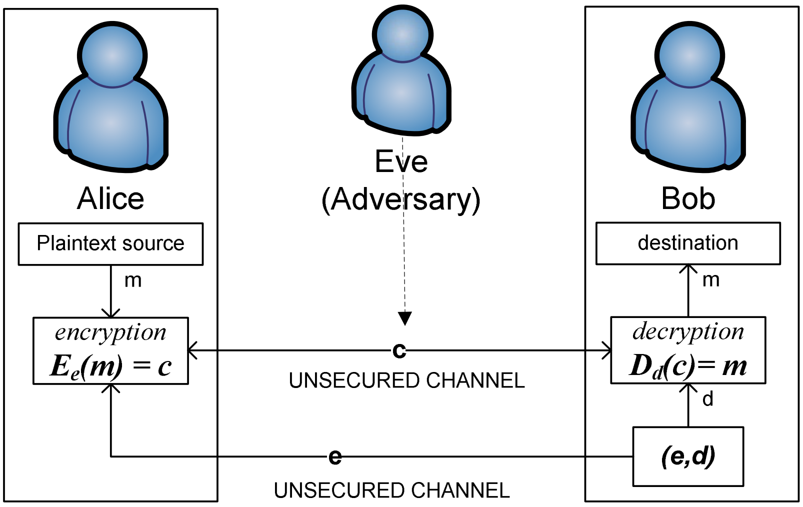


Figure 1.2 - Public-Key Encryption

Authentication is necessary in public-key cryptosystems due to the man-in-the-middle attack illustrated in Figure 1.3. In this attack Eve positions herself between Alice and Bob and impersonates Alice to Bob and Bob to Alice; as far as Alice and Bob are concerned they are communicating only with each other and have no knowledge of Eve's presence. Elegant solutions to the authentication issues in public-key cryptosystems are addressed through a Public-Key Infrastructure (PKI) and public-key Certificates discussed in Section 1.2.5.

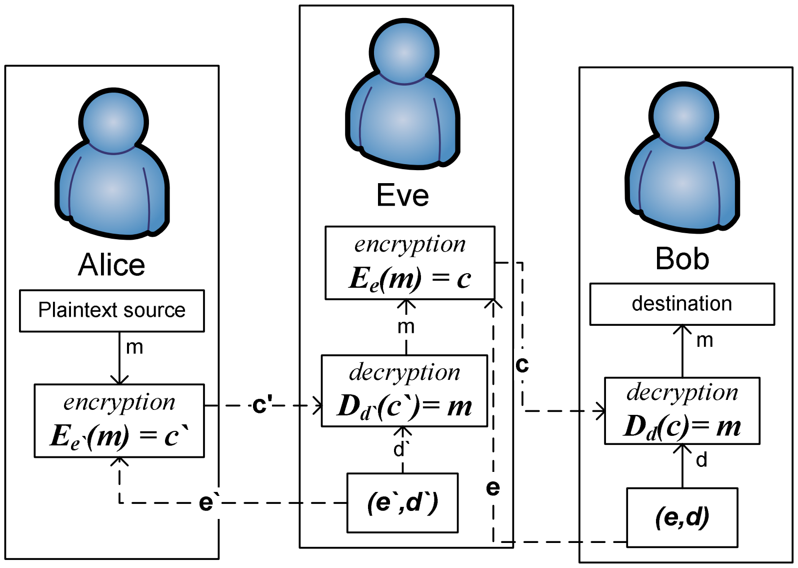


Figure 1.3 - Man-in-the-Middle Attack on a Public-key Cryptosystem

RSA is "probably the most widely used public-key cryptosystem in the world." [5] However, RSA cryptosystems can be computationally intensive and restrictive in low power environments. Systems using a version of the discrete logarithm problem using a field of the form GF(2^n), can provide "very fast and very secure public-key systems." [6]

1.2.2.3 RSA Public-Key Encryption

The RSA public-key cryptosystem is named after its inventors R. Rivest, A. Shamir, and L. Adleman was first proposed in 1977 [7] and is based on the assumed intractability of the factoring problem. This section will briefly describe the key generation algorithm and the encryption/decryption algorithms used in this cryptosystem. The following algorithmic descriptions were obtained from the Handbook of Applied Cryptography. [1]

1.2.2.3.1 RSA Key Generation Algorithm

Summary: each entity creates an RSA public key and a corresponding private key.
Each entity *A* should do the following:

1. Generate two large random (and distinct) primes *p* and *q*, each roughly the same size.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$
3. Select a random integer *e*, $1 < e < \phi$, such that $\gcd(e, \phi) = 1$.
4. Use the extended Euclidean algorithm to compute the unique integer *d*, $1 < d < \phi$, such that $ed \equiv 1 (\bmod \phi)$
5. *A*'s public key is (*n*, *e*); *A*'s private key is *d*.

1.2.2.3.2 RSA Public Key Encryption Algorithm

Summary: *B* encrypts a message *m* for *A*, which *A* decrypts

Encryption:

1. Obtain *A*'s authentic public key (*n* ,*e*)
2. Represent the message as an integer *m* in the interval [0, *n*-1]
3. Compute $c \equiv m^e (\bmod n)$
4. Send the ciphertext *c* to *A*.

Decryption

To recover plaintext *m* from *c*, *A* should do the following:

1. Use the private key *d* to recover $m \equiv c^d (\bmod n)$

1.2.2.4 Discrete Logarithm Cryptosystems

The very first discrete logarithm system was proposed by Diffie and Hellman in 1976 and described in their key exchange algorithm [3] while the basic public-key encryption scheme utilizing discrete logarithms was proposed in 1984 by ElGamal. [8] This section briefly describes the basic ElGamal public-key encryption scheme using discrete logarithms; the algorithmic descriptions below were obtained from the Guide to Elliptic Curve Cryptography. [9]

In discrete logarithm systems, a key pair is associated with a set of public domain parameters (p, q, g). Here, *p* is a prime, *q* is a prime divisor of $(p-1)$, and $g \in [1, p-1]$ and has order *q*. The private key *x* is an integer randomly selected from the interval $[1, q-1]$ and the corresponding public key is $y = g^x \bmod p$. The discrete logarithm problem is the problem of determining *x* given domain parameters (p, q, g) and *y*.

1.2.2.4.1 Discrete Logarithm Domain Parameter Generation

INPUT: Security parameters *l, t.*

OUTPUT: Discrete Logarithm domain parameters (p, q, g).

```
1. Select a $t$-bit prime $q$ and an $l$-bit prime $p$ such that $q$ divides $p$-1
2. Select an element $g$ of order $q$:
    a. Select arbitrary $h \in [1, p-1]$ and compute $g = h^{\frac{(p-1)}{q}} \bmod p$
    b. If $g = 1$ then go to step a
3. Return $(p, q, g)$
```

1.2.2.4.2 Discrete Logarithm Key Pair Generation

INPUT: Discrete Logarithm domain parameters (p, q, g).

OUTPUT: Public key *y* and private key *x*.

```
1. Select $x \in_R [1, q-1]$
2. Compute $y = g^x \bmod p$
3. Return $(x, y)$
```

1.2.2.4.3 Basic ElGamal Encryption

INPUT: Discrete Logarithm domain parameters (p, q, g), public key *y*, plaintext $m \in [1, p-1]$

OUTPUT: Ciphertext (c_1, c_2)

```
1. Select $k \in_R [1, q-1]$
2. Compute $c_1 = g^k \bmod p$
3. Compute $c_2 = m \cdot y^k \bmod p$
4. Return $(c_1, c_2)$
```

1.2.2.4.4 Basic ElGamal Decryption

INPUT: Discrete Logarithm domain parameters (p, q, g), private key *x*, ciphertext (c_1, c_2)

OUTPUT: Plaintext *m*

1. Compute $m = c_2 \cdot c_1^{-x} \bmod p$
2. Return m

1.2.2.5 Elliptic Curve Cryptosystems

Elliptic curve cryptosystems are discrete logarithm cryptosystems described in the abstract setting of a finite cyclic group. [9] This section provides a brief description of elliptic curve cryptosystems and assumes an elementary knowledge of group theory; the algorithmic descriptions below were obtained from the Guide to Elliptic Curve Cryptography. [9]

A multiplicative cyclic group $(G,\cdot)$ of order *n* with generator *g* can be used to describe the discrete logarithm problem. In the setting of *G* the domain parameters are *g* and *n*, the private key is an integer *x* selected randomly from the interval $[1, n-1]$ and the public key is $y = g^x$. The problem of determining *x*, given *g, n* and *y* is the discrete logarithm problem in *G*.

The cyclic subgroup of elliptic curve groups can be used to form *G*. An elliptic curve *E* over F_p (where F_p denotes the field of integers modulo *p*) is defined by

$$y^2 = x^3 + ax + b,$$

Where $a, b \in F_p$ satisfy $4a^3 + 27b^2 \neq 0 (\bmod p)$. A pair (x, y), where $x, y \in F_p$ is a point on the curve *E* if (x, y) satisfy the above equation defining the curve *E*.

If *E* is an elliptic curve defined on the finite field F_p and *P* has prime order *n* and is a point in $E(F_p)$ then the cyclic subgroup $E(F_p)$ generated by *P* is

$$\langle P \rangle = \{\infty, P, 2P, 3P, \ldots, (n-1)P\}$$

The prime *p*, the equation of the elliptic curve *E*, and the point *P* and its order *n*, are the public domain parameters. The private key is an integer *d* that is selected uniformly at random from the interval $[1, n-1]$ and the corresponding public key is $Q = dP$. The problem of determining *d* given the domain parameters and *Q* is the Elliptic Curve Discrete Logarithm Problem (ECDLP).

A plaintext *m* is first represented as a point *M*, and then encrypted by adding it to *kQ* where *k* is a randomly selected integer, and *Q* is the intended recipient's public key. The sender transmits the points $C_1 = kP$ and $C_2 = M + kQ$ to the recipient who uses their private key d to compute $dC_1 = d(kP) = k(dP) = kQ$ and thereafter recovers $M = C_2 - kQ$.

1.2.2.5.1 Elliptic Curve Key Pair Generation

INPUT: Elliptic Curve domain parameters (p, E, P, n).

OUTPUT: Public key Q and private key d.

```
1. Select $d \in_R [1, n-1]$
2. Compute $Q = dP$
3. Return $(Q, d)$
```

1.2.2.5.2 Basic ElGamal Elliptic Curve Encryption

INPUT: Elliptic Curve domain parameters (p, E, P, n), public key Q, plaintext m.

OUTPUT: Ciphertext (C_1, C_2)

```
1. Represent the message $m$ as a point $M$ in $E(F_p)$
2. Select $k \in_R [1, n-1]$
3. Compute $C_1 = kP$
4. Compute $C_2 = M + kQ$
5. Return $(C_1, C_2)$
```

1.2.2.5.3 Basic ElGamal Elliptic Curve Encryption

INPUT: Elliptic Curve domain parameters (p, E, P, n), private key d, plaintext (C_1, C_2)

OUTPUT: Plaintext m

```
1. Compute $M = C_2 - dC_1$ and extract $m$ from $M$
2. Return $M$
```

1.2.2.6 Key Size Comparison

Table 1.1 illustrates estimates for parameter sizes providing comparable levels of security for RSA, Discrete Logarithm (DL) and Elliptic Curve (EC) cryptosystems. The estimates were based on the time taken for the best known algorithms to solve the respective problems. For instance the best known algorithm to solve the integer factorization problem is the Number Field Sieve, while the best known algorithms to solve the discrete logarithm problem are the Number Field Sieve and Pollard's rho algorithm. Pollard's rho algorithm is also the best known algorithm to solve the ECDLP.

Table 1.1 - RSA, DL and EC key sizes for equivalent security levels [9]

	Security Level (bits)				
	80 (SKIPJACK)	112 (TRIPLE-DES)	128 (AES-Small)	192 (AES-Medium)	256 (AES-Large)
DL parameter *q*	160	224	256	384	512
EC parameter *n*					
RSA modulus *n*	1024	2048	3072	8192	15360
DL modulus *p*					

1.2.3 Digital Signatures

Digital signatures are particularly useful in entity and data authentication, authorization and non-repudiation. A Digital Signature is a transformation that "provides a means for an entity to bind its identity to a piece of information." [1] Figure 1.4 illustrates a digital signature scheme in which Alice creates a signature (s) for a message (m) through a signature function (S_A) before transmitting the message and signature to Bob. Bob then computes a Boolean function $u = V_A(m, s)$ to verify the signature on the message: Bob will accept the signature as being created by Alice if the function returns true and reject the signature if the function returns false.

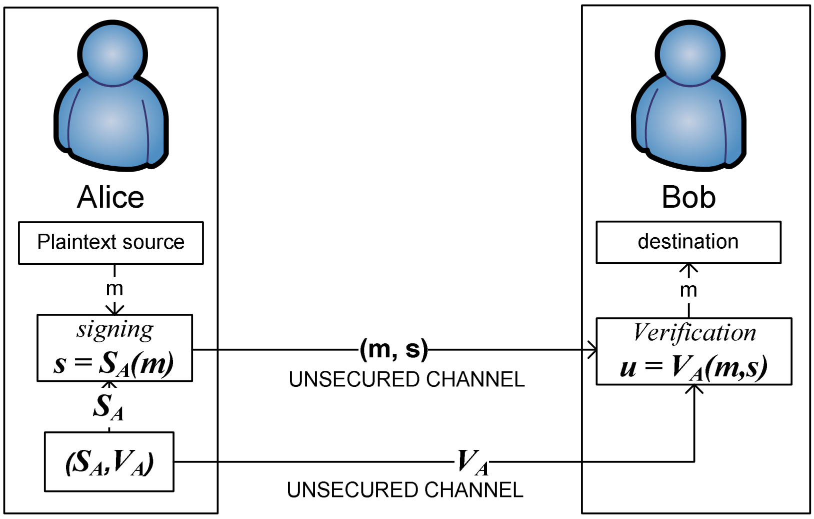


Figure 1.4 - Digital Signature Scheme

Public-key cryptography is especially useful in establishing the transformation functions $S_A(m)$ and $V_A(m)$. In a public-key signature scheme the transformation $S_A(m)$ can be represented as $S_A(m) = D_{d_A}(m)$, where $D_{d_A}(m)$ is the decryption function using Alice's private key. In the same scheme the transformation $V_A(m)$ can be represented as $V_A(m,s) = \begin{cases} true, if : E_{e_A}(s) = m \\ false, otherwise \end{cases}$, where $E_{e_A}(s)$ is the encryption function using Alice's public key.

1.2.4 Hash Functions

A *hash function* is "a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called *hash-values*." [1] A hash-value of a string acts as a compact representation of the string. A cryptographic hash function *h* is one such that it is "computationally infeasible to find two distinct inputs which hash to a common value (i.e. two colliding inputs *x* and *y* such that *h(x)=h(y)*), and that given a specific hash-value *y*, it is computationally infeasible to find an input (pre-image) *x* such that *h(x)*=y." [1]

Cryptographic hash functions are most often used in conjunction with digital signatures for data integrity. In this situation a long message is hashed and the hash-value is digitally signed. The receiving party then hashes the original message and verifies that the digital signature on the received hash-value is correct. This scheme is particularly useful as it saves time and space compared to signing the message directly using an encryption scheme and transmitting the original message and the signature.

1.2.5 Key Management and Public-Key Certificates

The key distribution problem, mentioned in Section 1.2.2.1, can be addressed through the use of public-key cryptography. The Diffie-Hellman Key Exchange, [3] based on the intractability of the discrete logarithm problem, is illustrated in Figure 1.5. The Diffie-Hellman Key Exchange uses public-key cryptography to establish a secret symmetric key between Alice and Bob.

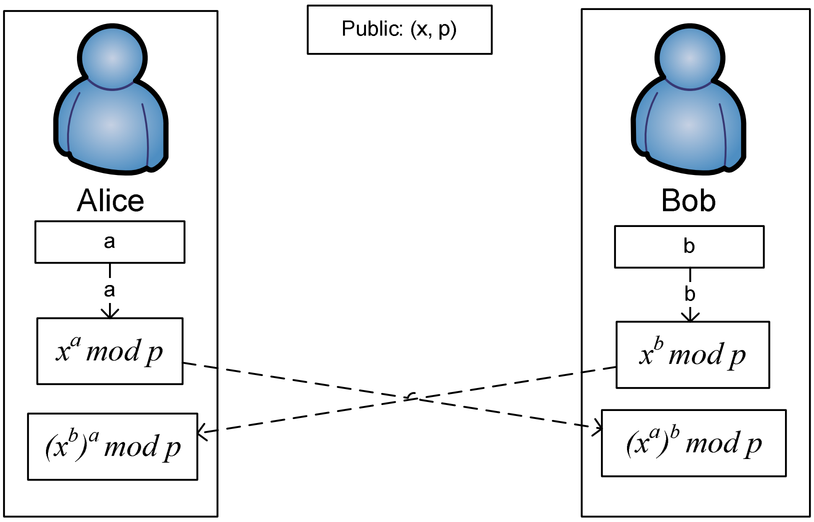


Figure 1.5 - Diffie-Hellman Key Exchange

In Figure 1.5 Alice possesses a public key (*x, p*) and a private key *a*, while Bob posses a public key (x, p) and a private key b. Alice computes x^a mod p and sends it to Bob. Bob computes x^b mod p and sends it to Alice. Both compute the new shared key: $(x^b)^a$ mod $p = (x^a)^b$ mod p. Due to the intractability of the discrete logarithm problem neither Alice or Bob can compute the other's private key from the shared key and an eavesdropper cannot compute either *a, b or* $(x^b)^a$ mod p. Similar key exchange algorithms can be developed using other public key techniques.

It can be observed in Figure 1.5 that the Man-in-the-Middle attack is still possible as there is no authentication of either Alice or Bob. A solution to the authentication problem is through the use of a Trusted Third Party (TTP) and public-key certificates. Figure 1.6 illustrates the concept behind a public-key certificate.

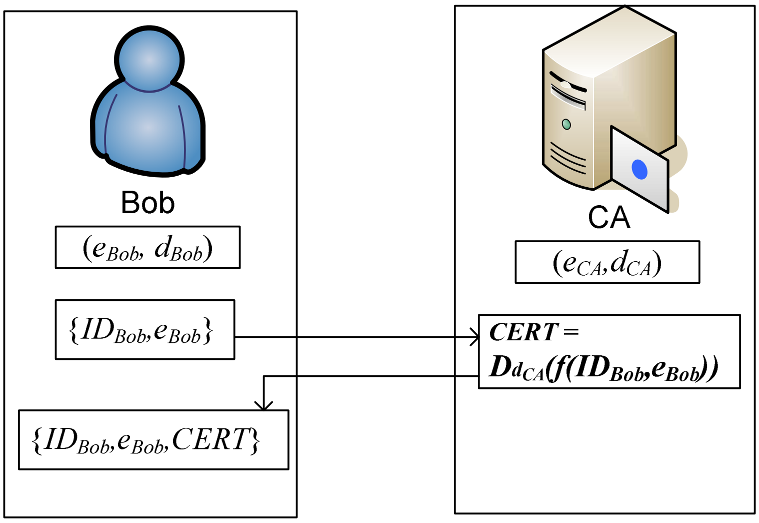

Bob
(e_{Bob}, d_{Bob})
$\{ID_{Bob}, e_{Bob}\}$
$\{ID_{Bob}, e_{Bob}, CERT\}$
CA
(e_{CA}, d_{CA})
$CERT = D_{d_{CA}}(f(ID_{Bob}, e_{Bob}))$

Figure 1.6 - Public-key Certificates

In the public-key certificate scheme illustrated in Figure 1.6 Bob generates his own public and private key pair, (e_{Bob}, d_{Bob}), which along with some identifying information (ID_{Bob}) is sent to the Certificate Authority (CA). The CA computes $CERT = D_{d_{CA}}(f(ID_{Bob}, e_{Bob}))$ where *f()* is a function of combining an ID, and a public-key into a well-known structure; the result of the function is then digitally signed by the CA.

Figure 1.7 illustrates an example protocol using public-key certificates to authenticate two communicating parties at the start of the communication. In Figure 1.7 Alice and Bob both have assigned identities (ID_A, ID_B) their own public and private key pair [(e_A,d_A) and (e_B,d_B)] and certificates obtained from the CA ($CERT_A$ and $CERT_B$). Additionally Alice and Bob both have pre-existing knowledge of the other's identity, public keys and the public key of the CA, illustrated with the dotted lines.

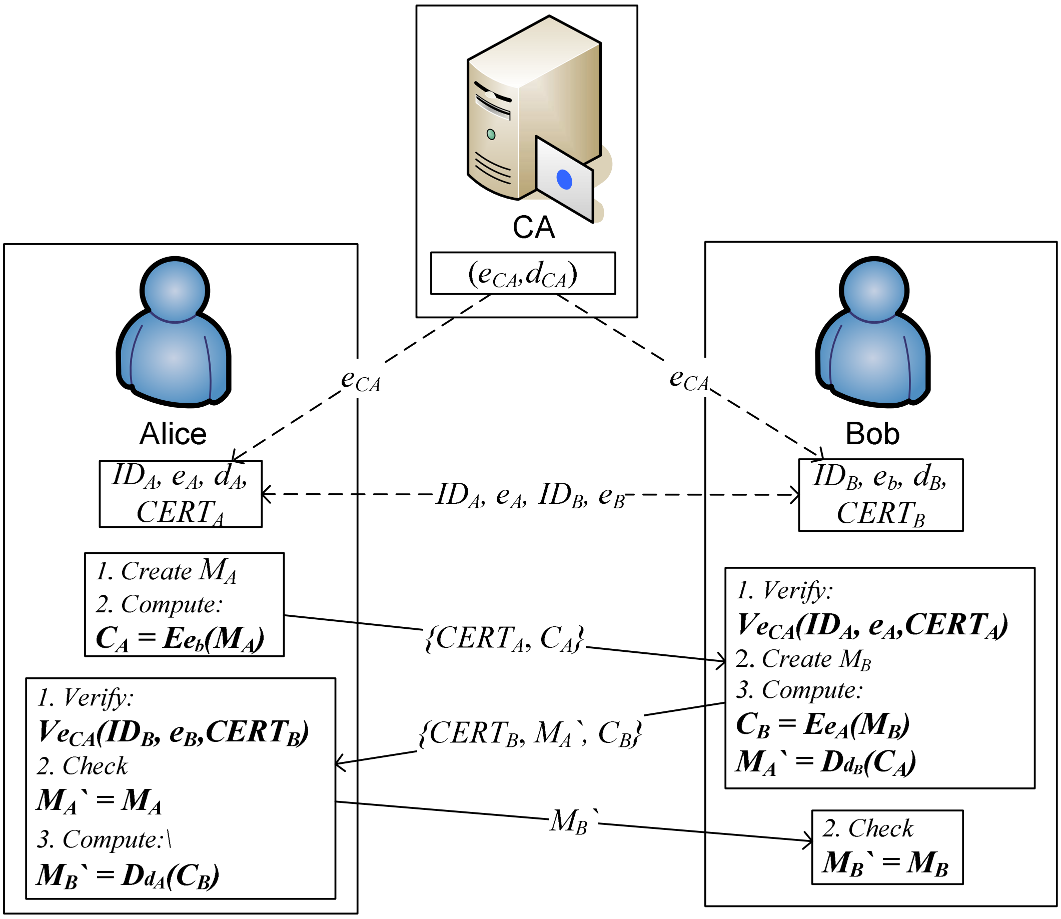


Figure 1.7- Authentication using Public-Key Certificates

In Figure 1.7 Alice first creates a random message, M_A, which she encrypts with Bob's public key to compute C_A; this message will be used to verify that Bob is in possession of his private key. Alice then sends her certificate and C_A to Bob: it should be noted that this transmission should have some form of data integrity such as a hash of the message signed by Alice to prevent a man in the middle attack. Bob will then verify that the received certificate is legitimate using the verification function $V_{e_{CA}}(ID_A e_A, CERT_A) = \begin{cases} true, if: E_{e_{CA}}(CERT_A) = f(ID_A, e_A) \\ false, otherwise \end{cases}$. If the verification passes Bob will create his own random message M_B, and compute C_B using Alice's public key and compute $M_A`$ using his own private key. Bob next sends his certificate, $M_A`$, and C_B to Alice. Alice then verifies Bob's certificate, checks that $M_A`$ matches $M_A$, and computes $M_B`$. Alice next sends $M_B`$ to Bob who checks that it match M_B.

At the end of the authentication protocol illustrated in Figure 1.7 Alice and Bob are assured of each other's identity through their demonstrated possession of a certificate issued by the TTP and the corresponding private key. Again it should be noted that this assurance is only present if a means of data integrity is included in the protocol to ensure that an adversary has not modified any of the messages in transit.

1.2.6 Secured Channel

A secured channel between two parties is a bi-directional channel conveying information in which "an adversary does not have the ability to reorder, delete, insert or read" [1] and is of particular interest in privacy-critical systems. The cryptographic primitives and techniques described previously can be used to create such a channel. Figure 1.8 illustrates an example of the steps involved in establishing a secured channel between Alice and Bob.

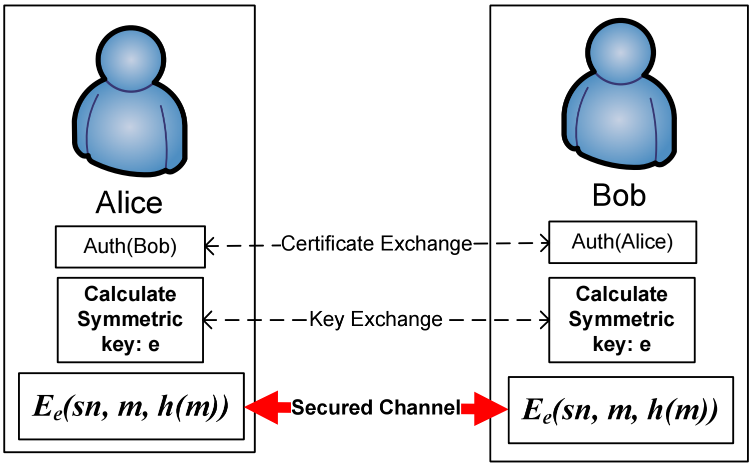


Figure 1.8 - Secured Channel Establishment

In Figure 1.8 Alice and Bob first authenticate each other using public key certificates exchanged with a means of data integrity such as a signed hash of the transmitted data. Once the identities are authenticated Alice and Bob invoke a key exchange method such as the Diffie-Hellman key exchange to calculate a shared key *e*. Secrecy of the communication between Alice and Bob can is obtained through the encryption of the data before transmit.

To ensure that an adversary cannot delete or reorder messages in the communication sequence numbers are also included in each message in the communication. It should be noted that these sequence numbers are also encrypted before being sent to ensure that an adversary does not discover them. Alice and Bob check the sequence number upon the receipt of the message and if it is not what expected will react accordingly.

To ensure that an adversary cannot insert data into the communication a means of data integrity is required. A good method to ensure data integrity is to compute the hash value of the message, and encrypt the hash value with the shared key before sending the hash value with the message. Communications between Alice and Bob on the secured channel will be of the form:

$E_e(sn, m, h(m))$, where *sn* is the sequence number, *m* is the message and *h(m)* is the hash value of *m*.

1.3 Overview of User Authentication

User Authentication is the process of establishing confidence in user identities presented to an information system. [10] This chapter will explore the authentication process establishing what will be referred to as the Generalized User Authentication Process, as well as understanding how the process establishes confidence in user identities.

The authentication process is divided into two separate phases: the Enrolment Phase (sometimes referred to as the Registration Phase) and the Authentication Phase (sometimes referred to as the Verification Phase).

In the Enrolment Phase an individual (the User) applies to the Registration Authority (RA) to become a registered user of the Credential Service Provider (CSP). If approved by the RA the user is issued with a credential, such as a user name, and is either issued or creates a token, such as a password, that binds the credential to the user's identity. The <credential, token> pair are submitted by the user during the Authentication Phase.

The goal of the Authentication Phase is to ensure that the party providing an identity and requesting authentication, known as the Claimant, is correctly verified, by a party known as the Verifier, as possessing the claimed identity. In the Authentication Phase the Claimant presents a <credential, token> pair to the Verifier; the Verifier then checks if the token is indeed the token that was registered to the credential in the Enrolment Phase. If the correct match is found by the Verifier the Claimant is assumed, through possession of a correct <credential, token> pair, to posses the identity associated with that pair. The Verifier then passes an assertion about the identity of the authenticated user to the Relying Party to allow for authorization and access control decisions to be made.

Access control and authorization decisions made by the Relying Party are beyond the scope of this thesis. This thesis focuses on how the output of the Authentication Process is determined rather than how it is acted upon by the Relaying Party.

Figure 1.9 illustrates the various entities and their interactions that are involved in the User authentication process. The box on the left represents the entities involved in the Enrolment Phase, while the box on the right represents the entities involved in the Authentication Phase. The interactions of the illustrated entities are as follows:

Enrolment Phase:

1. An individual applies to an RA through the system's enrolment process.
2. The RA identity proofs that Applicant.

3. On successful identity proofing, the RA sends the CSP a registration confirmation message.
4. A secret token and a corresponding credential are established between the CSP and the new user.
5. The CSP maintains the credential, its status, and the registration data collected. The user maintains his or her <credential, token> pair.

Authentication Phase:

1. The Claimant proves to the Verifier that he or she possesses and controls the <credential, token> pair through an authentication protocol.
2. The Verifier interacts with the CSP to validate the token and credential and confirm that the Claimant is a user of the CSP.
3. If the Verifier is separate from the Relying Party (application), the Verifier provides an assertion about the Claimant to the Relying Party, which uses the information in the assertion to make an access control or authorization decision.
4. An authenticated session is established between the Claimant and the Relying Party.

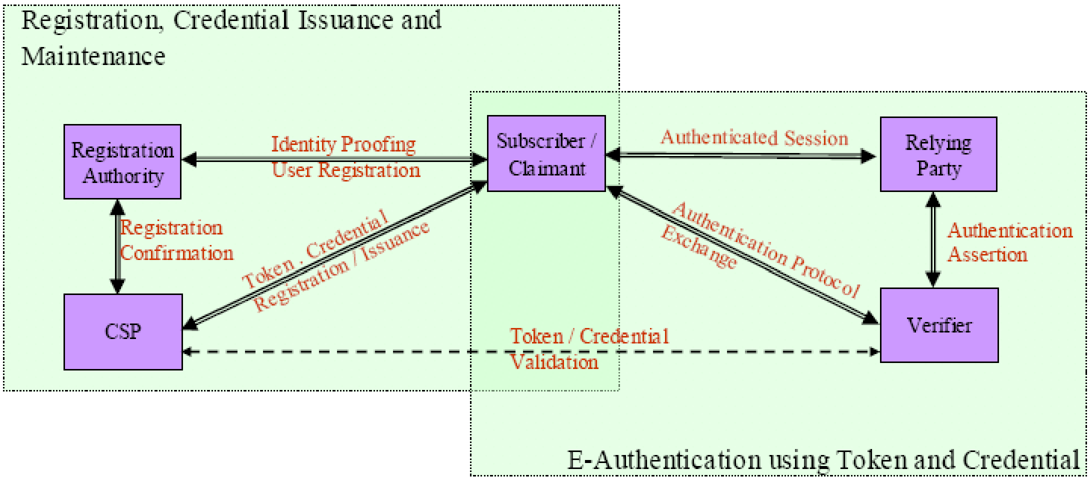


Figure 1.9 - Authentication Entities [10]

1.4 The Generalized User Authentication Process

The various entities and their interactions illustrated in Figure 1.9 cause rise to what will be referred to throughout this thesis as the Generalized User Authentication Process. This so-called Generalized User Authentication Process (GUAP) is illustrated in Figure 1.10.

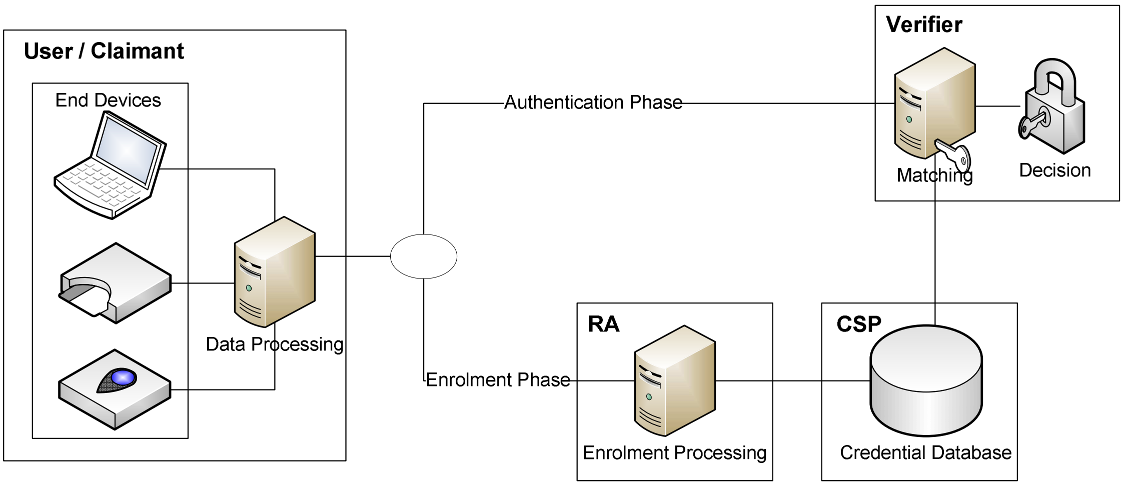


Figure 1.10 - Generalized User Authentication Process

In Figure 1.10 the User/Claimant entity is represented by two devices: the End Devices and a Data Processing Unit (DPU). End devices are any number of devices that together form the ability to accept the <credential, token> pair from a user or claimant. After an end device receives a <credential, token> pair the pair is forwarded to the DPU where they are converted to a format that is understandable to the other entities in the GUAP. The processing can be simple such as concatenating characters inputted from the keyboard into a string in the case of a username and password or more complex such as extracting identifying characteristics from a fingerprint.

During the Enrolment Phase information about the User is inputted through the End Devices and DPU before being forwarded to the RA for Enrolment Processing; in some systems the User may even select his or her own <credential, token> pair. The RA will then process the information and create the record that will serve as a matching template for the <credential, token> pair and will then forward that record to the CSP who stores it in a database.

During the Authentication Phase the Claimant will use the End Devices to input a <credential, token> pair which will then be processed and forwarded to the Verifier. The Verifier consists of a matching device which accepts the Claimant's <credential, token> pair, extracts the credential and then queries to the CSP for the template token corresponding to that credential. Once the matching device receives the template token it will compare the Claimant's token to the template token and a matching score is calculated and forwarded to the decision making component of the Verifier.

Depending on the type of authentication factors and how they are implemented in the token the matching between the submitted token and the template token could be a simple yes or no decision or it could be a more complicated decision involving statistics and threshold limits. For this reason the device that makes a decision regarding the matching score is considered to be a separate component of the Verifier from the component that computes a matching score; although in several systems the two components reside within the same physical device.

It should be noted that while it was described that the User/Claimant possesses a <credential, token> pair this is not always the case. There are in fact two different approaches to matching: One-to-Many and One-to-One matching.

One-to-One matching is the described case, where a user of the system has been issued a <credential, token> pair and when requesting authentication submits the pair. The matching device is then able to extract the submitted credential and token and compare the submitted to token to the template token in the database corresponding to the submitted credential.

One-to-Many matching is slightly different where instead of a user possessing a <credential, token> pair the user only posses a token. A claimant will submit the token and the matching device will search the entire database for a template token matching the submitted token; if a match is found the claimant will be authenticated as having the identity corresponding to the matching template token.

It should be noted that in One-to-One matching the Claimant identifies his or her self and the system authenticates that claim; while in One-to-Many matching the system performs both the identification and the authentication. As a result the One-to-Many scheme should only be used in systems where the uniqueness of the token can be guaranteed.

1.5 Tokens

It is important to understand exactly what the token in the <credential, token> pair is and a suitable definition is provided by National Institute of Standards and Technology (NIST): "Tokens generally are something the Claimant possesses and controls that may be used to authenticate the Claimant's identity." [10] User Authentication is based on the assumption that if a claimant is able to prove the possession of a token associated with an identity the claimant is in possession of that identity; as a result the token must be something secret and/or unique that can allow possession of the token to reside only with the correct user and not allow an impostor to be falsely authenticated as a legitimate User.

The requirement for the secrecy and/or uniqueness of authentication tokens led to the recognition of what is considered to be the three different factors of authentication:

1. something you know
2. something you have
3. something you are

In the first factor, "something you know," there exists some secret information that is shared between the user and the authenticating entity that is established during the Enrolment Phase. A very common example of the <credential, token> pair in this scheme is a <user id, password> pair. During the Authentication Phase a claimant will provide the <credential, token> pair and verification consists of checking a match between the submitted token and the template token. A system using this type of factor usually will require an exact match between the two tokens to allow authentication to occur.

Authentication in this system is based on the assumption that only the correct user is aware of the shared information.

Authenticating a user based on information shared between the system and the user is considered to be the weakest form of authentication. This is largely due to the fact that it is difficult for the system to ensure the information is only in the possession of the user. While it is possible for the system to ensure the protection of the secret information and prevent it from becoming available to an attacker the system has no way of controlling the information once it is in a user's possession.

The second factor, "something you have," uses a unique item issued to a user for authentication. This factor is very familiar as government issued identification cards, such as a passport, birth certificate or drivers license, have been issued to most people in the world. Computer systems have expanded on this and a very common method of electronic user authentication using this factor is an identification card with a magnetic strip that allows a card reader to extract the <credential, token> pair; in this case the token is often a secret cryptographic key that was placed onto the card along with the credential during the Enrolment Phase. Through possession of the physical identifying object the system assumes that a claimant possesses the claimed identity.

Authenticating a user based on something they have is considered stronger than authenticating based on something they know since the system has control over the issuance of the identifying possessions. The previous problem of a user being able to share their authenticating information is eliminated using this measure, however, the system is still not authenticating the individual, merely the item they posses.

The third factor, "something you are," is considered the strongest authentication factor since the factor actually seeks to determine identifying physical characteristics that make the individual unique. The previous two measures did not authenticate the individual: the first authenticated secret information and assumed since only the individual knew the information the claimant must be the individual; while the second method authenticated the token issued to the individual and assumed that since the token had been issued to the individual the claimant must be the individual. Authentication schemes using the third measure are known as biometric authentication systems; biometric being a word derived from the Greek bios, meaning life and metric meaning measure.

It is possible for an authentication system to use any combination of the three factors. If an authentication system incorporates two of the factors, for example "something you are" and "something you know", it is known as a two-factor authentication system and similarly if it involves all three types it is called a three-factor authentication system. Generally speaking the more factors an authentication system incorporates the more secure it is. It should be noted however that adding additional levels of the same factors (ex. requiring two passwords) is still considered to be a single-factor authentication system and not considered as secure as a two-factor authentication system (ex. requiring one password and one smart card)

1.5.1 Token Attributes

A token, based on some combination of the three authentication factors, possess attributes that are used in the matching phase to authenticate a claimant. The attributes are some measurable quality that is extracted from the submitted token and compared to the template token for a matching score. A password-based token, for example, has the attribute that it is a unique string of characters: the submitted password is compared against the template password and if they are a perfect match the claimant is authenticated as being in possession of the claimed identity. The attributes that comprise a token fall into two categories: static or dynamic.

Static attributes of a token do not change between authentication sessions and always require a perfect match to the template token stored in the CSP. Some examples of tokens consisting entirely of static attributes are:

- **Passwords:** A secret passphrase that is shared between the user and the system consisting of a unique string of characters that is established during enrolment. The string is unchanged during the authentication phase and the submitted string must match the template string perfectly.
- **Magnetic Key Card:** A unique identifying binary key is embedded on the card when issued to the user at enrolment; the key is also known to the system and stored in the CSP. At authentication the key is read from the card and compared to the template key, if a perfect match is found the claimant is authenticated.
- **Fingerprints:** At enrolment a user's fingerprints are recorded into the CSP. At authentication a claimant will make an identity claim and press a finger or fingers to a fingerprint scanner. The image collected from the scanner is compared to the template image and if a perfect match is found the claimant is authenticated.

The unchanging property of static attributes adds the ability for the system to support both identification and verification. Static attributes provide a measure that can be considered unique to a User.

Dynamic attributes of a token can change between authentication sessions and do not require a perfect match to a template token stored in the CSP. Some examples of tokens consisting of dynamic attributes where the submitted token will not match the template token are:

- **Voice Samples:** Due to the nature of human speech no two voice samples of the same spoken word and from the same person are exactly the same. Measurable qualities of the voice are the same however and these measurable qualities are used in voice authentication.
- **Signatures:** Handwritten signatures by the same person are never completely identical; however, measurable qualities are discernable.

In these examples the dynamic attributes of a token help to validate a submitted token: if a token is submitted that is completely identical to the template token or a previously submitted token then with extremely high probability it can be concluded that the token is being fraudulently submitted by an attacker.

Sometimes a token could consist of dynamic attributes that change between authentication sessions but still match the template token. An example of this is a smart-card where an embedded processor constantly changes the key that will be read from it in the authentication phase; the template token in this situation is software that determines what the valid key at the time of authentication is.

Strong tokens contain both static and dynamic attributes allowing them to support both identification and authentication of the claimant. The static attributes are important to provide identification support as well as verification support. Dynamic attributes provide additional verification support by ensuring that a previously submitted <credential, token> pair that was captured by an attacker can not be used to successfully pass authentication.

1.6 Passwords

The most commonly used authentication factor used in tokens is the "something you know" factor, and manifested in what is known as a password: a secret shared between the User and the authentication system. [11] Passwords have been used as authentication tokens for years and most people are familiar in their use and have accepted them as normal authentication mechanisms.

Passwords-based tokens consist solely of static attributes. Either the password itself or an algorithmic transform, such as a cryptographic hash, of the password is stored in the CSP to act as the template token. At authentication the Claimant provides the password and if necessary the transform is calculated and compared to the template. If the provided password perfectly matches the one stored in the CSP a passing match score is returned.

There are differing types of passwords and rules associated with them. In a normal computing system where the User uses a standard keyboard to enter information a password can be a sequence of any combination of numbers, letters or symbols. In a system where the User only has a number pad to enter information, such as at a bank machine or over the phone, the password can only consist of numbers: this number-based password is often known as a Personal Identification Number (PIN).

Inherent to the idea of "something you know" authentication and thus passwords is that the authentication is based on the Claimant's knowledge of the information. In the case of passwords the information is a secret that is shared between the User and the system resulting in two common difficulties: the User forgetting their password and the purposeful or accidental disclosure of the password.

A forgotten password is a common problem in password-based authentication systems. As a result many of these systems implement some policy of handling forgotten passwords and allow some form of password reset. A password reset is usually accomplished through a separate means of authentication, such as e-mailing a new password to a registered address or through knowledge based authentication.

Disclosure of a password represents a complete break in the authentication mechanism. Disclosure violates the assumption that only the User and the CSP know that password. Passwords can be disclosed in a variety of ways, including but not limited to:

- The user voluntarily giving their password to an acquaintance
- An Attacker accesses the <credential, password> database

- A user, afraid of forgetting their password, writes it on a piece of paper which falls into the hands of an Attacker

While the idea of passwords is simple, straightforward easily and passwords widely deployed authentication mechanisms there are many attacks and weaknesses associated with passwords. These attacks and weaknesses will be explored in Chapter 2.

1.7 Knowledge Based Authentication

Knowledge Based Authentication (KBA) is the process of verifying an identity based on information provided by a claimant. During the KBA Enrolment Phase a collection of information is provided by the User to the RA and then saved in a reference database. During the Authentication Phase the claimant is asked some, or all, of the same questions asked during the Enrolment Phase; acceptance of the claimed identity depends upon satisfactory consistency between the reference data and the answers provided by the claimant.

The generic model of KBA is illustrated in Figure 1.11. In this model the Claimant, who possesses specific knowledge, provides the Service Provider with the information specific to the Service Provider's verification of the Claimant's identity such as account information and provides the Verifier with the information needed for the Verifier's evaluation of the Claimant's identity.

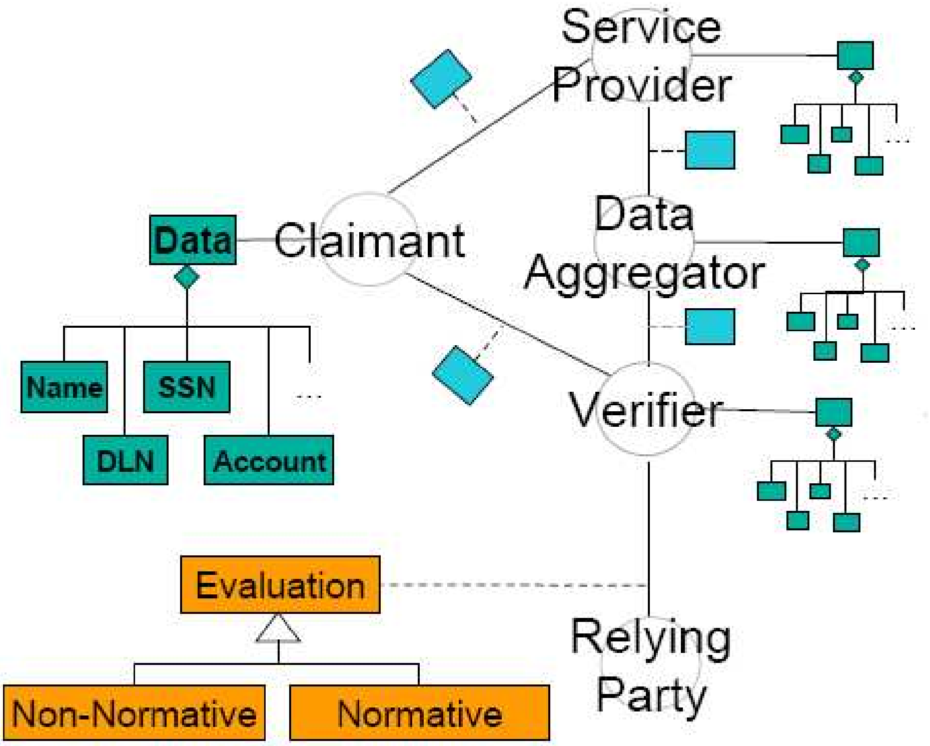


Figure 1.11 - Generic Ideal KBA Model [12]

The generic model illustrated in Figure 1.11 is the "ideal" generic model where there is no specific prior relationship between the Claimant and the Verifier. In practice, however, this ideal model is rarely the case as often the Service Provider assumes the role of the Verifier and the most commonly deployed KBA systems are similar to the one illustrated in Figure 1.12.

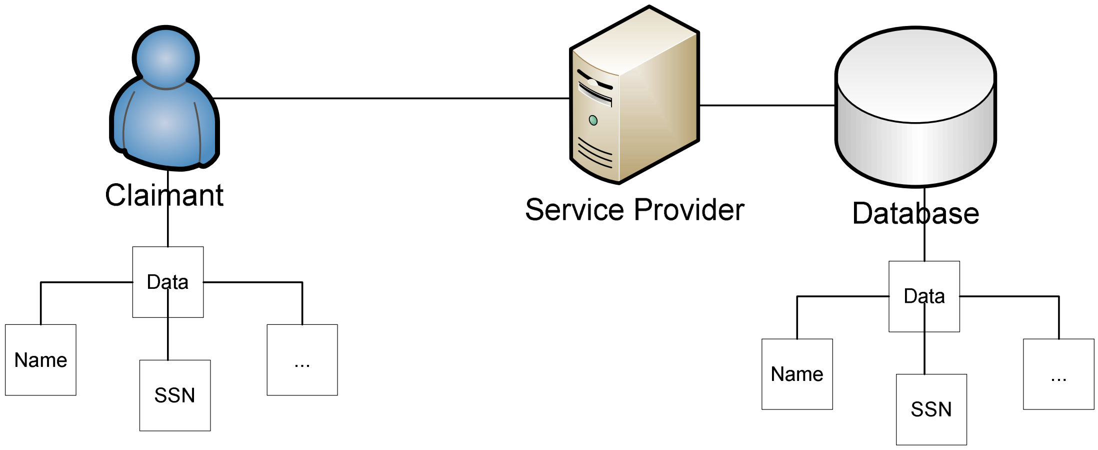


Figure 1.12 - Deployed Generic KBA Model

Unlike most authentication systems based on the "something you know" authentication factor, where a secret is often shared between the User and the CSP, the information used in KBA is not necessarily secret; often the information is personal but not secret. Common examples of questions posed in KBA include:

- What street did you grow up on?
- What city where you born in?
- What is your mother's maiden name?
- What is your first pet's name?
- What was the make of your first car?

The list of potential questions is endless as the list is simply composed of questions about facts in the personal life of the user.

A single question in a KBA scheme is known as one factoid. Since the information used in KBA is not secret the security of KBA is based on the ability to add several factoids to the process. This means that a KBA scheme must seek to have a number of questions whose answers can be considered independent from each other. The basic premise behind KBA is that the more details that the Claimant knows about the personal life of the owner of the claimed identity the more likely the Claimant possess that identity.

The previously mentioned independence requirement arises from the idea that for additional factoids to have an integer multiplicative increase on the level of authentication the factor must be completely independent from other factoids. However, in the case of KBA it is impossible for the answers to the questions to be truly independent from each other sine the answers all relate to the personal life of the User. As a result KBA questions can be considered to be independent if the answer to one question can not lead to an answer to another question.

KBA is an authentication system that seeks to address a common difficulty in the "something you know" authentication employed systems which is the User forgetting some or all of their <credential, token> pair, such as a forgotten user ID and/or password. The forgotten credentials problem is addressed through the idea that a user would not forget certain personal facts about themselves and authentication is no longer based on knowledge of an arbitrarily selected shared secret but on facts about the User's life.

1.8 Biometrics

Biometrics is the "automated approach to authenticate the identity of a person using the individual's unique physiological or behavioural characteristics such as fingerprint, face, voice, signature, etc." [13] Of the three authentication factors "a biometric is the most secure and convenient authentication tool. It can't be borrowed, stolen, or forgotten, and forging one is practically impossible." [14] A biometric measures an individual's "unique physical or behavioural characteristic to recognize or authenticate their identity. Common physical biometrics includes fingerprints; hand or palm geometry; and retina, iris or facial characteristics. Behavioural characteristics include signature, voice (which also has a physical component), keystroke pattern, and gait." [14]

There are many different biometrics that can be used to identify a person, for example: Facial Recognition, Fingerprints, Hand and Finger Geometry, Iris Scans, Retinal Scans, and Voice Recognition. While each of these technologies measures a different biometric there is a general process for feature extraction, seen in Figure 1.13 that is common to all of them.

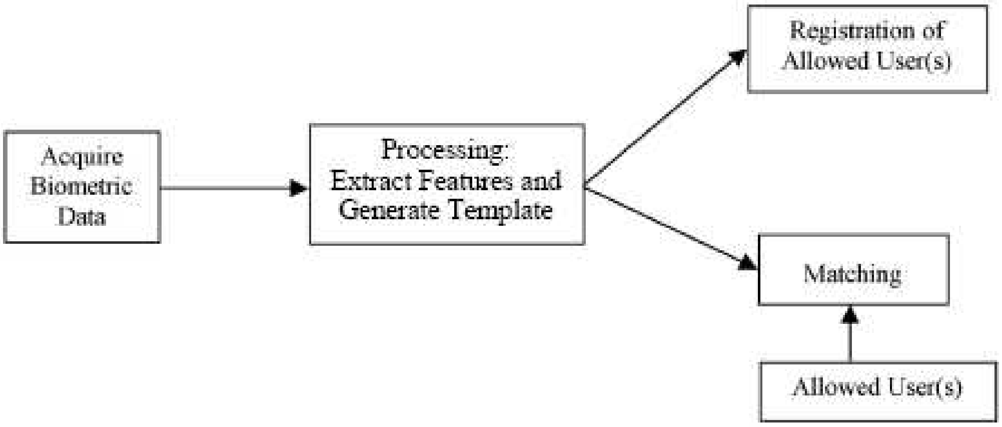

Acquire Biometric Data
Processing: Extract Features and Generate Template
Registration of Allowed User(s)
Matching
Allowed User(s)

Figure 1.13 - Common Biometric Process Flow [13]

In Figure 1.13 the first step is to "Acquire Biometric Data" this is done through a sensor such as a camera or a microphone. Once the biometric data is acquired it is then sent for processing. The processing generally consists of enhancing the data and removing noise and then extracting the unique features of the biometric data sample.

In the Enrolment Phase the unique features are extracted from the biometric data and a template is generated. The template is then associated with the identity of a user and stored in a secure database. By extracting the unique features out of the biometric sample the template forms the basis for future comparison when a claimant submits a biometric sample for authentication during the authentication phase.

As previously mentioned there are several different biometrics that can be used to identify an individual. Since there is such an extensive list of biometric technologies available a brief overview of several of the most commonly used is presented in the following subsections.

1.8.1 Facial Recognition

For humans facial recognition is the most common biometric; we identify and recognize people by looking at their faces every day. As such it is only natural that a technology be developed to allow a computer to authenticate a person by their facial features. There are two main approaches used to perform face recognition, namely holistic or global approach, and feature-based approach. [13]

Facial recognition using a feature-based approach relies on the identification of certain points on the face that are less susceptible to alteration such as the points at the eyes, the side of the nose and the mouth and the points surrounding the cheekbones. A facial recognition system will acquire an image of the person's face through a camera and then compute geometric relationships between the points on the person's face. It should be noted that since detection of the feature points occurs before the analysis the system is capable of adapting to position variations in the image, however, automatic detection of the points is not accurate and consistent enough to yield a high accuracy rate for the face recognition. [13]

The holistic approach differs from the feature-based approach by processing the entire face simultaneously without attempting to localize the individual points. By processing the whole face simultaneously the holistic approach does not lose information by only processing certain points as the feature-based approach does. The holistic approach thus generally "yields more accurate recognition results, however, such a technique is sensitive to variations in position and scale." [13]

1.8.2 Fingerprint

Fingerprint identification is a very common, well understood and accepted biometric. The sample fingerprint seen in Figure 1.14 illustrates the main features of a fingerprint and how its unique features are used to identify an individual.

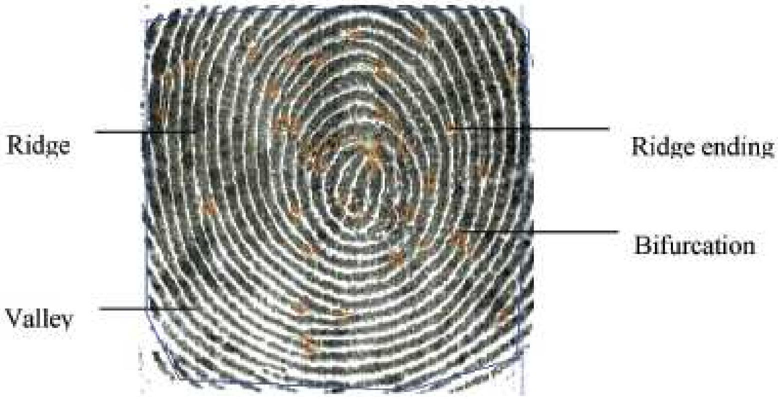


Figure 1.14 - Sample Fingerprint [13]

Every fingertip has a unique pattern of ridges that give rise to feature points, called minutiae, which are used to identify it. There are "two main technical approaches to fingerprint recognition: minutia matching and pattern matching." [13] Minutia matching locates the minutia in the fingerprint and compares their geometric information, type, direction and relationship to those of the template fingerprint's minutia to determine a match or not. Pattern matching extracts data from the regions surrounding the minutia and matching is performed by trying to find the same area in the template.

Fingerprint recognition can generally "achieve good accuracy sufficient for both verification and identification." [13] Due to its familiarity among the population it is also a generally accepted biometric and is viewed as being relatively unobtrusive and easy to use. However, the sensors can not always recognize a fingerprint, for example the captured image of the fingerprint is unusable from people with very wet or very dry skin. Additionally people tend to injure their finger frequently (for example a cut on the tip) and thus leading to a rejection during the authentication process.

1.8.3 Hand Geometry

In hand geometry authentication a user places their hand on a large surface and a camera which is under the surface looking up acquires an image of the hand. A side view image of the hand is also acquired. The two images are then processed measuring the length, width, thickness and curvatures of the fingers and hand and their relative geometry (Figure 1.15). The processed data is then compared to the template data and if a match occurs the User is authenticated.

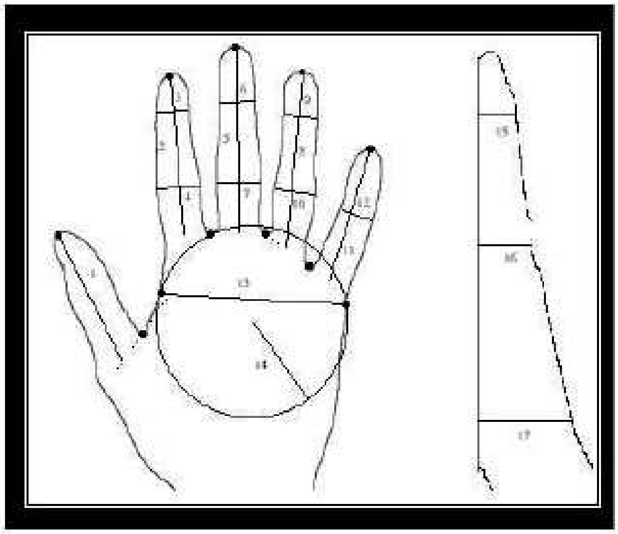


Figure 1.15 - Hand and Finger Geometry Measurements [13]

The drawback in the system is that the equipment can be rather bulky and expensive. Additionally health problems such as arthritis and aging may cause problems with the image.

1.8.4 Iris

The coloured part of the eye, known as the iris, is composed of a tissue that gives the appearance of layered radial lines or mesh when examined closely. The visible mesh consists of characteristics that are unique to an individual eye and is stable throughout an individual's lifetime. An image of the iris is acquired "using a monochrome camera with visible and near infra red light (700-900nm)." [13] The processing stage then extracts features from the image and is able to "reveal 266 independent degrees-of-freedom of textural variation, making it a very accurate biometric." [13]

Iris recognition is a very accurate biometric available. It is fast, efficient and a template requires very little storage space. However, the equipment used to scan an eye is very expensive, bulky and not readily available to the average user. Additionally, an iris scan is a very invasive and time consuming method of authentication.

1.8.5 Retinal Scans

Whereas an iris scan uses the unique patterns of a person's iris for identification, a retinal scan uses the unique patterns of a person's retina. Like the iris the retina is stable from birth to death. The patterns in the retina are the unique arrangements of blood vessels at the back of the eye, a scan "involves a low intensity light source and coupler that are used to read the blood vessel patterns." [13]

Retinal Scans are the most accurate biometric measure available. [13] However, like the iris scan, they require sophisticated and expensive equipment that is not readily available. A retinal scan is also a very invasive and time consuming method of authentication.

1.8.6 Voice Biometrics

Next to facial recognition voice recognition is the most natural and common form of recognition for humans. Almost daily humans practice voice recognition when they receive a telephone call and recognize the voice of the caller on the other end; this recognition is possible because of unique features contained within a person's voice such as timbre, frequency, and rate of speech. "At the primary level, speech conveys a message via words, but at other levels speech conveys information about the language being spoken and the emotion, gender and generally the identity of the speaker." [15] An effective voice biometric system will receive a speech signal from a Claimant and then extract, characterize and recognize the various levels of information contained in the signal and use this information to authenticate the Claimant's identity.

Voice biometrics differs from other biometrics by combining both static (physical trait) and dynamic (behavioural trait) attributes into a single biometric template. This combination occurs because voice is a combination of the "anatomical structures of the vocal tract and learnt behaviour – the habitual way we speak." [16] Additionally voice biometrics can combine both biometric (something you are – your physical voice attributes) and personal information (something you know such as a password – the spoken message) into a single credential.

There are two fundamental approaches to implementing a voice biometric system: speaker identification and speaker verification. In speaker identification the Claimant only presents a voice sample for verification, making no identity claim, and the system searches the template database for a match and if found identifies the Claimant. In speaker verification the Claimant provides an <identity, voice sample> pair and the system compares the voice sample to the claimed identity. In general, most compelling applications of speaker recognition technology use speaker verification. [15]

In these two approaches the speech used can range from pure text-dependent to text-independent. In a pure text-dependent system the User will recite a pre-determined message, that the system has knowledge of, during the enrolment and verification phases. A less pure text-independent system, known as a text constrained system, is one in which the User can only recite text from a limited vocabulary, the system has knowledge of the constrained vocabulary; an example of this is prompting the User to recite a random text from one of five possible pre-determined texts. In both text-dependent and text constrained systems “it is expected that the User will cooperatively speak the fixed text or words from the prescribed vocabulary. The prior knowledge and constraint of the text can greatly boost performance of a recognition system.” [15] In a text-independent system the User will be able to speak whatever text they desire as the system has no knowledge of the text that is to be spoken. A voice recognition in a text-independent system is much more difficult but more flexible. “Of the two basic tasks, text-dependent speaker verification is currently the most commercially viable and useful technology, although there has been much research conducted on both tasks.” [15]

1.8.6.1 Operation of a Voice Biometric Authentication System

As was previously discussed an authentication system has two phases of operation, the Enrolment Phase and the Authentication Phase, this is no different for a biometric system. Figure 1.16 is an illustration of the operation involved in the two phases of a voice biometric system.

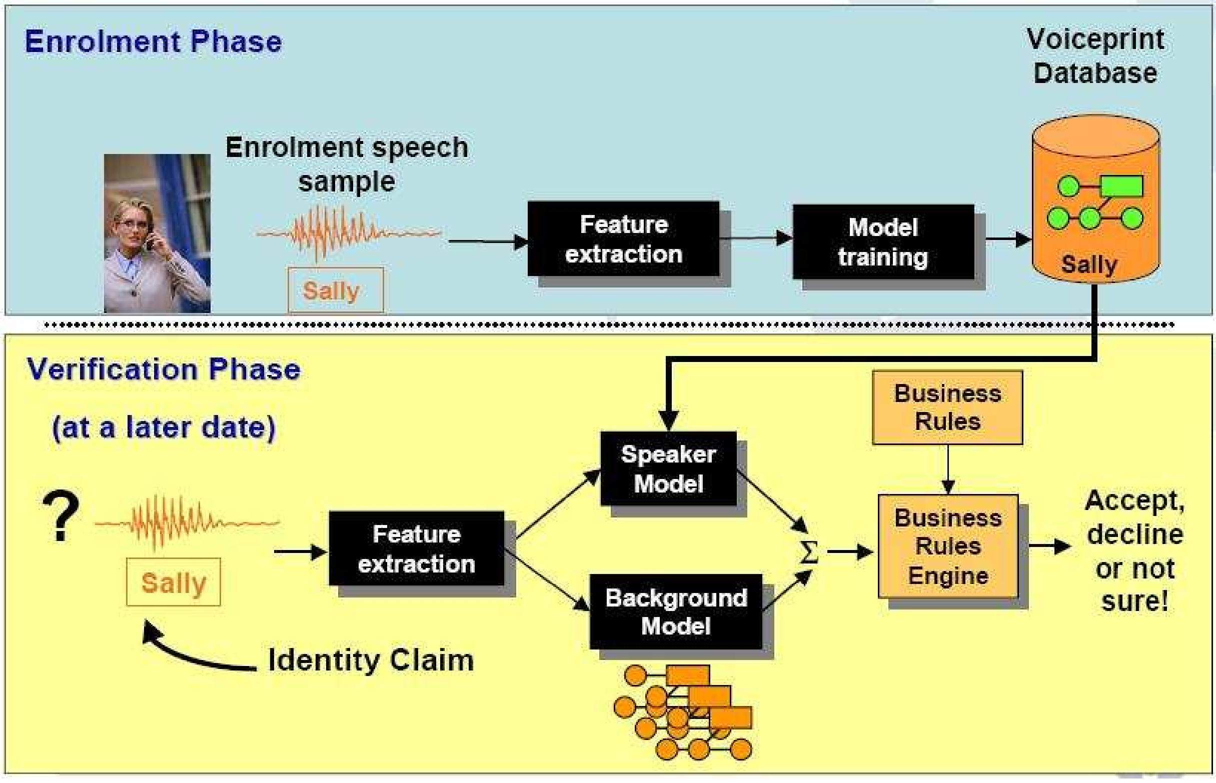


Figure 1.16 - How a voice biometric system works [16]

In the Enrolment Phase, illustrated in Figure 1.16, a user will provide a speech sample to the system which will then create a model of the User’s voice and add it to the voiceprint database. This will allow the system to authenticate the User at a later date.

In the Verification Phase the Claimant makes an identity claim and provides a speech sample which the system will then compare against the recorded voiceprint in the database and apply the appropriate rules and makes a decision. The Verification Phase, illustrated in Figure 1.16, is an instance of speaker verification since the Claimant (Sally) has identified herself through her identity claim when she provided a speech sample.

The Verification Phase differs slightly between the two different approaches to a voice biometric system. The basic structure of a speaker identification system can be seen in Figure 1.17(a) while the basic structure of a speaker verification system can be seen in Figure 1.17(b). In both systems the features are first extracted from the voice sample through the front-end processing.

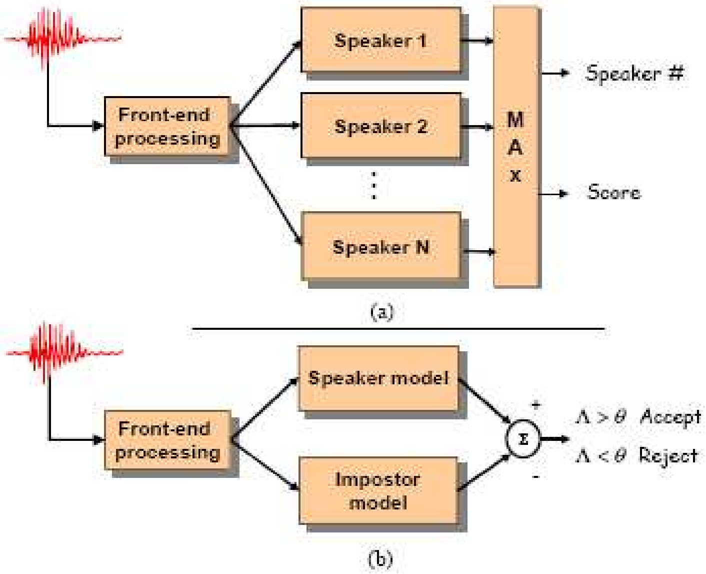
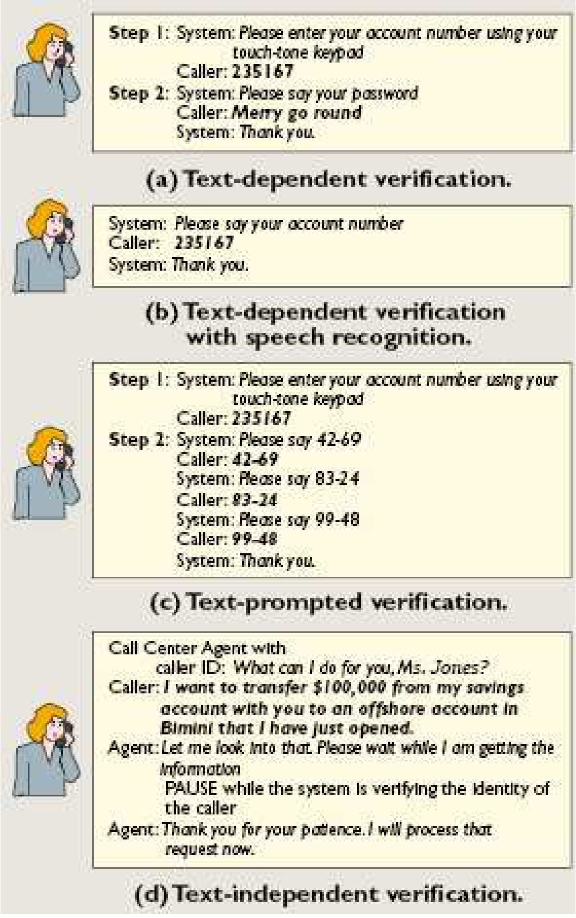


Figure 1.17 - Basic structure of (a) speaker identification and (b) speaker verification systems [15]

For a speaker identification system, like the one illustrated in Figure 1.17(a), the speaker identity that best matches the sample information is selected as the identified speaker through a maximum likelihood classifier.

In the speaker identification system of Figure 1.17(b) after the voice sample is processed the information is then compared to all the voice templates in the database that had been previously determined during the Enrolment Phase. It is essentially "a likelihood ratio test to distinguish between two hypotheses: the test speech comes from the claimed speaker or from an imposter." [15] The likelihood ratio statistic (Λ) between the speaker and imposter models is then compared to a threshold value (θ) to decide whether to accept or reject the speaker.

Figure 1.18 shows several different ways of interacting with a speaker verification system. Figure 1.18 (a) and (b) are both text dependent systems which strengthens the authentication process by incorporating two different classes of security (something you know and something you are).



Step 1: System: *Please enter your account number using your touch-tone keypad*
Caller: **235167**
Step 2: System: *Please say your password*
Caller: ***Merry go round***
System: *Thank you.*

(a) Text-dependent verification.

System: *Please say your account number*
Caller: ***235167***
System: *Thank you.*

(b) Text-dependent verification with speech recognition.

Step 1: System: *Please enter your account number using your touch-tone keypad*
Caller: ***235167***
Step 2: System: *Please say 42-69*
Caller: ***42-69***
System: *Please say 83-24*
Caller: ***83-24***
System: *Please say 99-48*
Caller: ***99-48***
System: *Thank you.*

(c) Text-prompted verification.

Call Center Agent with caller ID: *What can I do for you, Ms. Jones?*
Caller: ***I want to transfer $100,000 from my savings account with you to an offshore account in Bimini that I have just opened.***
Agent: *Let me look into that. Please wait while I am getting the information*
PAUSE while the system is verifying the identity of the caller
Agent: *Thank you for your patience. I will process that request now.*

(d) Text-independent verification.

Figure 1.18 - Types of speaker verification [17]

Figure 1.18 (c) is a text-prompted voice authentication system. In a text-prompted system the system asks the User to speak a series of randomly selected numbers or phrases.

Figure 1.18 (d) is an example of a text independent verification system. In a text independent verification system it is possible to design an unobtrusive verification system by verifying the User's voice regardless of what they are saying. It is even possible in a text independent system to authenticate a user without them even knowing they are being authenticated. This unique ability of the text independent system makes it very attractive for customer related applications as it does not distract the User from their primary objective by forcing them to go through rigorous security checks.

1.9 Accuracy of Biometrics

One of the most obvious requirements of a biometric system is that it needs to be accurate; meaning that the system does not falsely authenticate an imposter and grant them access (known as a False Accept) or deny a legitimate user access to the system (known as a False Reject). To evaluate the accuracy of a biometric system there are several important criteria to consider:

- **False Accept Rate (FAR):** This is also known as Type I error. It measures the percentage of impostors being incorrectly accepted as genuine user. Since almost all biometric systems aim to achieve identity authentication, this number should be as low as possible. [13]
- **False Rejection Rate (FRR):** Also known as Type II error, this measures the percentage of genuine users being incorrectly rejected. In order to minimize the inconveniences (or embarrassment) to the genuine user, this number should also be low. Nevertheless, in general, this error is more acceptable as usually the User can make a second attempt. [13]
- **Equal Error Rate (ERR):** FAR and FRR are related. A stringent requirement for FAR (as low as possible) will inadvertently increase the FRR. [13] The ERR is the point where the FAR equals the FRR as is illustrated in Figure 1.19. A small ERR indicates a good balance in the sensitivity of the system.
- **Failure to Enrol (FTE):** The FTE is the percentage of people who cannot register themselves on the system. Biometric systems have certain physical requirements that some users simply cannot meet, for example, someone with severely scarred fingertips cannot register in a fingerprint biometric system as a suitable model cannot be created for the User.

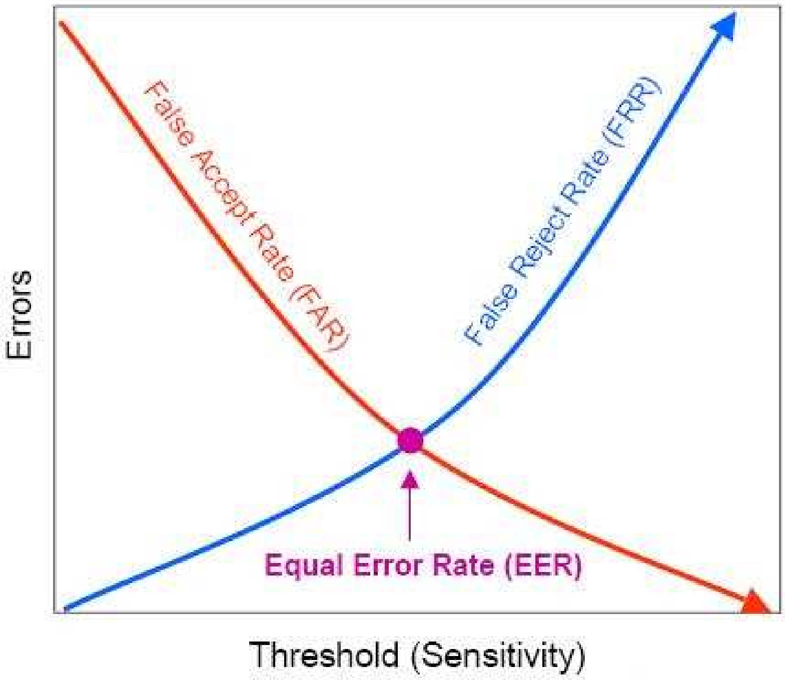
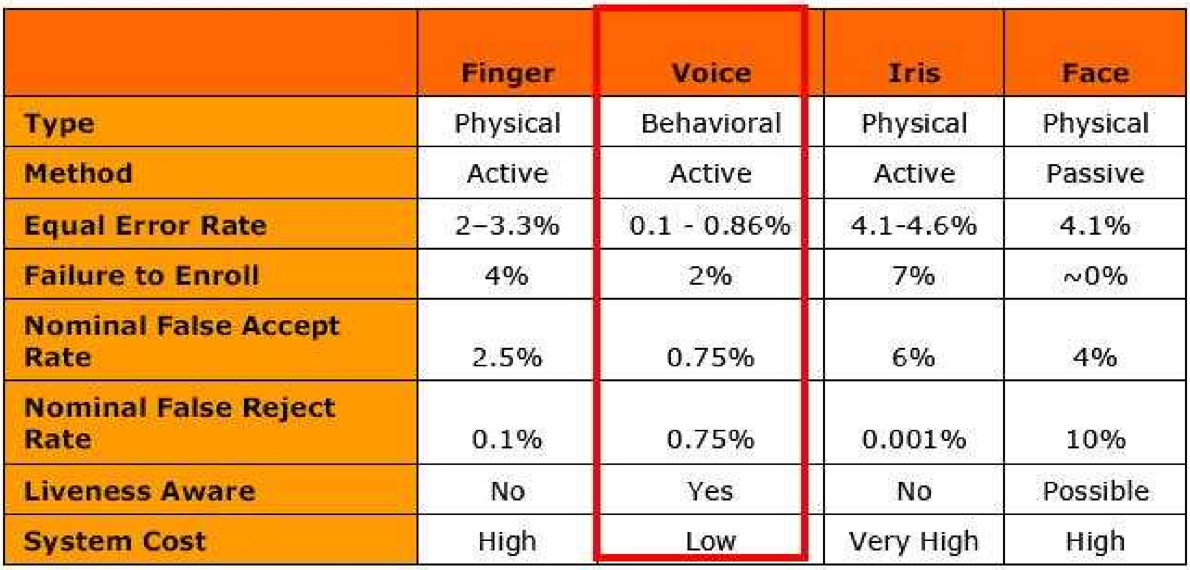


Figure 1.19 - Equal Error Rate [18]

Figure 1.20 is a table containing measurements of interest when evaluating biometric systems of some common biometrics: Fingerprint scan, voice recognition, Iris scan and face recognition. It can be observed in Figure 1.20 that the FRR of voice recognition falls third in the while the FAR is the much lower than any of the other biometrics.



	Finger	Voice	Iris	Face
Type	Physical	Behavioral	Physical	Physical
Method	Active	Active	Active	Passive
Equal Error Rate	2–3.3%	0.1 - 0.86%	4.1-4.6%	4.1%
Failure to Enroll	4%	2%	7%	~0%
Nominal False Accept Rate	2.5%	0.75%	6%	4%
Nominal False Reject Rate	0.1%	0.75%	0.001%	10%
Liveness Aware	No	Yes	No	Possible
System Cost	High	Low	Very High	High

Figure 1.20 - Biometric Comparison Table [18]

Further examination of Figure 1.20 will reveal that while voice recognition has a very low FTE rate, the FTE rate of face recognition is quite a bit lower and is in fact approximately zero. However, as was previously discussed, the true effectiveness is measured by the EER and it can be observed in Figure 1.20 voice recognition has an EER that is much lower than the EER of any of the other biometrics.

An investigation [13] into biometrics created the table seen in Figure 1.21 which is a comparison of many different biometric technologies. The following list of different aspects was measured in this investigation:

- Universality - how common the biometric is found in each person
- Uniqueness - how well the biometric separates one person from another
- Permanence - how well the biometric resists aging
- Collectability - how easy it is to measure the biometric
- Performance - its accuracy, speed and robustness
- Acceptability - willingness of the public to use it
- Circumvention - level of difficulty involved in a successful attack

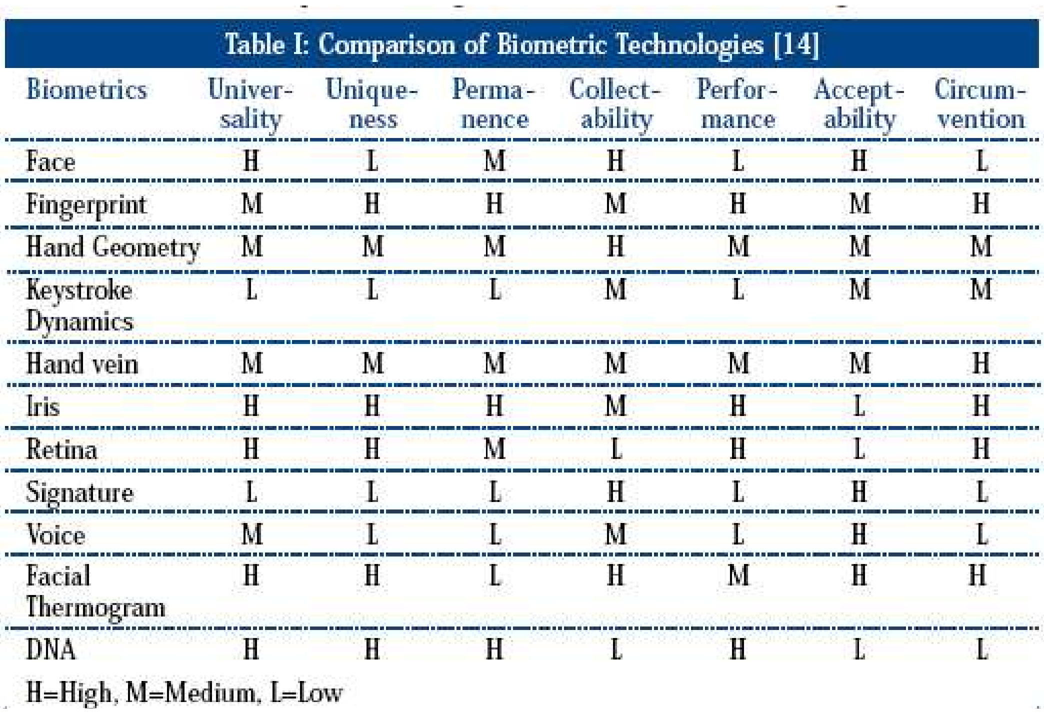


Table I: Comparison of Biometric Technologies [14]

Biometrics	Univer-sality	Unique-ness	Perma-nence	Collect-ability	Perfor-mance	Accept-ability	Circum-vention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retina	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

H=High, M=Medium, L=Low

Figure 1.21 - Biometric Comparison Table [13]

1.9.1 Accuracy of Voice Biometrics

Figure 1.22 contains the results of an investigation in [15] and depicts a detection error trade-off (DET) plot which shows the trade-off between false rejects and false accepts as the decision threshold changes in a speaker verification system. In this investigation "a verification system is making a single comparison of test speech to a claimed speaker model, so results are not a function of speaker set size." [15] This DET shows four equal error rate points for four different verification experiments.

1. Text-dependent using combinations lock phrases (e.g., 35-41-89). Clean data recorded using a single handset over multiple sessions. Used about 3 min of training data and 2 s test data. (0.1% – 1%)
2. Text-dependent using 10 digit strings. Telephone data using multiple handsets with multiple sessions. Two strings training data and single string verification (1%-5%)
3. Text-independent using conversational speech. Telephone data using multiple handsets with multiple sessions. Two minutes training data and 30 s test data. (7%-15%)
4. Text-independent using read sentences. Very noisy radio data using multiple military radios and microphones with multiple sessions. Thirty sec training and 15 s testing. (20%-35%)

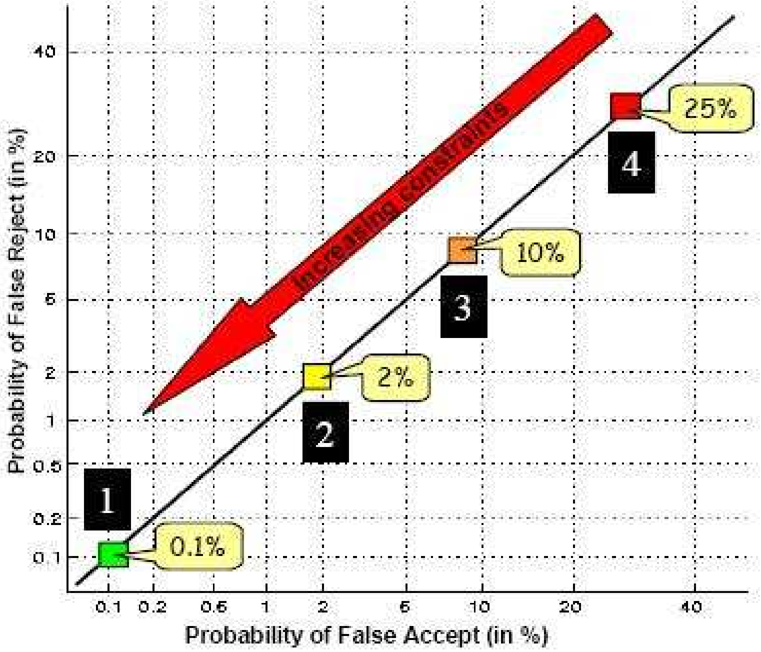


Figure 1.22 - Range of speaker verification performance [15]

It was observed in [15] from Figure 1.22 that "performance tends to improve with increasing constraints on the application (more speech, less noise, known channels, text-dependent). It was further concluded that "determining acceptable performance for a particular application will depend on the benefit of replacing any current verification procedure, the threat model (claimant to imposter attempts) and the relative costs of errors."

1.9.2 Further Benefits of Voice Biometrics

Perhaps the greatest benefit of voice biometrics is its familiarity to the Users. People are accustomed to recognizing and authenticating other people through their voice, for example when making or receiving a phone call. Among other benefits of voice biometrics [18] lists "User Friendly" as a chief benefit solely due to its "natural feeling" and that it "does not require behavioural changes;" [15] expands on this argument by saying that "speech is a natural signal to produce that is not considered threatening by users to provide. In many applications speech may be the main (or only, e.g. telephone transactions) modality, so users do not consider providing a speech sample for authentication as a separate or intrusive step."

The above observations can be evidenced by a study [19], the results of which can be seen in Figure 1.23. The study found that while the acceptance rate of biometrics was increasing (69% at the time of the study) voice recognition had one of the highest acceptance rates (84%). The primary reason that the study found for the acceptance of voice biometrics was convenience followed by faster transactions and increased security.

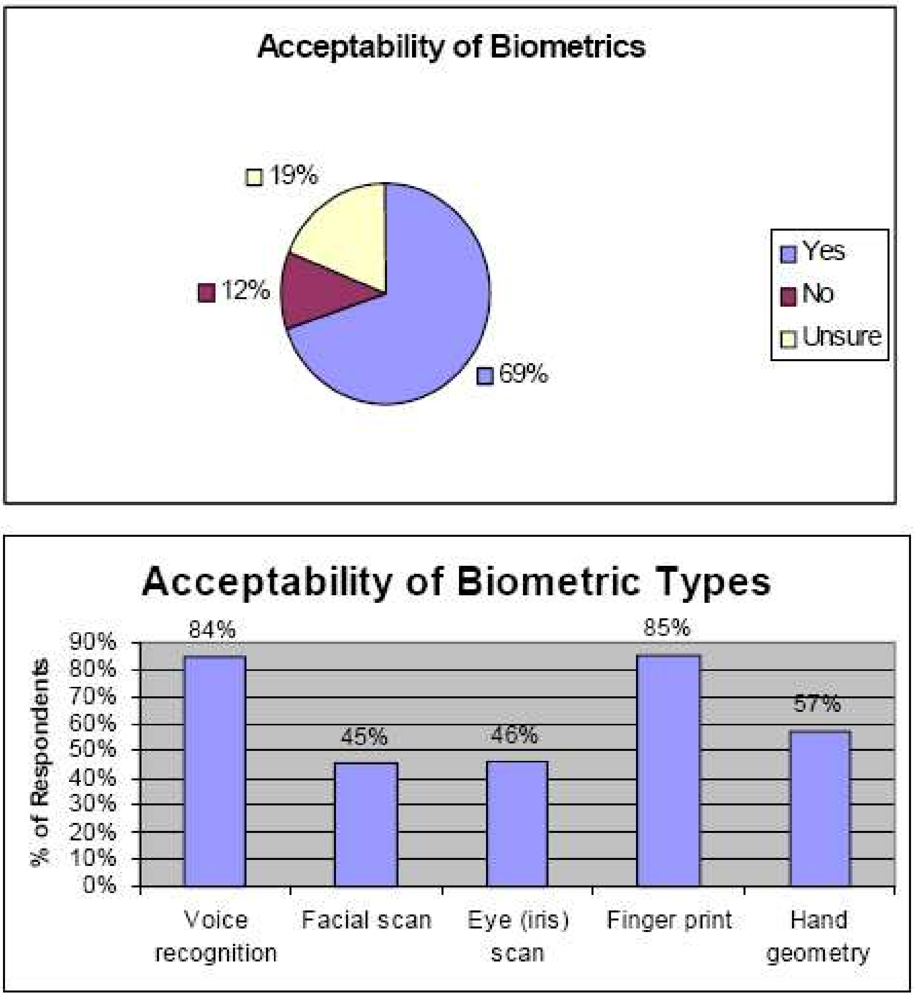


Figure 1.23 - Acceptability of Biometric Types [19]

An additional benefit to voice biometrics is that it is cost-effective to implement. Unlike other biometrics types which all require imaging equipment and/or scanners the only equipment required for a user to access a system with voice access control system in place is a relatively inexpensive microphone. Additionally, it is fairly simple to integrate a voice biometric system into existing authentication infrastructure. [18]

A voice biometric system is well suited for a phone channel. In most phone communications voice is in fact the only method of exchanging information thus making it only natural and in fact more convenient for a user to authenticate his or herself using their voice. A phone based voice biometric system: “doesn’t require specialized client hardware; it supports mobility; and it works well from any phone.” [18]

1.10 Chapter Summary

In this chapter the concept of the Generalized User Authentication Process was introduced, as well as the three different authentication factors. Additionally, three different technologies that are commonly used for authentication in privacy critical systems were discussed. In the next chapter the GUAP will be examined to highlight the points of weakness and attacks as well as attacks and weaknesses on the commonly used authentication factors and technologies.

Chapter 2
Requirements of an Authentication System

2.1 Chapter Overview

As introduced in Chapter 1, User Authentication is the process of establishing confidence in user identities presented to an information system. In privacy-critical systems this confidence in user identities is crucial to the system's operation. For instance, remote-banking depends on the ability for the bank's system to remotely authenticate a client before allowing them to access their account; otherwise anyone could access and make changes to any of the accounts at that bank.

In addition to privacy and access control user authentication is important for non-repudiation in privacy-critical systems. For example, in remote banking the bank needs to be able to prove that it was indeed their client that remotely transferred the contents of their account to a different bank account, otherwise, their client could actually do that and then claim they did not and if the bank is unable to provide a means of non-repudiation they would be held responsible for the missing funds.

In the previous chapter the concept behind user authentication and the entities involved were introduced, the factors involved explained, and the general process defined. This chapter first explores the weaknesses in the three different authentication factors before further exploring the GUAP, highlighting its weaknesses, and defining attacks on the process. After exploring and categorizing the attacks an understanding of what is required for a privacy-critical system to have confidence in the output of the authentication process will be established.

2.2 Points of Attack on the GUAP

Figure 2.1 re-illustrates the GUAP, introduced in the previous chapter, and the flow of data in the authentication process; additionally, Figure 2.1 highlights the transmission lines between the entities in the process. This section will begin with highlighting the attacks and weaknesses of the transmission lines before examining the attacks possible at each of the entities in the process.

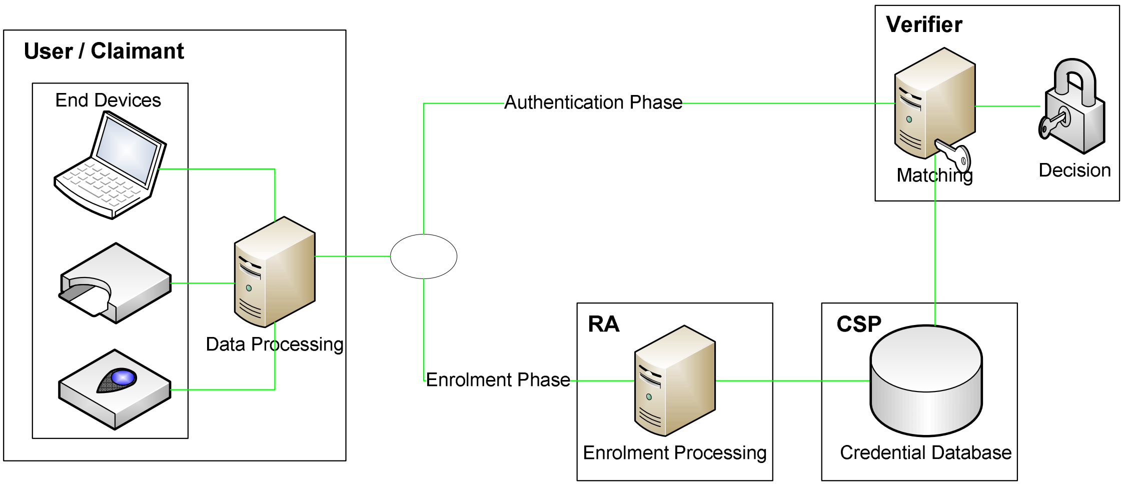


Figure 2.1 - Transmissions in the GUAP

2.2.1 Transmission-Based Attacks

At any point where data is in transit from one entity to another it is subject to a Transmission-Based attack. Transmission-Based attacks are well known and studied attacks on conventional security systems and can vary from being passive attacks, where an attacker just observes and captures data as it passes, to active attacks where the attacker manipulates the transmitted data in some manner.

2.2.1.1 Eavesdropping

Eavesdropping is a passive transmission based attack where the Attacker simply listens to the transmission line to capture the information being transmitted. If an eavesdropping Attacker happens to capture important data, such as the <username, password> pair the Attacker can masquerade as a legitimate user in a subsequent attack.

It should be noted that there are many different methods available to an Attacker to passively capture the <credential, token> pair, the easiest to understand being the one described above. Other methods vary with the type of authentication factors deployed in the system, examples include: observing a user enter a password over their shoulder; installing a keystroke capturing device on the User's system; installing an additional card reader in front of the system's legitimate card reader; and fingerprinting a fingerprint scanner for residual fingerprints.

The threat of an Eavesdropping attack can be reduced by either eliminating the transmission channel or by protecting it. A transmission channel can be eliminated by incorporating the communicating entities into the same tamper proof device. The most effective way of protecting the transmitted data is by encrypting it with a secret key that is shared between the communicating entities.

2.2.1.2 Replay Attack

A replay attack is a Transmission Attack that begins with the passive Eavesdropping attack on the communication channel. In a replay attack the Attacker passively monitors a transmission channel, such as the link between the User/Claimant and the Verifier entities and records the data as it is transmitted. Later, the Attacker will "replay" the captured data, meaning that the Attacker will pose as the legitimate user and simply send all the captured data to the Verifier and since the captured data contains legitimate credentials the Verifier will falsely authenticate the Attacker.

A very common and successful method of preventing a replay attack is to incorporate the current time into the information submitted to the Verifier; this method is known as "time-stamping." It is important to note that in order for time-stamping to be effective the time must be incorporated in such a way that an Attacker cannot capture the data as before, modify the time field, and replay the data. To prevent the Attacker from modifying the time field time-stamping often involves the time and credentials being hashed together and/or encrypted before transmission.

An additional problem with time-stamping is clock synchronization between the End Device and the Verifier. Very often the internal clocks within the two entities are not synchronized resulting in legitimate timestamps created by end devices that do not match the current time within the Verifier. As a result the Verifier implements a window that defines acceptable variation between the submitted timestamp and its own clock. If the window is designed to be too large an attacker with knowledge of the window size could be able to perform the necessary processing inside the window to successfully perform a replay attack; on the other hand if the window is too small legitimate users may be unable to pass authentication.

2.2.1.3 Session Hijacking

Session Hijacking "is intercepting and carrying on a session begun by another entity." [11] In this attack, the Attacker waits until after two entities have entered into a session and then intercepts the traffic and continues to carry on the session masquerading as the other entity.

A prime example of session hijacking is an Attacker will wait until after the Claimant submits his or her <credential, token> pair to the Verifier and intercept the response from the Verifier. At this point the Attacker can enter into a session with the Relaying Party, masquerading as an authenticated user of the system.

Methods to reduce the threat of a Session Hijacking Attack are discussed in section 2.2.1.4.

2.2.1.4 Man-in-the-Middle Attack

Another form of a hijacking attack is referred to as the Man-in-the-Middle Attack. In this situation, similar to the previously described Session Hijacking, the Attacker positions themselves in the middle between the communicating entities in a protocol. However, in the Man-in-the-Middle Attack the Attacker intercepts all transmissions between the entities, generating and forwarding their own responses to each entity.

There are several technologies that can be implemented to reduce the risk of a Man-in-the-Middle Attack. The first, and perhaps most effective, is to incorporate all components into a single tamper-proof device thus eliminating the vulnerability of data in transmission. This first method however is not always possible.

In situations where it is infeasible to incorporate the components into one device and a data transmission is necessary the data should be transferred over a secure and mutually authenticated channel. Meaning that when the two entities involved in the data transmission establish the connection they mutually authenticate each other through the use of public-key certificates and a challenge response mechanism. Additionally, after mutually authenticating each other the two entities agree upon a shared secret session key to encrypt the data with before transmission. Having every entity involved in the session authenticate each other, establish a shared secret key and encrypting the data is an effective way of reducing the threat of both passive and active transmission based attacks.

2.2.2 Verifier Impersonation

A Verifier Impersonation attack is one where the Attacker fraudulently acts as the Verifier and a legitimate User of the system believing they are communicating with the actual Verifier voluntarily provides their <credential, token> pair to the fraudulent Verifier. The Attacker can then submit the <credential, token> pair to the actual Verifier to be falsely authenticated. Verifier Impersonation attacks generally fall into two categories: Phishing and Pharming.

In a Phishing Attack the Attacker generally sends an official looking email to a user that identifies the Attacker as the Verifier. The email often highlights some made up problem with the User's account and asks them to log in to fix the problem; the email will then contain a link to a fraudulent webpage that resembles the authentic Verifier and the User will authenticate themselves to the fraudulent Verifier using their authentic <credential, token> pair. Often at this point the fraudulent Verifier will return a server error to the User asking them to return later, and the User will continue about their day unaware that they have mistakenly given out their <credential, token> pair.

In a Pharming Attack the Attacker creates a fraudulent Verifier, much like he did in the Phishing Attack, only instead of sending emails to users he attempts to redirect users intending to go to the authentic Verifier to his fraudulent one. The most common ways of causing this redirection is by corrupting the Domain Name Service (DNS) through a technique called DNS poisoning or by corrupting the User's local routing tables. Once the User has been redirected to the fraudulent site the User, believing they are at the authentic Verifier, will begin the authentication process by providing their <credential, token> pair to the fraudulent Verifier.

The most common way of preventing Verifier Impersonation is to incorporate a two-way authentication mechanism into the authentication process. In this two-way authentication mechanism before supplying their <credential, token> pair the User ensures that they are indeed communicating with an authentic Verifier. On the web this is often done through the use of Public-Key certificates.

In some systems, such as telephone banking, the User is unable to ensure they are talking to the authentic Verifier without providing some information first. A technique that is sometimes used in this situation is for the User to provide only the credential portion of the <credential, token> pair to the system, the system will respond with the time the User was last authenticated and the User will choose whether or not to proceed based on the correctness of that information. While not ideal, this approach does provide some means of protection from Verifier Impersonation attacks to the User, however, the method is not widely deployed and in many systems it is up to the User to ensure that they have for instance dialled the correct number and are connecting to the correct Verifier without any assistance from the Verifier.

2.2.3 Attacks on the End Devices

An attack on an End Device is an attempt to exploit a weakness in the manufacture of the physical device that reads the <credential, token> pair from the Claimant. Usually these attacks focus on the Attacker attempting to capture the <credential, token> pair as it is inputted by the User. Examples include:

- an over-the-shoulder attack, where the Attacker watches the User input their <username, password> combination
- installing a keystroke logger to capture any inputted <credential, token> pairs
- installing a phony card reader on top of a legitimate card reader
- dusting a fingerprint scanner for residual fingerprints

The best way to reduce the threat of End Device Attacks is to build the components that accept inputs into a tamper-proof device. Additionally, they should be constructed in such a way that inputted <credential, token> pairs are not visible to, or obtainable, outside of the device without the exclusive permission of the device.

2.2.4 Attacks on the Data Processing Unit

The Data Processing Unit is where user inputted <credential, token> pairs are processed into a format that can be understood by the Verifier. An attack at this point consists of the Attacker gaining access to the DPU and simulating the data being submitted to the Verifier.

This type of attack, known as a Data Simulation attack, is very applicable to biometric authentication. In the case of biometrics the attack is made possible because biometric features are not secret and an Attacker can for example simulate a legitimate user's physical signature or mimic their voice to attack a system. A Data simulation attack is more likely to be successful on authentication systems that rely on behavioural characteristics, such as signature verification, than physiological characteristics, such as a fingerprint, due to the complexity of simulating a physiological trait.

The success of a Data Simulation attack relies on the ability of the Attacker to inject their simulated data into the DPU. Preventing the Attacker from injecting their simulated data can be accomplished moving the DPU onto the same device that the End Device is located on.

2.2.5 Registration Authority Attacks

An attack on the Registration Authority entity is an attempt to exploit a weakness in the RA that will allow an Attacker to create a user account with sufficient privileges to pass through access controls to accomplish their objectives. In attacks discussed thus far an Attacker was attempting to gain enough information to pass authentication as a previously registered user, even attacks at the transmission points into and out of the RA can accomplish this, however the RA attack is an attempt to create a user that shouldn't be created.

RA attacks can be something along the lines of the Attacker connecting remotely to the RA and creating an account with privileges that allows it to pass through all access controls. Another example, in a physically operated RA, the Attacker forcing an Agent of the system to create an account, or the Attacker may even be an Agent and creates a fraudulent account to allow them to mount attacks at a later time.

To prevent an Attacker from connecting to the RA and creating an account on their own the RA should not have the ability to have remote connections made to it, except through the input from the device registering a user and the internal account creation tools. It is difficult to prevent the discussed insider attack since it is necessary that the system trust its Agents, the most successful way of reducing the threat of insider attacks on the RA is requiring multiple Agents to create a single account.

2.2.6 Credential Service Provider Attacks

An attack on the CSP is an attempt to capture the any portion of the database of <credential, token> pairs, allowing the Attacker to later submit the stolen pairs to an End Device and be falsely authenticated. The success of this attack hinges on an Attacker being able to gain access to the CSP and then once having gained access being able to retrieve and decipher the contents of the database.

The success of an Attacker being able to use a <credential, token> pair from the captured database depends on the authentication factors deployed in creating the token. For instance, in the case of voice biometric authentication the token in the database is a voice template and creating a suitable voice sample to submit to the Verifier from the template requires a great deal of computing resources as well as specific knowledge about the voice verification system. However, in the case of the token being a password the Attacker can very easily submit a captured <credential, token> pair.

An attack on the CSP can be thwarted by eliminating the ability of remote connections to the CSP from being. The CSP can still maintain its functionality of responding to queries from the Verifier by presenting an interface that first establishes a mutually authenticated connection with the Verifier and then accepts a request for a single set of credentials but does not allow a remote connection to access the entire database. To improve security even further it is recommended that the Verifier and CSP are incorporated into the same tamper-proof device, ensuring that none of the <credential, token> pairs are available outside of the two entities that use them.

Enrolment and modifications of users in the CSP can be accomplished by the administrators by only allowing connections to the CSP through terminals connected only to the CSP. This will reduce the threat of an Attacker being able to remotely access the CSP and the database.

Encrypting the database is recommended to reduce the threat of an attack in the case that an Attacker is able to obtain access to and copy the database. It is important to note that in some cases encryption of the <credential, token> pairs database does not necessarily mean an Attacker must break the encryption to obtain a valid pair if he or she has a copy of the encrypted database; particularly if the token being used is a secret password: if the Attacker knows the type of encryption he or she can guess passwords and encrypt them until a match in the database is found.

2.2.7 Verifier Attacks

A successful attack on the Verifier entity involves the Attacker gaining access to the matching or decision making devices inside the Verifier. Once the Attacker has inside access to the entity he or she may record all authentication sessions in an attempt to capture <credential, token> pairs and matching scores. An Attacker may even inject an artificial match into the device allowing the Attacker to pass through the device to the decision mechanism which accepts the artificial match and falsely authenticates the Attacker as a legitimate user.

Preventing a successful attack on the Verifier is similar to preventing a successful attack on the CSP in that the Verifier should be constructed in such a way that an Attacker can not establish a connection with the entity and inject an artificial matching score or obtain information from the Verifier other than an authentication pass or fail response.

2.3 Token and Authentication Factor Weaknesses

In addition to the attacks against the GUAP explored in the previous section attacks against the authentication token itself are also possible. In the previous chapter it was explained that during the Enrolment Phase the User receives a <credential, token> pair that is later submitted for verification during the Authentication Phase and that the requirement for the secrecy and/or uniqueness of the token led to the recognition of the three different authentication factors: something you know, something you have and something you are.

If an Attacker can gain control of a token he or she may be able to masquerade as and be falsely authenticated as the owner of the token. The attacks and threat to the token are dependent on the types of authentication factors that comprise the token. Assuming that the owner of the token is not colluding with the attacker, Table 2.1 lists the threats to the tokens used in user authentication along with some examples.

Table 2.1 - Token Threats [10]

Token Threats / Attacks	**Description**	**Examples**
Theft	A token with a physical manifestation is stolen by an Attacker	• Hardware cryptographic device stolen • One-time password device stolen • Lookup token stolen • Cell phone stolen
Duplication	The Subscriber's token has been copied with or without his or her knowledge	• Passwords written on paper disclosed • Passwords stored in electronic file copied • Software PKI token (private key) copied • Lookup token copied
Eavesdropping	The token secret or authenticator is revealed to the Attacker as the Subscriber is submitting the token to send over the network	• Shoulder surfing of passwords • Keystroke logging on keyboard • PIN captured from PIN pad device • Fingerprint data captured from reader
Offline Cracking	The token is exposed using analytical methods outside the authentication mechanism	• Differential power analysis on • Stolen hardware cryptographic token • Software PKI token is subjected to dictionary attack to identify correct PIN to use the private key within token
Phishing or Pharming	The token secret or authenticator is captured by fooling the Subscriber into thinking the Attacker is a Verifier or Relying Party	• Password revealed by Subscriber to website impersonating as the Verifier • Password revealed by bank Subscriber in response to an email inquiry from a Phisher pretending to represent the bank • Password revealed by Subscriber

		at a bogus Verifier website reached through DNS re-routing
Social Engineering	The Attacker establishes a level of trust with a Subscriber in order to convince the Subscriber to reveal his or her token or token secret	• Token revealed by Subscriber to officemate asking for password on behalf of Boss • Token revealed by Subscriber in telephone inquiry from masquerading system administrator
Online guessing	The Attacker connects to the Verifier online and attempts to guess a valid token authenticator in the context of that Verifier	• Online dictionary attacks to guess passwords • Forging Claimant's handwriting based on stolen handwriting sample • Online guessing of secret token registered to legitimate Claimant

It can be observed in Table 2.1 that the most common threat to the "something you know" and "something you have" authentication factors are the theft and/or duplication of the token. This threat is the result of the authentication depending on the verification of the possession of the static information that comprises the token or the possession of the physical token itself. Verifying the only static attributes of the token allows for an attacker that can seize that information and submit it for verification to be falsely authenticated as a legitimate user.

While the attacks in Table 2.1 exploit weaknesses in the authentication factors that comprise the token, some of the attacks, such as Phishing and Pharming, can be considered attacks on the entities or transmission lines of the GUAP. The remainder of this section will examine the weaknesses of two of the most commonly deployed types of "something you know" tokens in privacy-critical systems: password or PIN-based tokens and KBA. Additionally this section will explore the weaknesses in biometric authentication.

2.3.1 Password and PIN Weaknesses

The security concept behind a password is that the password is considered to be a random string of characters. For instance if a password is a random string of characters from the set of 94 printable ASCII characters (space is not included), than the set of all possible passwords is calculated by the following formula:

$$PasswordSet = \sum_{k=n}^{m} 94^k$$

Where n and m are the minimum and maximum allowed password length respectively.

For example in a system that requires a password to be between the lengths of six and eight characters than the set of all possible passwords is:

$$94^6 + 94^7 + 94^8 \approx 2^{52}$$

However, several different studies into password security over the last thirty years of computing history have found that the assumption that a password is a random string of characters is simply not true as users tend to find it difficult to remember random strings such as *t^4M$zqg* and are more likely to use easy to remember passwords such as *cat*. Table 2.2 and Table 2.3 summarize results from two such studies that measured the statistics of what composes user selected passwords. The first study [20] found that a dictionary attack, which took five minutes to complete, produced about one third of the passwords, while the second study [21] found that approximately 20% of the passwords were found using simple dictionary and wordlist attacks.

Table 2.2 - Distribution of Passwords [20]

Characters	Count	Percent
Were a single ASCII character	15	0.5
Were strings of two ASCII characters	72	2
Were strings of three ASCII characters	464	14
Were strings of four alphanumeric characters	477	14
Were five letters, all upper-case or all lower-case	706	21
Were six letters, all lower-case	605	18
Were in dictionaries or word lists	492	15
Total of all above categories	**2831**	**86**

Table 2.3 – Character Distributions [21]

Characters	Count	Percentage
Lower-case only	3988	28.9%
Mixed case	5259	38.1%
Some upper-case	5641	40.9%
Digits	4372	31.7%
Meta-characters	24	0.2%
Control characters	188	1.4%
Space and/or tab	566	4.1%
.,;	837	6.1%
-_=+	222	1.6%
!#$%^&*()	654	4.7%
Other non-alphanumeric	229	1.7%

Some studies have even been able to develop common password lists and these lists are available for download on the internet. It has been reported, for instance, that the four most common passwords are *God*, *sex*, *love*, and *money*. [11] Sometimes users don't even change their password from the default one provided by the system.

The weaknesses in a user selected passwords has led to the creation of the steps that an Attacker might perform to derive a password:

1. no password
2. the same as the User ID
3. is, or is derived from, the User's name
4. common word list (for example "password," "secret," "private") plus common names and patterns (for example, "asdfg," "aaaaaa")
5. short college dictionary
6. complete English word list
7. common non-English language dictionaries
8. short college dictionary with capitalizations (PaSsWorD) and substitutions (0 for O, and so forth)
9. complete English with capitalizations and substitutions
10. common non-English dictionaries with capitalizations and substitutions

11. brute force, lowercase alphabetic characters
12. brute force, full character set

While the last step will always succeed, the steps preceding it are so time consuming that they will deter all but the dedicated Attacker. [11]

To prevent some of these attacks some password based authentication systems have introduced rules to limit user selected choices for their passwords. These rules are generally broken into two categories:

1. **Dictionary rules**: The system maintains word lists that the User selected password is compared against, and if the selected password is in the list it is not allowed.
2. **Composition rules**: Require the Users to select passwords that include some combination of lowercase and uppercase letters, symbols, and numbers.

NIST uses password entropy has a way to measure the security of passwords in their authentication guidelines. [10] Figure 2.2 summarizes their estimates of bits of entropy in a password of a given length, full estimated password guessing entropy can be seen in Appendix A.

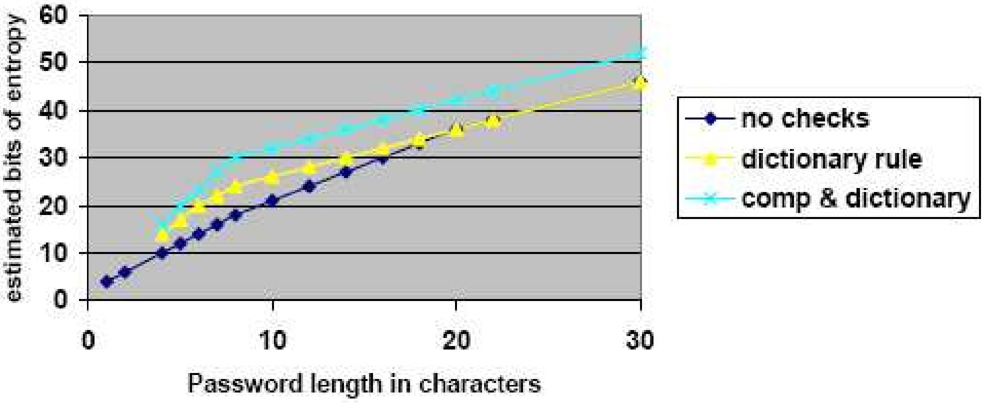


Figure 2.2- Estimated User Selected Password Entropy vs. Length [10]

In addition to guessing a password there are other ways the password may be disclosed to an Attacker. Other common attacks include:

- Social engineering – the User is somehow tricked into divulging a password to the Attacker
- Purposeful discloser – this is a subset of social engineering where the User voluntarily gives the password to an acquaintance to accomplish some task, allowing the acquaintance complete access to the User's account
- Written down – Users afraid of forgetting their passwords will often write them down and leave them in an easily accessible location
- CSP attack – the Attacker may obtain the Username and password list from the CSP

- Password reuse – Users will often use the same passwords in multiple systems, meaning that an Attacker may only have to obtain the password for a person's home computer or webmail account to guess the password to their bank account.

2.3.2 KBA Weaknesses

Since KBA authenticates a claimant based on the correctness of their answers to questions about the life of the claimed identity the authentication is based on the claimant's knowledge of those facts and not the claimant. Thus KBA is considered to be a form of "something you know" authentication. However, unlike other authentication factors of this type, the information is not secret.

KBA tends to focus itself around facts about the personal life of an individual. However, sometimes the questions used in KBA are not necessarily "facts." Examples of such questions include questions such as:

- What is your favourite colour?
- What is your favourite drink?
- Who is your best friend?

At the time of enrolment when these questions are first posed to the individual the individual provides an answer that may be true at the time. Since the answers to these types of questions are not necessarily facts and the answers may vary over time during the Authentication Phase a legitimate user may not know the answer provided during enrolment leading to the legitimate user being falsely rejected.

An interesting weakness in KBA is that it is time consuming, both during the enrolment phase and the authentication phase. This time consuming nature of KBA can cause users to zero out and not think about their answers to the questions. If there are too many questions in the enrolment phase a user may provide silly or untrue answers just to finish enrolment and this can lead to the User's inability to correctly answer the same questions during the authentication phase. The same is true during the authentication phase the User may find the number of questions ridiculous and be frustrated with the system and provide answers without thinking about them leading to a false rejection.

The responses to the question in KBA are often not secret. A number of these questions can be considered "out-of-wallet" questions where the correct responses can be found in a legitimate user's wallet or known by someone with some familiarity with the user. A determined attacker can conceivably with some effort obtain all the information necessary to successfully pass authentication. This *data-mining attack* is possible through the non-secrecy of the information used in KBA, the attacker can obtain this information through various forms of social engineering with the user themselves or through the user's friends and family (the attacker may even be a friend or family member); examining public records of the user such as birth certificates, driver licenses, credit history; and by intercepting and examining the User's mail both electronic and physical.

Since the information used by KBA is not secret it is important that the information can not be found by an Attacker in a single place: such as looking in a user's wallet or by eavesdropping on an un-encrypted KBA session. However, since the same information may be used in KBA systems at multiple different service providers such as an individual's bank, health services provider, phone and utility providers as well as various online services and retailers, there is an abundance of places where all the information required to successfully impersonate a user resides. With these multiple information repositories in existence an attacker has an ever increasing number of opportunities to gain access to that information allowing an attacker to possibly seize that information from one service provider and impersonate a user not only at that service but at a different service.

Often, for example in telephone banking, KBA is performed by an Agent of the service provider. The Agent will receive the Claimant's identity and query the verification system for a set of questions to ask the Claimant. The Agent could possibly record the questions and the correct responses and either provide that information to an Attacker or act as an Attacker.

A criticism of KBA is that it can alienate certain individuals. Often a KBA system will have a set list of questions that an enrolling user must provide responses for, however, sometimes due to the life experiences of the User they may be unable to provide suitable answers for example an individual who moved frequently as a child could not provide a suitable answer to the question "what street did you grow up on."

Like in all systems that use "something you know" authentication, the answers to the questions can be guessed. The ability of an Attacker to successfully guess an answer depends on who the Attacker is and their relationship with the User and what the question is (i.e. how personal the information is to the User). Santosh Chokhani [22] formulated a measure for the "Guessability of KBA", seen in Appendix B.

2.3.3 Biometric Weaknesses

Ideally, a biometric is measurable, unique to an individual, doesn't change over time and not easily duplicated or forged; however, it is possible to exploit weaknesses in the collection and processing methodologies implemented in the system to successfully attack the system. For example, recent research has shown that it is not very difficult to steal a biometric trait, create its copy and use the fake trait to attack biometric systems, particularly fingerprint biometrics. [22]

Like tokens employing either of the other authentication factors a token employing biometric authentication is susceptible to attacks at any point in the authentication process. Figure 2.3 illustrates a typical biometric authentication process and the common attacks at the entities and transmissions involved.

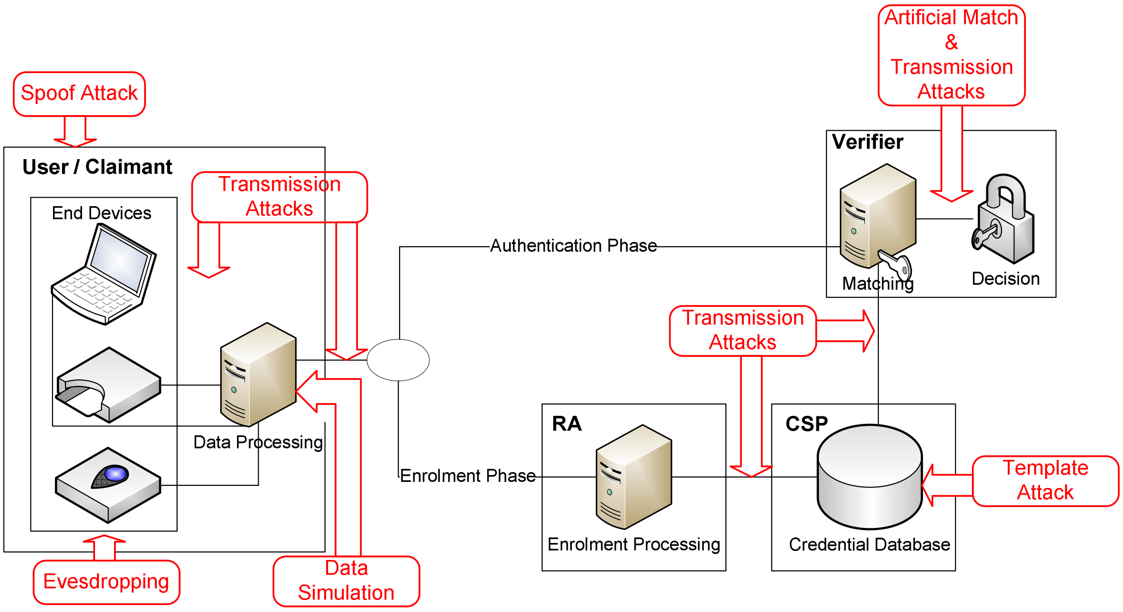


Figure 2.3 - Attacks on Biometric Authentication

Many of the attacks highlighted in Figure 2.3 are attacks on the authentication process rather than an attack specific to the biometric and were discussed in the previous section. However, some of these previously discussed attacks may be more severe or have longer lasting effects when applied to biometric-based tokens.

Of particular interest is the Template Attack, where an Attacker manages to steal the biometric templates: once an individual's biometric template is stolen the individual will forever lose the use of that biometric. Additionally a stolen biometric template can be used to reverse engineer the system to create synthetic biometric samples to use during the authentication phase. While the threats to a biometric CSP has a longer lasting effect than to a CSP in a password-based system, the attacks, threats and protection methods remain the same.

Of the attacks illustrated in Figure 2.3 a Spoof Attack is of particular interest in biometric-based authentication since it is a special attack on the biometric authenticator. Since biometrics are not secret they can not be protected like passwords or smartcards and since people can leave a biometric trail without knowing it and that information can be captured, copied and forged. [22] A biometric spoof attack occurs in two stages: "first, capturing the biometric sample belonging to the enrolled user and then creating a copy of the captured sample by means of an artefact." [22]

For example, in order to spoof attack a voice authentication system an attacker must obtain a copy of a user's voiceprint. This voiceprint can be obtained through social engineering fairly easily: an example presented in [22] is of a user receiving a phone call informing them of technical difficulties and then being asked to read some phrases, numbers and words several times and thus allowing the attacker to successfully capture a voiceprint without the user's knowledge.

It is, however, “fairly difficult to defeat a good commercial speaker-verification system with a recording. The voice signal input into a microphone or telephone held close to the speaker’s mouth differs markedly from a signal captured even as close as a foot away from the speaker. Moreover, many commercial speaker-verification systems look for telltale auditory signals, distortions, exact matches, and other indications that a recording has been used. As a result, creating a recording that can fool these systems is a difficult and costly challenge.” [17]

A common approach to prevent spoofing in a biometric system is to incorporate a method known as liveness detection. In liveness detection the system attempts to detect whether the sample is being provided by a live human by checking the physical properties of the live biometric. In the case of voice examples of liveness detection include using a text-dependent system that randomly selects what text the User is to speak and verifying that the correct phrase was spoken and measuring the flow, pitch and timbre of the phrase as well as statistically analyzing the voice sample to detect if it had been digitally altered or pre-recorded.

A recent United State patent [24] describes a method of conversational data mining. This method is able to extract unique acoustic features of a speech waveform and correlate the features to attributes such as “gender, age, accent, native language, dialect, social economic classification, educational level and emotional state.” [24] While the original intended use of this voice system was for data mining to support business logic the abilities of this system can be used to support liveness detection and spoofing resistance. This system demonstrates the ability of a computerized system to uniquely identify key attributes of a user’s voice and even their current emotional state demonstrating the ability of a voice biometric system to successfully identify and resist spoof attacks.

Voice biometric systems, which have different modes of operation, text-dependent, text-prompting and text-independent, can add or subtract from the general level of confidence in the authentication. A text-dependent voice authentication application “is more or less like two factor authentication, which adds an extra layer of security to the system.” [22] Text-prompting is similar to text-dependent and adds the extra layer of security by creating a challenge-response mechanism, which is a form of liveness detection. Text-independent applications, however, by allowing the User to choose to say any word or phrase to authenticate themselves have decreased system security for the higher user convenience.

A text-dependent voice biometric system, where the user has speaks a password to the system or a text-prompted system that uses KBA factoids for the prompts is a two-factor authentication system employing both “something you are” (voice biometric) and “something you know” (password, KBA) authentication factors. Such text-dependent systems also incorporate both static and dynamic authentication attributes: static attributes such as the spoken text (password, KBA responses) and dynamic attributes such as the way it is being spoken (flow, pitch, timbre, emotional state). Such a system can ensure that the data submitted in any two authentication sessions will ever be exactly the same.

Another method of preventing spoofing in a biometric system is multi-modal biometric fusion. Multi-modal biometric fusion involves combining multiple mono-modal biometric systems; the example seen in Figure 2.4 combines a voice biometric with face recognition. In Figure 2.4 the biometrics are associated together by using lip movement as a visual speech feature. “Research into the fusion of visual and thermal face imageries has shown that the combination of thermal IR image data with correlation filters has improved the performance of face recognition techniques." [22] The combination of two biometrics can make it more difficult for a spoofing attack to be successful because the (in the example) relationship between speech and lip movement is being monitored.

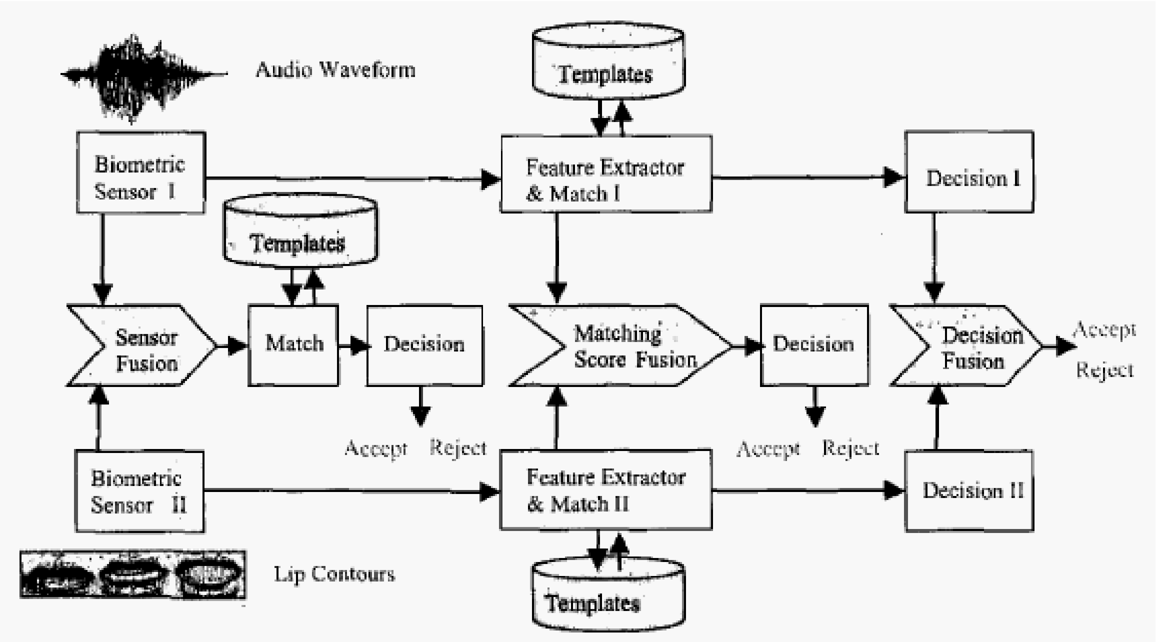


Figure 2.4 - Multi-modal biometric system [22]

If the two (or more) biometrics used in multi-modal biometric fusion are independent, for example voice and fingerprint, each factor can be tried independently and thus make the system "more or less like a two-factor authentication such as password and biometrics." [22] and sensor fusion cannot be performed.

2.4 Authentication Requirements

The previous section explored weaknesses and attacks in the authentication process and how the main objective of establishing confidence in a user's identity can be defeated. This section will establish resistance requirements to these attacks and threats to establish measures of confidence in the resulting authentication.

The United States Office of Management and Budget (OMB) issued a memorandum [25] to all government agencies that containing E-Authentication Guidance for Federal Agencies. The OMB E-Authentication Guidance, applying to remote authentication of human users, sought to help agencies to "identify and analyze the risks associated with each step of the authentication process" as well as establishing four identity authentication assurance levels for e-government transactions. The four established assurance levels, which were defined in terms of the consequences of the authentication errors and misuse of credentials, are:

- **Level 1:** Little or no confidence in the asserted identity's validity.
- **Level 2:** Some confidence in the asserted identity's validity.
- **Level 3:** High confidence in the asserted identity's validity.
- **Level 4:** Very high confidence in the asserted identity's validity.

NIST in a special publication entitled Electronic Authentication Guidance [10] provides "technical guidelines to agencies for the implementation of electronic authentication," as presented in the OMB E-Authentication Guidance. Table 2.4 contains the threat resistance guidelines contained in the NIST special publication describing what threats the authentication protocol must be resistant to in order to obtain a certain assurance level. Table 2.5 is a table summarizing the NIST requirements for what token types can be used to obtain a certain assurance level.

Table 2.4 - Authentication Protocol Threat Resistance per Assurance Level [10]

Authentication Process Attacks/Threats	**Threat Resistance Requirement**			
	Level 1	**Level 2**	**Level 3**	**Level 4**
Online Guessing	Yes	Yes	Yes	Yes
Replay	Yes	Yes	Yes	Yes
Session Hijacking	No	Yes	Yes	Yes
Phishing/Pharming (Verifier Impersonation)	No	No	Yes	Yes
Man in the Middle	No	Weak	Weak	Strong
Denial of Service/Flooding	No	No	No	No

Table 2.5 - Token Type by Assurance Level [10]

Token Type	**Assurance Level**			
	Level 1	**Level 2**	**Level 3**	**Level 4**
Hard Cryptographic Token	Yes	Yes	Yes	Yes
Soft Cryptographic Token	Yes	Yes	Yes	
Zero Knowledge Password	Yes	Yes	Yes	
One-time Password Device	Yes	Yes	Yes	

Strong Password	Yes	Yes		
PIN	Yes			

It can be observed in Table 2.4 and Table 2.5 that for Level 4 authentication, the highest assurance level, that the authentication protocol is resistant to all the attacks discussed previously and that a hard cryptographic token is required. To achieve Level 1 authentication, the lowest assurance level, the minimum requirements is resistance to Online Guessing and Replay attacks and that a password or PIN may be used as a token.

NIST [10] sought to identify the requirements of an authentication protocol to meet user-identity assurance levels required for certain government systems; the assurance levels were predetermined based on the consequences of authentication errors. Using this information and the previously presented information regarding the GUAP and its weaknesses as well as the weaknesses of Tokens and authentication factors this section will establish measures on the confidence level in an authentication system.

2.4.1 Authentication Confidence

The goal of an authentication system is to "establish confidence in the user identities presented to an information system." [10] The GUAP is the process that is used to accomplish this goal; however, it has many weaknesses and points of attack, both on the process itself and the factors used in verification. Therefore, for an authentication system to be confident in its authentication of a user's identity it must be confident in its own operation.

The entities and transmission lines in the GUAP can be divided into two sections: those within the immediate control of the authentication system and those outside of that control. For example the Verifier, CSP, RA and their communications are always inside the control of the authentication system and the User/Claimant entity is always outside its control. In some situations the End Devices are inside the control of the system, such as when all the entities except the Claimant are in a single tamper-proof device, such as a retinal scanner attached to a safe door. In remote authentication, the End Devices are outside the control of the system, such as the keyboard, or even a fingerprint reader, on the User/Claimant's personal computer, or perhaps the User/Claimant's personal telephone. Most commonly, in remote authentication systems, the points outside the direct control of the authentication system are to the left of the Verifier and RA entities in Figure 2.1; this includes the End Devices, the transmission from the End Devices to the DPU and the transmission out of the DPU.

For an authentication system to have a measure of confidence in its authentication of a user's identity the authentication system must have a measure of confidence in the parts of the authentication process inside its control as well as confidence in the parts of the process that it does not control. Confidence in the parts that are inside the control of the system means that the system has confidence in its ability to protect the authentication process from the discussed weaknesses and attacks. Confidence in the parts of the process outside the control of the system means that the system has confidence that even if at any point any of these parts were compromised to an attack the system is still confident in its ability to authenticate the identity of a user.

Table 2.6 illustrates the requirements in inside and outside confidence for an overall authentication confidence level.

Table 2.6 - Authentication Confidence Level

Authentication Confidence Level (%)	Confidence in Inside Entities and Transmission Lines	Confidence in Outside Entities and Transmission Lines
100	Confident	Confident
75	Confident	Moderately Confident
50	Confident	Not Confident
25	Moderately Confident	Don't Care
0	Not Confident	Don't Care

In order for a system to have an authentication confidence level above fifty percent it is necessary that the system has complete confidence in the entities and transmissions lines inside its control. If the system believes that the entities and their communications that it has control over are vulnerable to any one attack then the system should not have any reasonable confidence in its ability to authenticate an individual.

Table 2.7 - Threat Resistance Requirement for Inside Entities and Transmission Lines

Authentication Process Attacks/Threats	Threat Resistance Requirement		
	Confident	Moderately Confident	Not Confident*
Eavesdropping	Yes	Yes	No
Replay	Yes	Yes	No
Session Hijacking	Yes	Yes	No
Man in the Middle	Yes	Yes	No
RA Attacks	Yes	Yes**	No
CSP Attacks	Yes	Yes	No
Verifier Attacks	Yes	Yes**	No
* failure to resist any one of these attacks qualifies for this level ** Attacks are limited in type, see below.			

The situation where the system is Moderately Confident in the inside entities and transmission lines is a case where the system has inherent flaws due to its design but has insufficient checks and balances to remove them. These flaws arise not from an Attacker from the outside breaking into the system but from an Attacker on the inside that is able to obtain <credential, token> pairs from the system undetected.

An example of a moderate confidence in the inside entities and transmission lines is an authentication system where the token is an Agent-operated KBA centre where the same questions are asked every authentication session: the Agent is able to record a Claimant's identity and responses and can use that information to be falsely authenticated at a later time. In this example, the system is designed in such a way that it has no choice but to trust the operating Agent, but due to an Agent selection screening process the system has confidence in the trustworthiness of the Agent; however, there remains an intrinsic flaw in the system design.

For confidence levels above fifty percent the system must have complete confidence in its own operation but varying levels of confidence in the outside entities and transmission lines. Confidence in the outside entities and transmission lines is determined by the threats and attacks the outside entities and transmission lines are vulnerable to. Table 2.8 contains the requirements to determine the confidence level of the entities and transmission lines outside the immediate control of the authentication system. It should be noted that in Table 2.8 resistance to a threat does not mean that the given attack may not be successful, just that the success of the attack will not lead to an Attacker being falsely authenticated by the Verifier.

Resistance to Token Threats referred to in Table 2.8 is the ability of the system to maintain its confidence that the token in the <credential, token> pair is actually being submitted by the owner of the credential. This confidence arises from verifying both static and dynamic attributes inside the token. For example in a text-dependent or text-prompted voice biometric system verifying both the spoken word as being correct as well as the voice doing the speaking. Resistance to Token Threats is confidence that if any of the information used in verification was captured at any point in time by an Attacker and submitted by the Attacker to the verifier the Attacker would not be falsely authenticated.

Table 2.8 - Threat Resistance Requirement for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	**Threat Resistance Requirement**		
	Confident	**Moderately Confident**	**Not Confident**
Eavesdropping	Yes	Yes	No
Replay	Yes	Yes	No
Session Hijacking	Yes	Yes	No
Man in the Middle	Yes	Yes	No
Verifier Impersonation	Yes	No	No

End Device Attacks	Yes	No	No
Data Processing Unit Attacks	Yes	No	No
Token Threats (Capture)	Yes	No	No

No confidence in the outside entities and transmission lines arises from the inability of the system to trust that the <credential, token> pair submitted by the Claimant is truly being submitted by the User identity corresponding to that pair. This lack of confidence arises from the outside entities and transmissions being subject to too many attacks and threats. An example of no confidence in the outside entities and transmission lines is a simple password sent to the Verifier unencrypted over a normal phone channel. In this example the Verifier has no confidence that the password has not been disclosed, intercepted, or guessed by an Attacker. To improve the confidence in the outside entities and transmissions steps need to be taken by the system to reduce the number of threats and attacks possible.

Moderate confidence in the outside transmission lines arise from the inability of the system to truly verify that the token submitted in the <credential, token> pair is actually being submitted by the owner of the credential. The inability to truly verify arises from the nature of the authentication factors being used, for example something you know authentication, where the Verifier is verifying that the Claimant knows some information, not the Claimant themselves. An example of moderate confidence in the outside entities and transmission lines is a complex password sent encrypted to the Verifier over the internet. In this example the system has no confidence that the User has disclosed their password to another person, an Attacker has captured the encrypted password, broken the encryption and recovered the password, or that the Attacker is successfully guessing the password.

Complete confidence in the outside entities and transmission lines arises from the ability of the system to verify that the <credential, token> pair is actually truly being submitted by the corresponding user identity. When completely confident in this verification ability it does not matter if any attack on the outside entities and transmission lines is successful or not since the system is actually authenticating the User.

An example of complete confidence in the outside entities and transmission lines is a two-factor authentication system utilizing a text-dependent or text-prompted voice biometric system, where the voice sample is sent over a secured channel to the Verifier. This system establishes complete confidence through the following:

- The secured channel prevents transmission-based attacks and Verifier Impersonations
- A well constructed voice verification system with liveness detection can detect spoof attempts such as a voice recording
- Verification of dynamic attributes (the physical traits of the voice) ensure that if all the information used for verification in a single session (or multiple sessions) were captured an attacker could not use it to spoof a user do the ever changing nature of the attributes

- Verification of static attributes in the token (the spoken words) provide a second authentication factor to strengthen the authentication
- End Device and DPU Attacks are prevented since the capture of the data used in the authentication session will not compromise future authentication sessions

2.5 Chapter Summary

This chapter explored the GUAP identifying threats, attacks and vulnerabilities in the process. Following the discussion on attacks and weaknesses this chapter introduced measures for an authentication system to provide a confidence level to meet its objective establishing confidence in user identities presented to an information system. The next section will examine currently existing systems, many of them privacy-critical, and establish their authentication confidence level.

Chapter 3
Analysis of Authentication Systems

3.1 Skype

3.1.1 Overview

The use of Voice over IP (VoIP) telephony has greatly increased in recent years: A 2004 study [26] that found that 1% of houses with broadband access were using some form of VoIP service and that by 2009 that number would increase to 17%; while a 2007 study [27] found that 20% of United States businesses were using VoIP and predicted that two-thirds of businesses will have some form of VoIP service by 2011. This increasing use of VoIP in home and business has led to an increased use of VoIP in privacy critical systems; as a result it is of interest to evaluate the authentication mechanism of a well known and widely deployed peer-to-peer VoIP system: Skype [28].

The Skype Client, which is installed on a standard personal computer is available for free download at www.skype.com The Client allows its users to place voice calls and send text messages to other Skype users and in addition to placing voice calls to any telephone accessed through the Public Switched Telephone Network (PSTN).

While the Skype Client is available for free download the software is proprietary and only compiled versions of the software client can be obtained, creating difficulties in analyzing the authentication method. Fortunately, several independent analyses of Skype have been performed and the protocols used by the software are understood. The analysis of the User authentication portion of the Skype protocols contained in this case study build on some of these analysis's of the Skype protocol.

3.1.2 The Skype Network

The Skype network is an overlay peer-to-peer network that consists of two types of nodes: ordinary hosts and super nodes. An ordinary host is a Skype application that can be used to place voice calls and send text messages to other applications. A super node is an ordinary host's end-point on the Skype network; any ordinary host with a public IP address and having sufficient computing resources and network bandwidth can become a super node. [29]

To login to the Skype network an ordinary host connects to a super node and then registers itself with the Skype Login Server. The Skype Login Server is not a node on the network; however, any connection to the network must authenticate itself to the Skype Login Server in order to operate on the network. All user names and passwords are stored on the Login Server and user authentication is performed at the Login Server. The Login Server is the only "central server" on the Skype network. Figure 3.1 illustrates the relationship between the ordinary hosts, super nodes and the Login Server. [29]

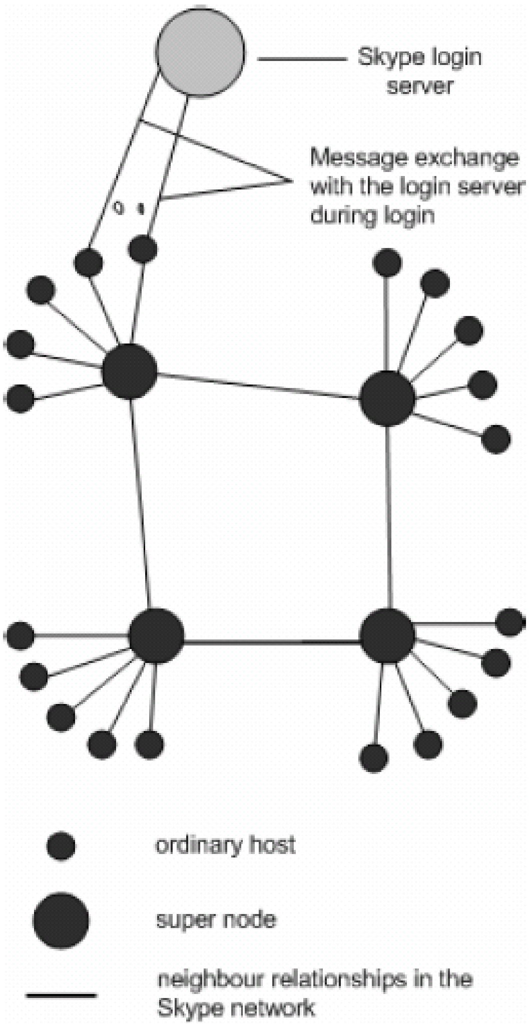

Skype login server
Message exchange with the login server during login
ordinary host
super node
neighbour relationships in the Skype network

Figure 3.1 - Skype Network [29]

3.1.3 The Skype Authentication Process

3.1.3.1 Enrolment Phase

The Enrolment Phase of the User Authentication process is performed during registration. During the registration process a new Skype user will select a username to use on the Skype network as well as an accompanying password. The complete registration process is illustrated in Figure 3.2. The below understanding of the Skype protocol is the result of Tom Berson's analysis [30] which was performed with the co-operation of Skype Technologies.

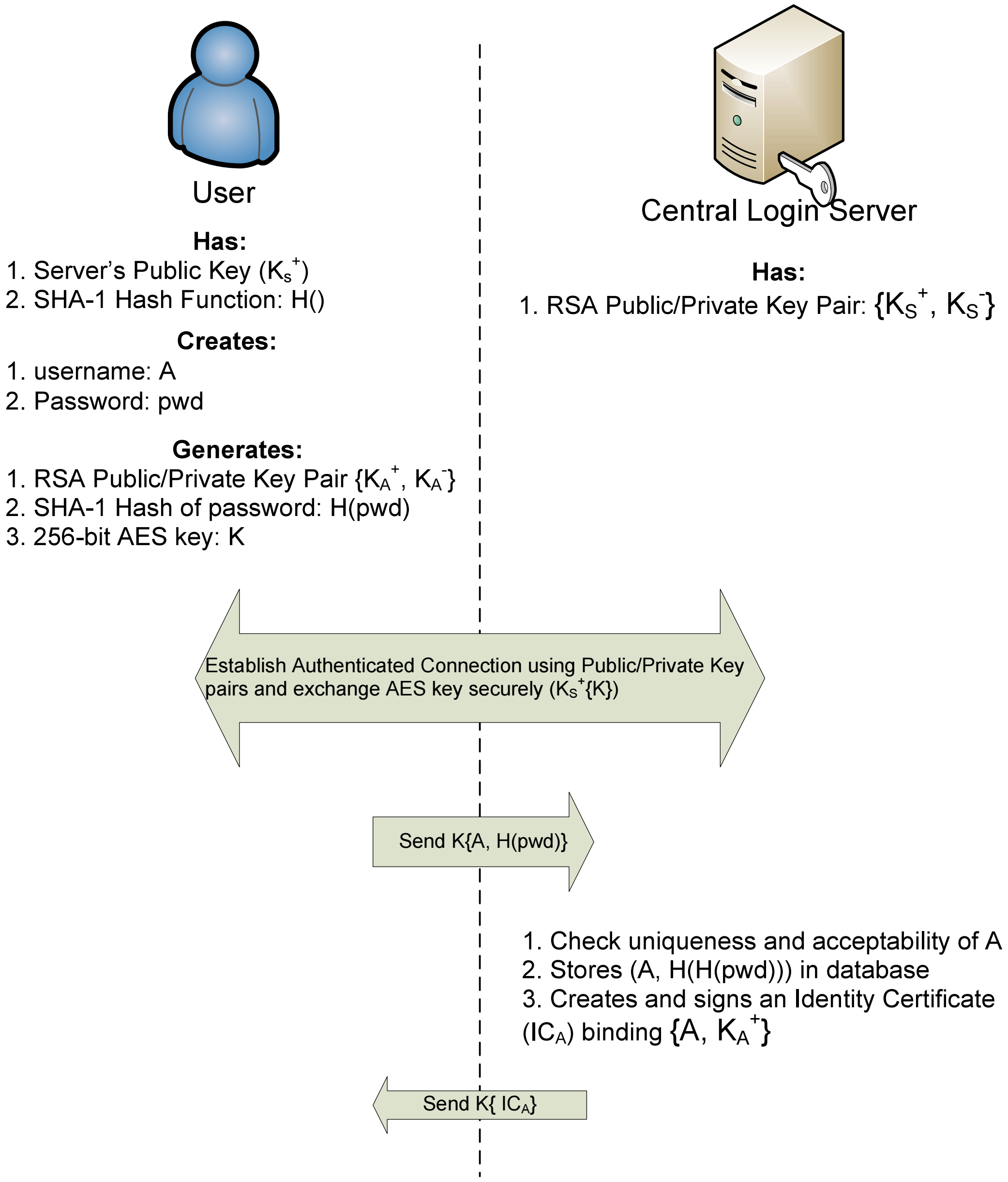

User
Has:
1. Server's Public Key (K_s^+)
2. SHA-1 Hash Function: H()
Creates:
1. username: A
2. Password: pwd
Generates:
1. RSA Public/Private Key Pair {K_A^+, K_A^-}
2. SHA-1 Hash of password: H(pwd)
3. 256-bit AES key: K
Central Login Server
Has:
1. RSA Public/Private Key Pair: {K_S^+, K_S^-}
Establish Authenticated Connection using Public/Private Key pairs and exchange AES key securely (K_S^+{K})
Send K{A, H(pwd)}
1. Check uniqueness and acceptability of A
2. Stores (A, H(H(pwd))) in database
3. Creates and signs an Identity Certificate (IC_A) binding {A, K_A^+}
Send K{ IC_A}

Figure 3.2 - Skype Enrolment Phase

It can be observed from Figure 3.2 that the enrolment phase consists of the following steps:

1. The User selects a desired username, denoted *A*, and a password, denoted *pwd*
2. The User's Client generates an RSA key pair: (K_A^+, K_A^-), computes a SHA-1 hash of the password: H(pwd) and stores K_A^- and H(pwd) as securely as possible on the User's platform.

3. The User's Client generates a 256-bit AES symmetric key (K)
4. The User's Client encrypts K using K_S^+ and encrypts (A, H(pwd), K_A^+) with K.
5. K_S^+\{K\} and K\{A, H(pwd), K_A^+\} are sent to the central server
6. The central server extracts K using its private key and decrypts (A, H(pwd), K_A^+) and determines if A is unique and acceptable.
7. The central server stores (A, H(pwd)) in its database
8. The central server creates and signs an identity certificate for A which contains the central servers RSA signature binding A and K_A^+.
9. The identity certificate is sent to A.

It can be observed that the authentication of the Skype Login Server by the Skye Client requires that the Skype Client know the public key of the Login Server. To allow for this the Login Server's public key is installed in every Skype Client at build time. [30]

Some simplification of the authentication process has been performed above. The central server in fact has two sets of RSA key pairs: one with a 1536 bit modulus and another with a 2048 bit modulus. The central server chooses to use the longer key pair if the User has subscribed to any premium service. Another simplification is that the central server is in fact many servers with different functions and replicated many times, however, security restrictions have been put in place that can allow the devices that compose the central server to be thought of as a single server. [30]

3.1.3.2 Authentication Phase

During the Authentication Phase the Claimant is asked by the Skype Client to provide a username and password, which is then used by the central Login Server to authenticate the Claimant's identity. The Authentication Phase is illustrated in Figure 3.3.

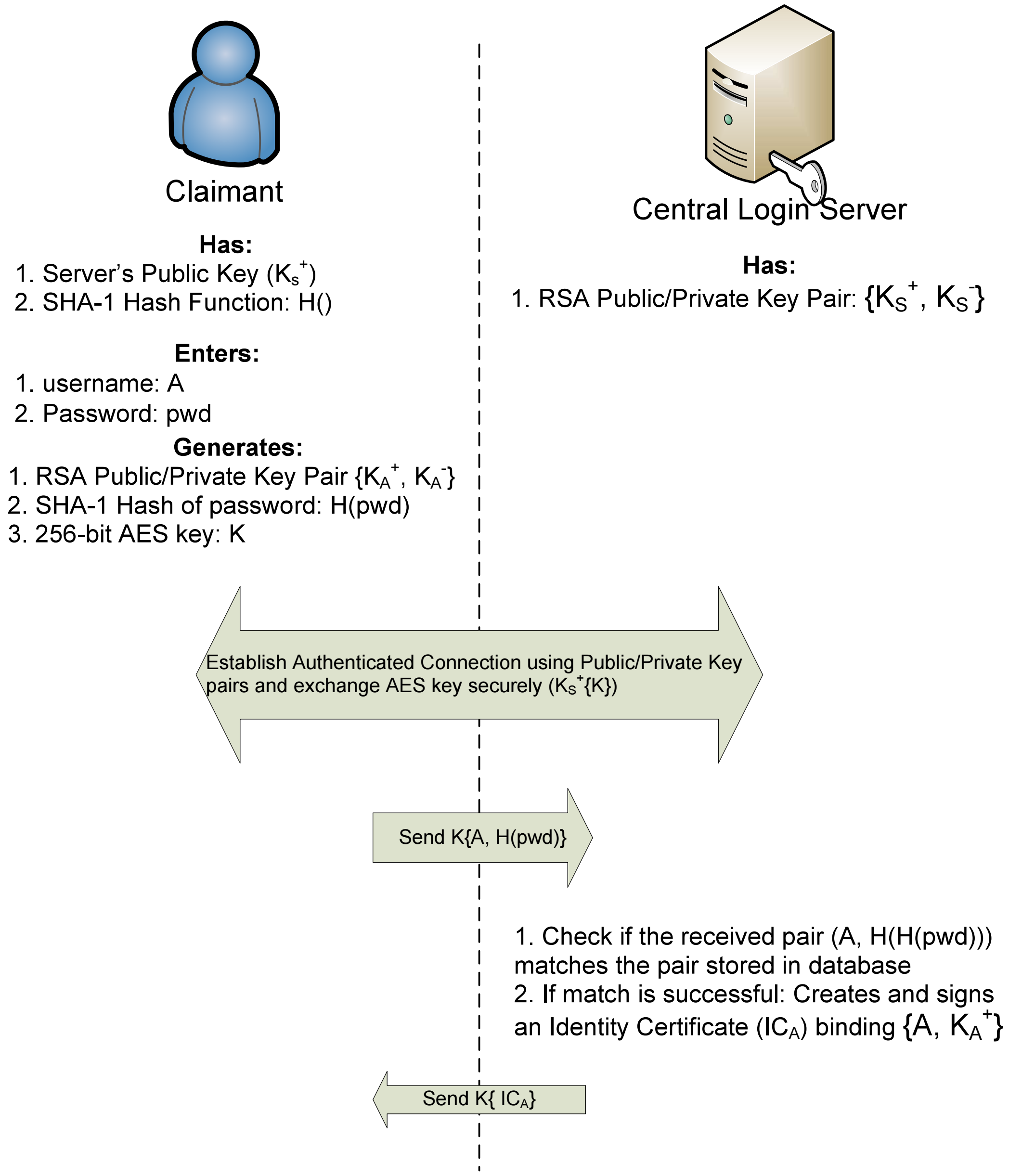

Claimant
Has:
1. Server's Public Key (K_S^+)
2. SHA-1 Hash Function: H()
Enters:
1. username: A
2. Password: pwd
Generates:
1. RSA Public/Private Key Pair {K_A^+, K_A^-}
2. SHA-1 Hash of password: H(pwd)
3. 256-bit AES key: K
Central Login Server
Has:
1. RSA Public/Private Key Pair: {K_S^+, K_S^-}
Establish Authenticated Connection using Public/Private Key pairs and exchange AES key securely (K_S^+{K})
Send K{A, H(pwd)}
1. Check if the received pair (A, H(H(pwd))) matches the pair stored in database
2. If match is successful: Creates and signs an Identity Certificate (IC_A) binding {A, K_A^+}
Send K{ IC_A}

Figure 3.3 - Skype Authentication Phase

It can be observed from Figure 3.3 that the enrolment phase consists of the following steps:

1. The Claimant enters their username, denoted *A*, and a password, denoted *pwd*
2. The Claimant's Client generates an RSA key pair: (K_A^+, K_A^-), computes a SHA-1 hash of the password: H(pwd) and stores K_A^- and H(pwd) as securely as possible on the User's platform.

3. The Claimant's Client generates a 256-bit AES symmetric key (K)
4. The Claimant's Client encrypts K using K_S^+ and encrypts (A, H(pwd), K_A^+) with K.
5. $K_S^+\{K\}$ and $K\{A, H(pwd), K_A^+\}$ are sent to the central server
6. The central server extracts K using its private key and decrypts (A, H(pwd), K_A^+) and determines if A is unique and acceptable.
7. The central server verifies (A, H(pwd)) from an entry in the database
8. The central server creates and signs an identity certificate for A which contains the central servers RSA signature binding A and K_A^+.
9. The identity certificate is sent to the Claimant.

3.1.3.2.1 Peer-to-Peer Authentication

Since the Skype network is a peer-to-peer based network, meaning that a Skype user will connect directly to another user, there is another instance of the authentication phase where the Users will authenticate each other at the start of a conversation. This peer-to-peer authentication is accomplished through the use of the certificates issued by the Login Server during the login process where the Users authenticated themselves to the central Login Server.

Users do not truly authenticate each other, merely authenticate the authenticity of the certificate issued by the central Login Server and ensure that the other user does indeed poses the private key corresponding to the public key contained in the certificate.

This method of authentication is sufficient provided that the cryptographic tools used in the certificates, their issuance and their verification are properly implemented and the central Login Server can be trusted to authenticate an individual correctly.

3.1.4 Legal Issues

It should be noted that while Skype is a widely deployed VoIP system it has been under heavy scrutiny by businesses and governments worldwide. The nature of its encrypted peer-to-peer connection allows it to traverse corporate firewalls and cause congestion on a corporate network as well as creating a channel for malware for malware to infect the network through.

Additionally, the encrypted peer-to-peer nature of Skype violates established legal requirements such as the mandatory requirement for telephony providers to provide a means for legal wiretapping of the phone channel. An additional example is unsanctioned usage of Skype by security brokers violates the requirement that they record and track all telephone calls.

Several countries, including the European Union and China, have passed laws that have declared the use of the Skype Client to be illegal.

3.1.5 Analysis of the Skype Authentication Process

Many steps have been taken by the developers of the Skype application to protect the authentication process from attacks through the use of cryptographic tools. Tom Berson's analysis [30] of the Skype software revealed that all cryptographic tools were properly implemented and there did not appear to be any attacks on the implementation of the tools, thus for the purpose of this analysis the cryptographic tools used will be assumed to be properly implemented.

The below analysis of the Skype user authentication process first examines the entities and transmission lines that are inside the control of the Skype authentication system before examining the entities and transmission lines outside the control of the authentication system. Following this analysis conclusions as to the confidence level of the authentication are made.

3.1.6 Analysis of Inside Entities and Transmission Lines

The entities and transmission lines inside the control of the Skype authentication system consist of: the transmission of the <username, password> pair from the Skype Client to the central Login Server; the database storing the <credential, password hash> pairs of user accounts (the CSP); the central Login Server (the Verifier); and the transmission between the database and the Login Server.

The analysis in [30] found that the internal entities involved in the authentication process were properly implemented and secure. No remote access to the Skype user database has been discovered and as a result is believed to be secure and inaccessible except from Skype administrative facilities. While it is not known if the database is encrypted it is known that the database stores a hash of the password hash, resulting in the stored pair: <username, H(H(password)>. [30]

No remote access the Login Server except for the verification of credentials submitted by the Skype Client has been discovered and as a result the Login Server is a result is believed to be secure and inaccessible except from Skype administrative facilities. Additionally, the transmissions between the Login Server and the database are also believed to be inaccessible by remote connections outside of the Skype administrative facilities.

3.1.7 Analysis of Outside Entities and Transmission Lines

The entities and transmission lines outside the control of the Skype Authentication system consist of: the User's keyboard (End Device), the Skype Client (Data Processing Unit) and the transmission between the keyboard through the User's personal computer to the Skype Client. In actuality the true "Data Processing Unit" is the personal computer of the User where the keystrokes are determined and there is an extra transmission between that module and the Skype Client process.

The transmission of the <username, password hash> pair from the Skype Client to the Skype Login Server, which is often a transmission line outside the control of the authentication system, is performed through a secure and authenticated channel where the Login Server authenticates the Skype Client as legitimate before establishing a secure (encrypted) connection. The secure and authenticated connection between the Skype Client and the Skype Login Server effectively establishes the transmission as inside the control of the Skype authentication system.

A password capture is perhaps the most effective way for an Attacker to spoof a user. Since the Skype Client operates on the application layer of the OSI protocol stack an attacker can install a keystroke monitoring tool onto a user's personal computer and capture the user's <username, password> pair as it is inputted by the user. The attacker can then use their own Skype Client to present the pair to the central Login Server and be falsely authenticated.

The Skype Client has a "Remember Me" feature that if selected by the User the Skype Client will store the username and password hash locally on a user's machine. The Skype Client uses the encryption algorithm available on the Operating System for the secure storage of the pair; for instance on a Windows machine the local store of the Username and password hash is stored using the Windows CryptProtectData API. [30] With the "Remember Me" feature in place any person able to access the User's machine can potentially either:

1. If logged onto the User's machine as the User simply open the Skype Client and submit the saved <username, password hash> pair to be falsely authenticated as the legitimate Skype user.
2. Recover the Operating System encrypted <username, password hash> pair and break the encryption to recover the Username and password hash.

A password guessing attack is also possible. In an attempt to thwart a guessing attack if there are seven consecutive failed login attempts Skype will lock a user account for one minute and prevent any further log in attempts, however, after that one minute has passed seven more attempts can be performed before the account is locked again for one minute. Locking the User account in this manner will limit the Attacker to only seven guesses per minute which will greatly increase the time taken for a guessing attack to be successful. However, the guessing attack is still feasible in a reasonable time frame as there are no password rules (other than length) enforced by the Skype authentication system.

Skype passwords can be anywhere in length from 4 to 256 characters and with a 94 character alphabet this results in a large number of possible passwords ($\sum_{k=4}^{256} 94^k$) which means that a brute-force guessing attack at a rate of seven passwords a minute can be considered impossible. However, since the passwords are user chosen and there are no rules governing the selection of passwords, a guessing attack is feasible as the Attacker can follow the previous described method of password guessing attack and may successfully guess a password in a reasonable time frame. The time taken in the guessing attack can be greatly reduced if the Attacker is familiar with the User they wish to impersonate. Additionally the Attacker may be able to perform an "over-the-shoulder attack"; where the Attacker observes the User enter their password and discovers information about the password, such as its length and certain characters in it, or even the password itself.

Skype also supports a "Forgot Password" feature. The feature is designed so that a legitimate user who has forgotten their password can enter their registered username and email address and request that the Skype service help them out. The Skype service will then send an email to the User's email address (it should be noted that this email address was provided by the User during registration) that contains a key. The user then goes to the website, enters their email address, username and the key from the email and creates a new password.

An attacker can exploit the "Forgot Password" feature of Skype to reset a user's password and gain access to the user's account. This exploitation, however, requires the attacker to either have access to or the ability to intercept the legitimate user's email.

Perhaps the greatest weakness in the Skype authentication process is the weakness of the authentication factor employed by the system. The Login Server has no assurances that a claimant presenting a username and password is the true owner of the username. The Login Server assumes that since the claimant presents a username and the correct corresponding password the claimant possesses the claimed identity. The login process is completely vulnerable to these types of attacks, where once an attacker has knowledge of a username and corresponding password they will be falsely authenticated.

3.1.8 Conclusions

Table 3.1 and Table 3.2 summarize the threat resistance of Skype to attacks on the inside and outside entities and transmission lines. It can be observed that Skype meets the requirements for complete confidence in the inside entities and transmission lines but does not meet the requirements for complete confidence in the outside entities and transmission lines.

Table 3.1 - Skype Threat Resistance for Inside Entities and Transmission Lines

Authentication Process Attacks/Threats	Resistance
Eavesdropping	Yes
Replay	Yes
Session Hijacking	Yes
Man in the Middle	Yes
RA Attacks	Yes
CSP Attacks	Yes
Verifier Attacks	Yes

Table 3.2 - Skype Threat Resistance for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	**Resistance**	**Location of Attack**
Eavesdropping	No	Transmission between keyboard and Skype Client
Replay	No	Attacker can replay data captured during transmission between keyboard and Skype Client
Session Hijacking	Yes	
Man in the Middle	Yes	
Verifier Impersonation	No	A user could be tricked into attempting to use a fraudulent Skype Client
End Device Attacks	No	Attacker can perform an "over-the-shoulder-attack" Keystroke logger
Data Processing Unit Attacks	No	Skype Client "Remember Me" feature
Token Threats	No	Determined and knowledgeable Attacker launched guessing or dictionary attack

It can be observed in

Table 3.2 that several attacks are possible on the entities and transmission lines outside of the control of the Skype Authentication System. Due to the number and type of attacks possible the Skype authentication system can have no confidence in the entities and transmission lines outside of its control.

While the Skype authentication process meets the requirements to have complete confidence in the entities and transmission lines inside of its control it fails to meet the requirements to have any confidence in the entities and transmission lines outside of its direct control and as a result the Skype system can only have a 50% Confidence Level in the output of its authentication process.

3.1.9 Recommendations for Improvement

Confidence in the outside entities and transmission lines can be improved by eliminating the transmission based attacks between the keyboard and the Skype Client. The main threat in these transmission attacks is monitoring software installed by the attacker on the user's personal computer that records keystrokes: this attack can be eliminated by incorporating a scheme that uses mouse clicks to enter some secret information, preferably in addition to the existing <username, password> pair.

An example improvement is one that maintains the static authentication attributes in the token from the password but also incorporates dynamic attributes as well. Dynamic attributes can be added after the <username, password> pair is verified by the Skype Login Server by having the server respond by sending the Client several randomly selected images and an image that the user as pre-selected as their own secret image. The Client then displays the images randomly distributed in a grid and the user selects their secret image; by having the Client display the images in a random order the possibility of an attacker process recording mouse location at the time of the click is eliminated. In this example the dynamic attribute that is thwarting monitoring software is the ever changing location of the secret image: thus if an attacker were to capture both keystrokes and mouse location the attacker would still be unable to pass authentication.

By eliminating the attacks on the transmission lines outside of the control of the Skype System the system can gain a 75% Confidence Level in the output of the authentication process. However, since attacks are still possible on the revised scheme (such as monitoring software that also captures the screen contents at the time of the mouse click) further enhancements are required.

To gain a 100% Confidence Level resistance to attacks that capture the token must be introduced. Since the success of these attacks depends on the authentication factor employed in the authentication token used in the Skype authentication process a good method of eliminating these attacks is to switch to a biometric-based token.

A well implemented text-dependent voice biometric system which incorporates spoofing resistance and liveness detection is well suited to Skype as it does not require a Skype user to obtain any additional hardware. The proposed authentication process will proceed as follows:

1. the Skype Client establishes a secured channel with the Skype Login Server as before
2. the user enters his or her username into the Client
3. the user speaks their password
4. the <username, spoken password> pair is sent over the secured channel to the Login Server
5. the Login Server extracts biometric data for biometric verification
6. the Login Server uses speech recognition to determine the submitted password for verification
7. if both the password and biometric data pass verification the user is authenticated

It can be observed that this new process is s two-factor authentication system incorporating both "something you know" and "something you are" authentication factors. Additionally the verification of the token in this proposed authentication process involves the verification of both static and dynamic attributes: static attributes such as the spoken password; and dynamic attributes such as the changing nature of the human voice.

3.2 Financial Call Centers

3.2.1 Overview

Financial Call Centers is a very broad category with many different, independent financial companies operating their own call center and performing authentication of their remote customers. However, most call centers follow the same general process of authentication and therefore rather than evaluate the authentication process at each individual company's call center this evaluation examines the general authentication process performed at the call centers. The general authentication process described below was determined through an examination of the call centers of several different financial services.

3.2.2 Call Center Authentication Process

The caller authentication process at a financial call center differs slightly from GUAP. The Enrolment Phase, illustrated in Figure 3.4, is typically performed completely offline with the User and an employee of the financial service, hereafter referred to as an Agent, physically present at the same location. The Authentication Phase, illustrated in Figure 3.6 occurs over the PSTN and consists of two Verifiers, one of which is very often operated by an Agent.

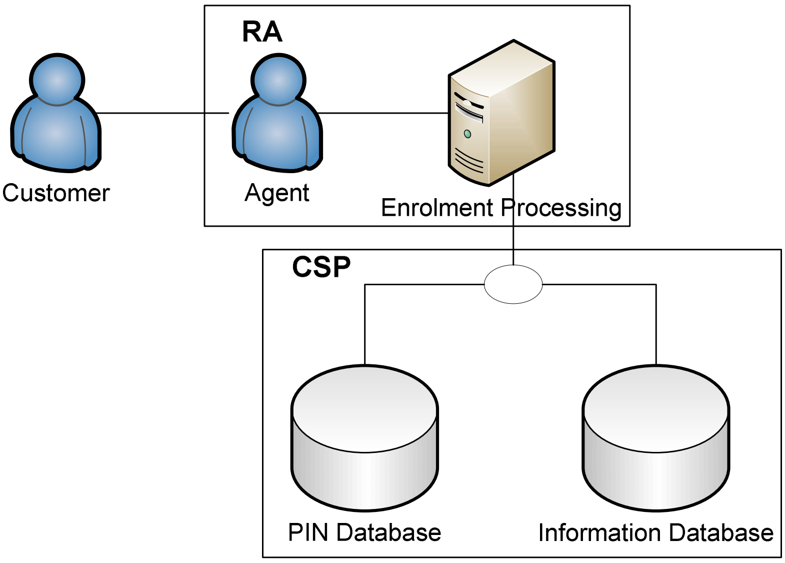


Figure 3.4 - Call Center Enrolment Phase

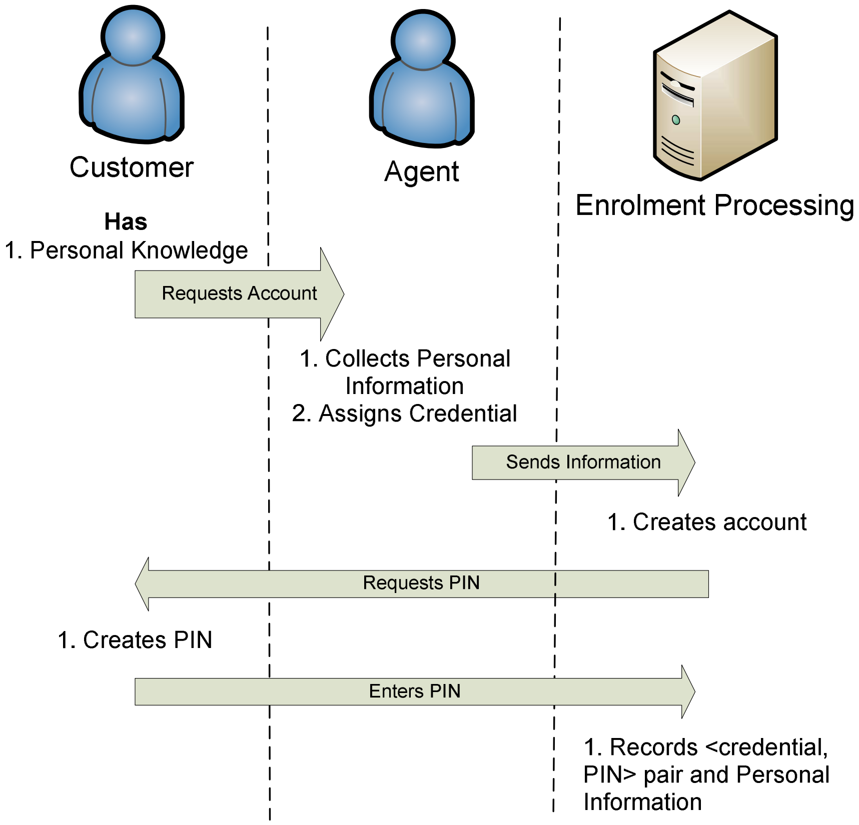


Figure 3.5 - Call Center Enrolment Process

Figure 3.5 illustrates the process that is undergone during the Enrolment Phase, namely when a customer registers or creates an account and becomes a user of financial service. This registration is performed physically by an Agent of the financial service while the customer is physically available to the Agent. At this time in the Enrolment Phase, a credential is issued to the new user; a common example of a credential is a bank card with an identifying 16-digit number printed on it. After the Agent assigns the credential to the new account the User will secretly select a PIN and enter it into the system; the secretly selected PIN will hereafter act as the User's token in the authentication process. After the PIN is entered the <credential, PIN> pair is stored securely in the financial institution's CSP.

Additionally wealth of personal information is collected by the Agent from the new User. Often this information relates directly to the service the User is obtaining from the financial institution, however, this personal information may not have any relation to the service and may just be additional information collected by the institution to be used for verification purposes as part of the Authentication Phase.

The Authentication Phase, illustrated in Figure 3.6 and Figure 3.7, is typically composed of two different Verifiers. In the Authentication Phase the Claimant connects to the system using their personal telephone over the PSTN.

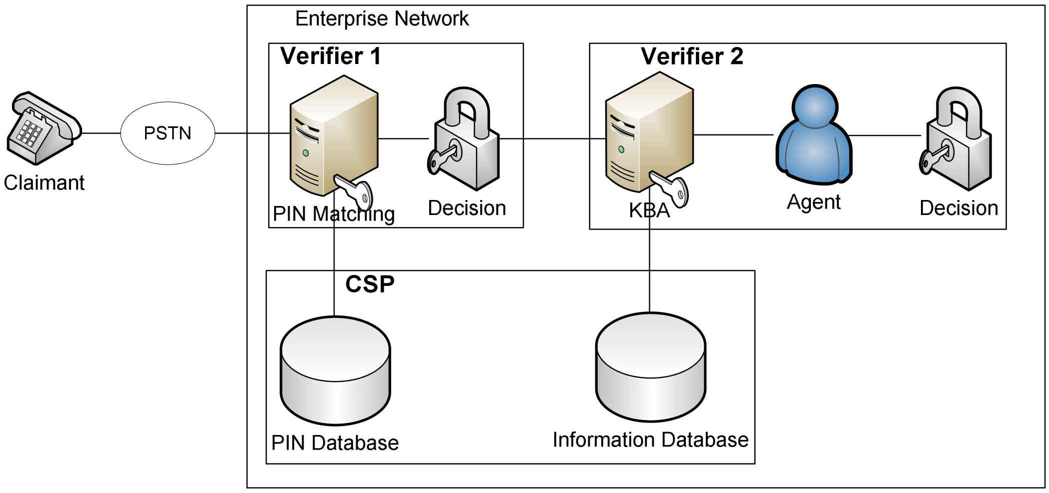


Figure 3.6 - Call Center Authentication Phase

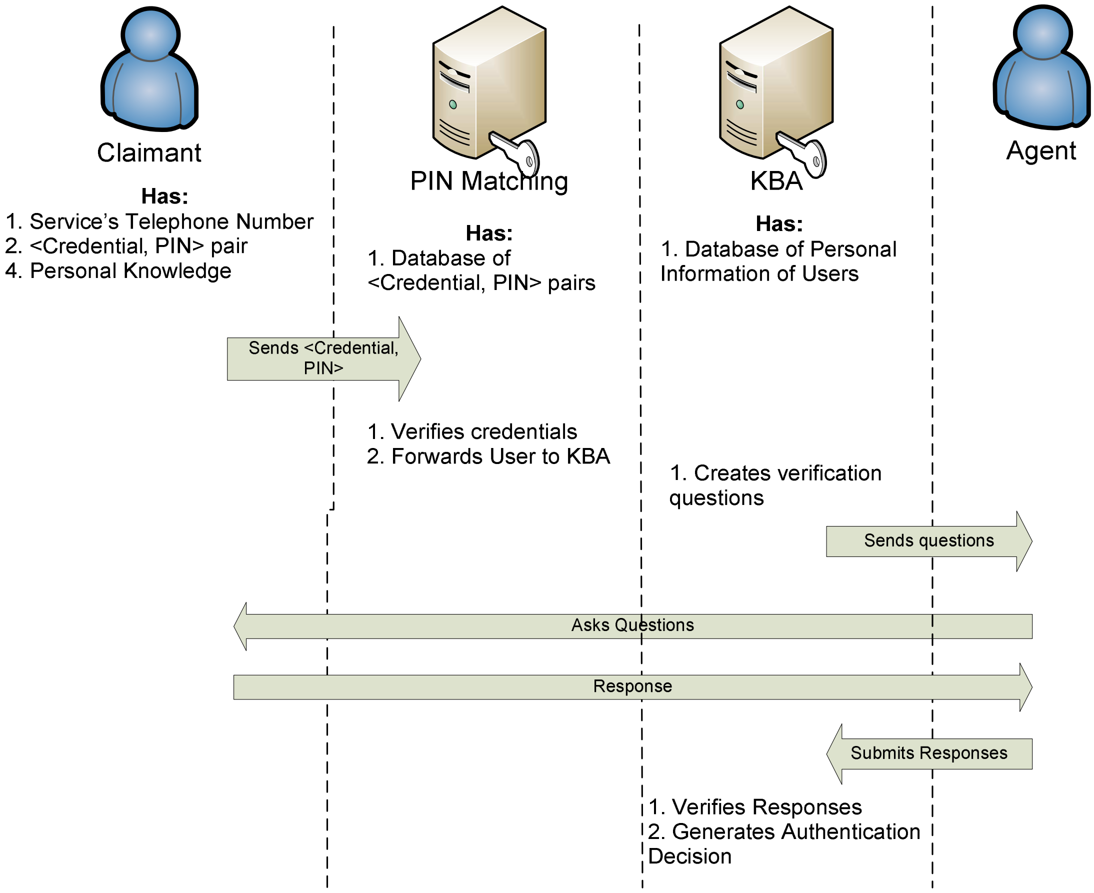


Figure 3.7 - Call Center Authentication Process

Immediately after connecting to the call center the Claimant is forwarded to first Verifier which requests the <credential, PIN> pair from the Claimant. The pair is entered by the Claimant using the touch-tone keypad on their personal telephone. The Verifier will then perform a matching operation and if the pair is verified as correct the Claimant is forwarded to second Verifier.

The second Verifier is very often operated by an Agent of the financial service and acts as a second level of identity verification through the use of KBA. The Claimant is forwarded to the second Verifier for the first Verifier along with the claimed identity allowing the second Verifier to automatically access the personal information corresponding to the claimed identity. The second Verifier will then generate a number of random questions from the personal information; these questions are submitted to the Agent who repeats them to the Claimant and enters the Claimant's responses into the Verifier. If the Claimant's responses to the questions are acceptable to the Verifier the Claimant will be authenticated as having the claimed identity.

3.2.3 Analysis of the Financial Call Center Authentication Process

The below analysis of the Financial Call Center User Authentication Process first examines the entities and transmission lines that are inside the control of the authentication system before examining the entities and transmission lines outside the control of the authentication system. Following this analysis conclusions as to the confidence level of the authentication are made.

It should be noted that it is during the Enrolment Phase for financial institution call centers that identity theft can occur through an Attacker fraudulently claiming the identity of another person. This section will set aside the idea of identity theft and fraudulent users in the Enrolment Phase and will examine how an Attacker can exploit weaknesses in the Enrolment and Authentication Phases to be falsely authenticated as a legitimate user of the financial service.

3.2.3.1 Analysis of Inside Entities and Transmission Lines

The entities and transmission inside the control of the call center are labeled in Figure 3.6 as the Enterprise Network. The interior entities consist of: Verifier 1; Verifier 2, which is often manned by an Agent; and the CSP, which may consist of multiple databases. The interior transmission lines are formed by the transmissions between these entities.

Since this evaluation is focused on the general Financial Call Center authentication process an examination of weaknesses in the interior entities and transmission lines at a particular call center is not significant. A particular Financial Call Center may implement no measures to protect the inside entities and transmissions or it may implement the methods discussed in Chapter 2 to protect the entities and transmissions from the discussed attacks. As a result it will be assumed that the general call center being analyzed has implemented prevention methods to secure the entities and transmission lines inside the control of the call center. The remainder of this subsection will analyze those entities and transmissions that differ from the GUAP in the Financial Call Center, most notably the Agent operating the KBA-based Verifier.

In a Financial Call Center there is an interesting transmission-based attack that can occur during the Enrolment Phase. Typically, a User enrols in the system with the assistance of an Agent while physically present at a branch office of the financial institution. In this situation the Agent is taking information from the enrolling user and entering it into the CSP. Often the information exchange between the User and the Agent is performed verbally or in some cases written documents. An Attacker can try to mount a transmission attack on this exchange by installing eavesdropping and monitoring devices (sound recorders, video cameras, keyboard loggers, screen captures, etc.) in the financial center. This Transmission Attack has less to do with the technology used in the authentication system and more to do with the physical layout of the financial institution. The most successful way of reducing this attack is isolation in the enrolment phase, including isolation of the terminal used to enter the information and isolation of the customer and Agent during the information exchange.

Due to the presence of an Agent in the Enrolment and Authentication Phases in order to have any level of confidence in the interior entities it is necessary that the system have confidence in the trustworthiness of the Agents. To have confidence in the trustworthiness of an Agent means that the Agent can be trusted to both authenticate a Claimant properly and not to falsely authenticate or reject them as well as be trusted not to record a User's personal information to either mount their own attacks or for sale to another Attacker.

During the Enrolment Phase, with the exception of the PIN privately entered by the new User all information in the Enrolment Phase passes through the Agent. The Agent could possibly record this information to mount an attack at a later time. Despite not having the PIN portion of the <credential, PIN> pair it is possible for a knowledgeable Attacker to mount a successful attack; how such an attack can succeed is discussed in the next subsection.

During the Authentication Phase an Agent could also record enough information from a legitimate User's responses to the KBA questions to mount an attack at a later time. If the KBA system does not randomly vary the question used between sessions, than the Agent has all the information required, however, if the system sufficiently varies the questions in the KBA phase the Agent may be unable to mount the attack.

3.2.3.2 Analysis of Outside Entities and Transmission Lines

The entities and transmission lines outside the control of the Financial Call Center consist of the User/Claimant, their End Device and Data Processing Unit (personal telephone) and the transmission over the PSTN between the telephone and the Financial Call Center's network.

The User/Claimant's End Device and their only interface with the remote system is a personal telephone that is connected to the financial institution over the Public Switched Telephone Network (PSTN). In the general situation the Claimant's personal telephone is a standard touch-tone telephone eliminating the ability for a secure, encrypted channel to be established with the financial institution; As a result all information exchanged between the Claimant and the financial institution is sent in the clear.

Attacks are possible on the End Device (the telephone receiver) and the Data Processing Unit (the audio converting mechanism). These attacks generally involve the Attacker capturing any portion of the information, such as the <credential, token> pair and the KBA questions and responses. End Device and Data Processing Unit attacks include but are not limited to:

- A "same room attack" where the Attacker is able to listen in the conversation to obtain KBA information
- Installing a monitoring device in the phone receiver to capture key presses and voice activity

Since the entire communication between the Claimant and the financial institution is unencrypted the authentication process is subject to a transmission attack by an Attacker between the Claimant's personal telephone and the Verifier. An Attacker can install a device somewhere on the communication channel between the two entities to observe and record the information sent by the Claimant to the Verifier and recover all information passed between the entities during the Authentication Phase. The Attacker can resubmit this recorded information at a later time and possibly be falsely authenticated as a legitimate user.

It is important to note that in addition to the Attacker capturing the <credential, PIN> pair during the Claimant's interaction with the first Verifier, the Attacker could also capture the questions and responses during the Claimant's KBA session with the second Verifier. Commonly, the KBA Verifier attempts to thwart such transmission-based attacks by randomly varying the questions queried to the Claimant. With this variance in the questions in place an Attacker would have to monitor several authentication sessions between the Claimant and the service and build a databank of responses to pass the KBA portion of the Authentication Phase. However, if the same verification challenges are used frequently or if there is little, if any, variation between the verification questions used per session an Attacker may be able to pass the KBA portion with only having monitored as little as one authentication session.

Another disadvantage with the PSTN is that there is no means of a user authenticating the Financial Call Center. A user only has knowledge of the Financial Call Center's phone number and by dialing that number assumes that they have connected to the Financial Call Center. However, an Attacker may be able to splice their own fraudulent service into the phone network and intercept all calls destined to the Financial Call Center. An Attacker who is able to intercept calls destined to the call center is able to perform both session hijacking and man in the middle attacks.

If the Attacker is unable to intercept call destined to the call center they could possibly register phone numbers differing from the call center by a single digit and install a fraudulent call center similar to the one belonging to the financial service. In this Pharming attack a legitimate user will accidentally enter the wrong phone number and will connect to the fraudulent call center and enter their <credential, token> pair and respond to the Attacker's KBA queries before the Attacker reports a system error and disconnects the call. A Phishing attack is also possible where an Attacker calls a user claiming to be the financial service, reporting computer errors, and asks for the <credential, token> pair and KBA information.

Another weakness in this authentication process is the employment of the "something you know" authentication factor. While the authentication system is using two different forms of this authentication factor to strengthen the authentication process the process is not truly two-factor authentication.

It can be argued that since the credential in the <credential, PIN> pair is very often a 16-digit number on the front of a physical card issued by the financial institution the authentication process is indeed two factor authentication using both "something you have" (card number) and "something you know" (PIN) factors. However, since the 16-digit number is written on the face of the card and the verification that occurs at the Verifier only verifies that the Claimant knows this number and the corresponding PIN and does not verify the Claimant's actual possession of the card the possible "something you have" factor reduces to the "something you know" factor.

Since a PIN is very often a four digit, user-selected number the possibility of an Attacker mounting a successful guessing attack against the first Verifier is very real. The success of a guessing attack, however, depends on the Attacker's ability to gain access to the credential portion (ex. 16-digit card number) of the <credential, PIN> pair as well as to successfully guess the PIN. Since the credential is very often the 16-digit number on the front of a card an Attacker who can gain access to this credential, for instance by stealing a user's wallet, may be able to mount a successful guessing attack.

To prevent a guessing attack, where the Attacker has knowledge of the credential, but not the PIN portion of the <credential, PIN> pair the financial service will suspend the account corresponding to that credential if there are several consecutive failed verification attempts. Most commonly an account is allowed three failed attempts before it is suspended and access to the system is denied, even if a correct PIN is entered following the suspension. Once the account is suspended the Attacker or account owner must go through the PIN reset procedure described below to reactivate the account.

To handle suspended user accounts and to provide a mechanism for a legitimate user who has forgotten their PIN, Financial Call Centers provide a means to reset a PIN. There are generally three approaches to dealing with resetting a PIN:

1. Authenticating the User over the phone using KBA and allowing a PIN reset
2. Receiving a PIN reset request and creating a new PIN which is delivered to the User using standard mail
3. Requiring the User to physically reset the PIN at a branch office

In the first, and most common, method a legitimate user connects to the Financial Call Center over the PSTN as usual, only instead of immediately entering their <credential, PIN> pair the User enters a pre-defined option to enter the password reset procedure. After entering this option the User is forwarded to an Agent who requests the User for their identity; after the Agent receives an identity claim the Agent will initiate the KBA procedure for that identity. If the User passes the KBA the User is authenticated as having the claimed identity and forwarded to the PIN reset tool where the User is allowed to create a new PIN.

By allowing a PIN reset to occur using the above described method, the authentication process is effectively reduced to just a single authentication factor: the KBA. In this system, an Attacker who is able to pass the KBA will be falsely authenticated as a legitimate user.

In the second method of resetting a PIN a user will connect to the call center, make an identity claim (possibly with using their credential) and notify the call center that he or she has forgotten their PIN. The CSP will then invoke an automated process where a new PIN is randomly created for the claimed identity and stored in the database. The new PIN is then printed on paper and physically mailed to the mailing address maintained in the records for that identity.

While physically mailing a new PIN does not immediately reduce the authentication process to only KBA it creates a written version of the PIN that is vulnerable to interception or seizure. Since the strength of PIN-based authentication relies on the secrecy of the PIN creating a written version of the PIN effectively eliminates the systems confidence in that the secret is only known by the individual.

To defeat the authentication mechanism on a system employing the second PIN reset procedure an Attacker would connect to the call center, make an identity claim, and claim a forgotten PIN. The Attacker would then, at any point before the legitimate user completely destroys the paper copy, intercept the new written PIN. With the new PIN in hand the Attacker would connect to the call center, enter the User's identifying credential and new PIN and be forwarded to the KBA phase of the authentication process. Thus the second forgotten PIN procedure reduces the authentication process to KBA and the ability to intercept a piece of paper.

The third, and least common, method of resetting a PIN is to require the User to physically be present at a branch office of the financial institution. In this method an individual goes to a branch office and speaks to an Agent; the Agent verifies the identity of the individual, usually through government issued photo identification and possibly KBA, and allows the individual to set a new PIN.

Unlike the previous two methods which relied on KBA to authenticate the individual's identity the third method relies on government issued photo identification. Thus, the authentication phase can be effectively reduced to the ability of an Attacker to provide adequate photo identification and to pass the KBA process.

It should be noted that in systems that use the number on the face of an issued card as the credential similar procedures exist for systems to replace a lost or stolen credential. In these procedures a user can either request a new card and receive it in the mail or request and receive a new card at a branch office. These procedures open a new avenue of attack for an Attacker who can possibly obtain the entire <credential, PIN> pair from the financial institution itself.

3.2.4 Conclusions

Table 3.3 summarizes the threat resistance of a Financial call center for its inside entities and transmission lines. The information in Table 3.3 assumes protection of the entities and transmissions in the GUAP, however, it does not assume Agent controls in place. As a result, due to the presence of a potentially untrustworthy Agent, there is the potential for attacks at either the RA or Verifier entities and the system can have only moderate confidence in the inside entities.

Table 3.3 – Call Center Threat Resistance for Inside Entities and Transmission Lines

Authentication Process Attacks/Threats	**Resistance**	**Location of Attack**
Eavesdropping	Yes	
Replay	Yes	
Session Hijacking	Yes	
Man in the Middle	Yes	
RA Attacks	No	Registration Agent collecting personal information of new user
CSP Attacks	Yes	
Verifier Attacks	No	Verifying Agent recording Claimant responses during KBA

Table 3.4 summarizes the threat resistance of the outside entities and transmission lines. It can be observed that since the transmission between the User/Claimant and the Enterprise Network in the authentication phase is performed completely in the clear and with no means of the User verifying the call center the authentication process is completely vulnerable to any of the transmission attacks as well as Verifier Impersonation Attacks. Furthermore, since the End Device is a personal telephone various methods of intercepting their communications through the End Device can be performed. Since the entities and transmission lines outside the control of the Financial Call Center have no resistance to any of the attacks or threats the system can not have any confidence that the Claimant indeed possesses the claimed identity.

Table 3.4 – Financial Call Center Threat Resistance for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	**Resistance**	**Location of Attack**
Eavesdropping	No	Between User's telephone and Enterprise Network.
Replay	No	Attacker can replay data captured during previous eavesdropping attack
Session Hijacking	No	Attacker hijacks connection after user passes authentication process
Man in the Middle	No	Attacker sits between user and enterprise network
Verifier Impersonation	No	User tricked into providing <credential, token> pair and KBA information to a fraudulent call center
End Device Attacks	No	Attacker can simply listen in to the User's conversation to obtain KBA information
Data Processing Unit Attacks	No	Listening device installed in phone receiver
Token Threats	No	• Guessing attacks • Out-of-wallet attack • Knowledgeable or friendly attack

Since the system has only moderate confidence in the inside entities and transmission lines and absolutely no confidence in the outside entities and transmission lines, overall, the system can only have a 25% confidence level in the output of the authentication system.

3.2.5 Recommendations

The confidence level could be increased to a 50% confidence level by increasing the confidence in the inside entities by eliminating the untrustworthy Agent attacks in Table 3.3. As was discussed in Section 2.3.2, the threat of these Agent-based attacks can be reduced by not exposing Agents to all in the information necessary to pass KBA. For example, in the enrolment phase, rather than have the personal information that will be used for KBA pass through the Agent to the CSP, the new user can enter this information in secret exactly as he or she does with their PIN. In the authentication phase, the KBA system will randomly generate a number of questions that are a subset of all possible questions; this will prevent the Agent from learning all the information necessary to pass KBA since the next KBA session will have a different set of questions.

Some, but not all, Financial Call Centers already implement controls to prevent their Agents from learning all the information necessary to pass KBA as one of their users. These financial services can claim to have a 50% confidence level in the output of the authentication system.

To increase the confidence level to a 100% confidence level modification must be made to the outside entities and transmission lines to increase confidence in them. Since a Financial Call Center must continue to operate on the PSTN the transmission attacks will always remain possible, meaning that an Attacker will always be able to eavesdrop and intercept the communication between a user and the call center. In order to increase confidence in the outside entities and transmissions it must be the case that if an Attacker is able to capture all the information and data sent by a legitimate user and resubmit it to the Verifier at a later time the Attacker will not be falsely authenticated.

Without installing special cryptographic hardware to establish a secured channel the communication channel between the user and the call center must remain an unsecure channel established over the PSTN. As a result the channel will always be vulnerable to monitoring and thus a token based on static attributes of the "something you know" and "something you have" authentication factors can be easily compromised by an attacker monitoring the communication channel. In order to increase confidence in outside entities and transmission lines the verification must verify dynamic attributes of the token; these dynamic attributes are most easily derived from the token "something you are" factor.

Incorporating voice biometrics as the "something you are" factor requires a user to possess no additional equipment since voice authentication can be performed using a standard personal telephone. Additionally, voice biometrics can incorporate "something you know" authentication as an additional factor, allowing the creation of a two factor authentication system.

Since an attacker can eavesdrop on the voice communication the Verifier must have some means of liveness detection in addition to spoofing resistance. A suitable authentication system to this environment is a text-prompted voice biometric system where the text is a KBA query. In this biometric system the system verifies the static and dynamic attributes of the voice sample as well as the correctness of the response to the KBA query. The presence of the KBA in the system acts as a means of liveness detection as well as providing a second factor making the authentication process a two-factor one. An attacker monitoring the channel would have to capture many different authentication sessions just to be sure to pass the static verification of the KBA but will be unable to pass the verification of the dynamic attributes of the voice samples.

To reduce the threat of session hijack and man-in-the-middle attacks voice verification should occur repeatedly throughout a user's session with the financial service. If at any time during the session it is found that the voice samples being provided by the user do not match the voice print in the database the user should not be allowed to proceed with any transaction. By introducing the repeated verification of the voice sample even if an attacker is able to sit completely in between the user and call center he or she is unable to have any influence on the transactions that occur and can only passively monitor the session.

Table 3.5 summarizes the new threat resistance of the outside entities and transmission lines with the introduction of a KBA text-prompted voice biometric authenticator.

Table 3.5 - Voice Biometric Enabled Financial Call Center Threat Resistance for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	Resistance
Eavesdropping	Yes
Replay	Yes
Session Hijacking	Yes
Man in the Middle	Yes
Verifier Impersonation	Yes
End Device Attacks	Yes
Data Processing Unit Attacks	Yes
Token Threats	Yes

It can be observed in Table 3.5 that by changing the Verifier to a KBA text-prompted voice biometric verifier the authentication process meets the requirements to have confidence in the outside entities and transmission lines. Combined with the previously described confidence in the inside entities and transmission lines this change will allow for a 100% confidence level in the output of the authentication system.

3.3 Web Authentication

3.3.1 Overview

Multiple privacy-critical services, such as banking, credit, loan and health services are now being conducted through the internet. The most common method of the User interacting with these web-based services is through a HTTP-based web browser. Since HTTP was originally unsecured and not suitable for sensitive applications the Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS) were designed to provide channel oriented security. [31] Privacy sensitive web applications most commonly use HTTP over TLS, known as HTTPS, to secure their transactions with a customer. This section will examine the User authentication process in HTTPS.

3.3.2 TLS Handshake Protocol

TLS, specified in [32], consists of two components: the handshake protocol and the record protocol. The Handshake protocol allows the server and client (the User's computer) to authenticate each other and to negotiate cryptographic keys, while the Record protocol uses the keys negotiated in the Handshake protocol to encrypt and authenticate transmitted data. This section will examine the security in the Handshake protocol, and its ability to allow a client to authenticate a server as well as a server to authenticate a client.

Figure 3.8 illustrates the full TLS handshake protocol. It can be observed that a client initiates a connection request to the server through a hello message. The server then responds with its hello message, an optional certificate, some cryptographic keying material and a request for the Client's certificate. The Client will then verify the server's certificate and respond with its own certificate (optional), keying material, and the cipher method to be used in the Record protocol. The server will then verify the client's certificate, if applicable, before setting the cipher method and initiating the TLS Record Protocol.

The exchange of keying information and the cipher to be used allows for the establishment for a secure channel to be used in the Record protocol. The exchange and verification of certificates is used for the authentication of the client to the server and the server to client.

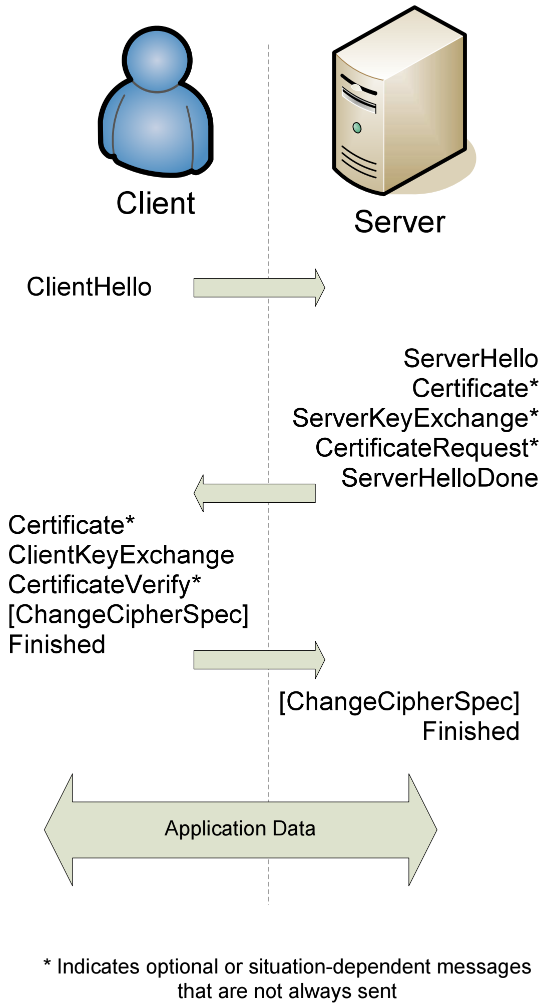


Figure 3.8 - Message Flow for a full handshake

The certificates mentioned in Figure 3.8 are issued by a Root Certificate Authority whose public key is pre-installed in an internet browser. A Client and Server can obtain a certificate from a Root CA for a fee; however, most users do no possess their own certificate. The key exchange in the TLS handshake uses public key encryption, using one of: RSA Key transport, Fixed Diffie-Hellman or Ephemeral Diffie-Hellman key exchange schemes. [32]

In the case where the Client and Server both posses certificates issued by a Root CA, the authentication is mutually authenticated with both entities authenticating each other through the mutually trusted third party. The authentication mechanism in this case is utilizing “something you have” authentication as both entities are demonstrating possession of the third party issued certificate and a corresponding private key.

In the case where the Client does not possess a certificate but the server does the authentication is only one-way with the Client authenticating the identity of the Server. In this situation the Server has no method of identifying the Client through the TLS protocol and must resort to an additional method that will operate after the TLS connection. In order for a Server to authenticate a User who does not posses a certificate the server must first establish a TLS connection with the User before requesting the User to provide a <credential, token> pair, most commonly this pair is a username and password.

3.3.3 Analysis of the Web Authentication Process

Web-based Authentication can be broken into two distinct processes: mutually authenticated TLS, where the Client and Server authenticate each other using the TLS handshake; and one-way authenticated TLS where the Client authenticates the Server using the TLS handshake and the Server authenticates the User through their own authentication process in their application. This analysis will first examine the security in the one-way authenticated TLS process before analyzing the mutually authenticated TLS process.

3.3.3.1 One-Way Authenticated TLS

Figure 3.9 illustrates the authentication process in one-way authenticated TLS. It can be observed that the process is similar to the GUAP except that the entity labeled as the "Enterprise Network" has been issued a certificate by the Root CA and the Data Processing Unit component of the User/Claimant entity has received a copy of the public key of the Root CA. Assuming again the correct operation of the Root CA, the entities and transmission lines inside the control of the authentication process are inside the Enterprise Network" rectangle; while the entities and transmission lines outside the control of the authentication process are those inside the User/Claimant rectangle and the transmission from the User/Claimant to the Enterprise Network.

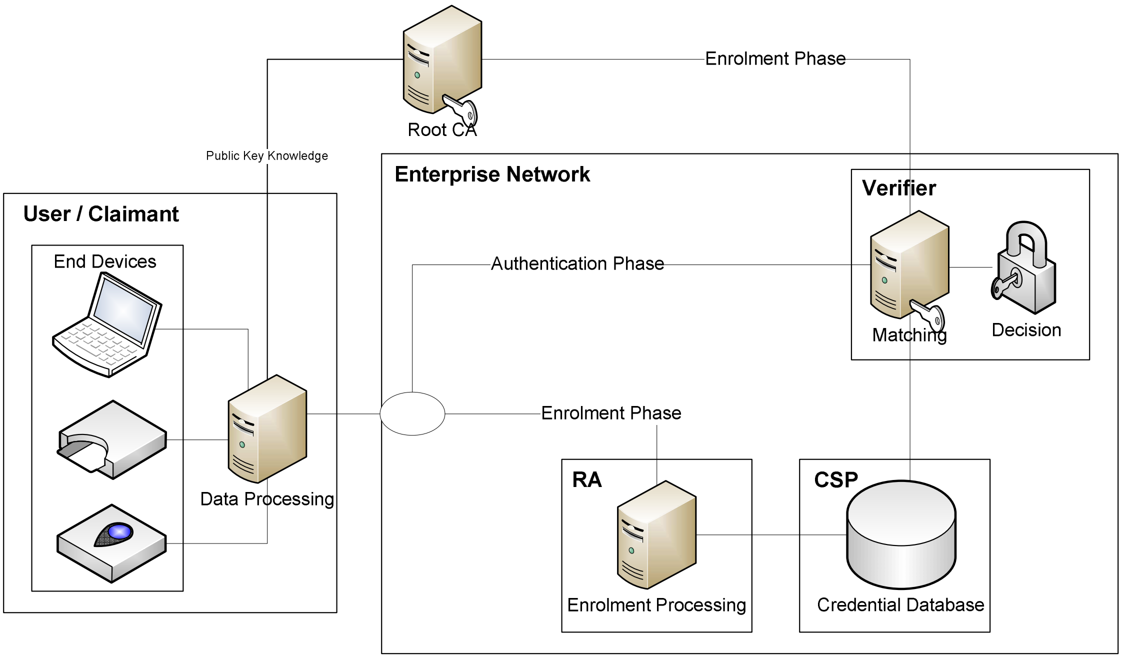


Figure 3.9 - One-Way Authenticated TLS

3.3.3.1.1 Analysis of the Inside Entities and Transmission Lines

The entities and transmission lines inside the control of the authentication system are those entities and transmission lines inside the Enterprise Network rectangle in Figure 3.9. Since this analysis is focused on the general case of web authentication and the security of the components inside the enterprise network is implementation specific this subsection will describe an example implementation that addresses the threats to these components.

Figure 3.10 illustrates an example configuration of the entities and transmission lines inside the control of the Authentication System. In Figure 3.10 the Verifier, RA and CSP entities and the transmissions between them have been combined into a single logical unit. A connection from outside will connect to either the Verifier Interface or the RA Interface.

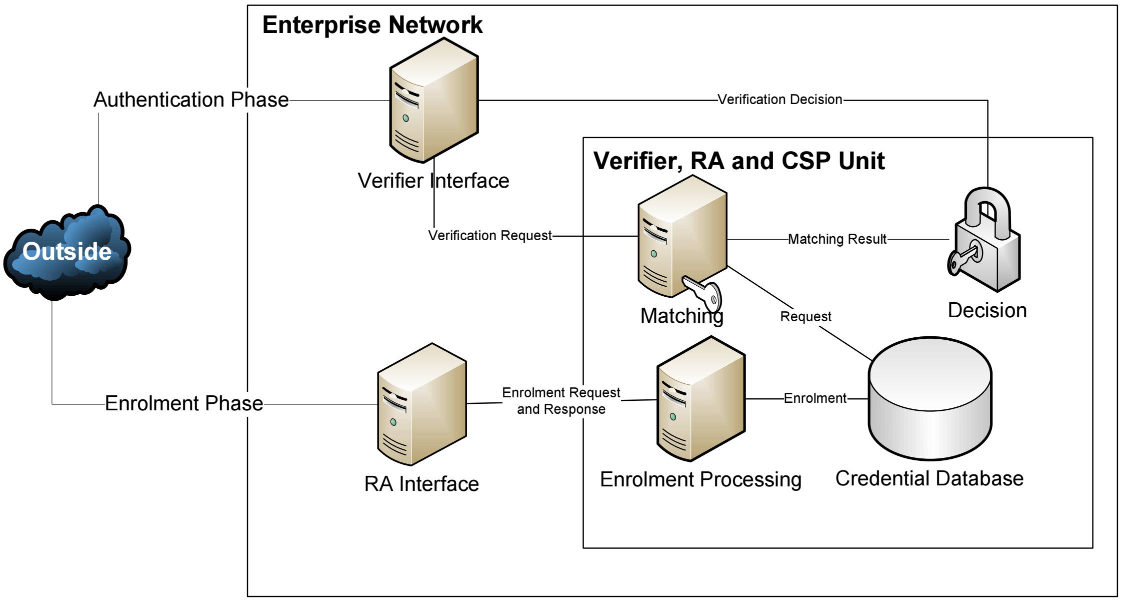


Figure 3.10 - Example Web-Based Authentication Enterprise Network - Inside Entities and Transmission Lines

By combining the Verifier, RA and CSP into a single logical unit the authentication system is ensuring that a <credential, token> pair is never available outside of that unit. A User will register in the system through the RA interface which will communicate with the actual RA inside the combined unit; a Claimant will request verification through the Verifier Interface, which will forward the submitted <credential, token> pair to the actual Verifier inside the combined unit.

The combined unit, through the use of internal encryption and authentication mechanisms is able to ensure that communications are performed only with the system installed interfaces. Additionally, the only information sent out of the combined unit is an enrolment response (successful or unsuccessful) and a verification decision (successful or unsuccessful).

The two interfaces ensure that information submitted by the User/Claimant is valid, meaning that it does not contain malicious code, in addition to ensuring that no outside, unauthorized connection are possible to the combined unit.

Table 3.6 summarizes the threat resistance of this example implementation of the entities and transmission lines inside the control of the authentication system. It can be observed that this example meets the requirements for complete confidence in the inside entities and transmission lines.

Table 3.6 - Web-Based Authentication Threat Resistance for Inside Entities and Transmission Lines

Authentication Process Attacks/Threats	Resistance
Eavesdropping	Yes
Replay	Yes
Session Hijacking	Yes
Man in the Middle	Yes
RA Attacks	Yes
CSP Attacks	Yes
Verifier Attacks	Yes

3.3.3.1.2 Analysis of the Outside Entities and Transmission Lines

The entities and transmission lines outside the control of the authentication system are those inside the User/Claimant rectangle in Figure 3.9 and the transmission from that rectangle to the Enterprise Network rectangle. It is important to note that the transmission from the User/Claimant to the Enterprise Network occurs after the TLS handshake has been established.

Since the transmission between the User/Claimant and the Enterprise Network happens after the TLS handshake, assuming that the TLS Handshake has been implemented correctly, the transmission line between the two entities can be considered secure against eavesdropping and replay attacks. However, very real threats on one-way authenticated TLS Web-based authentication session are Verifier Impersonation attacks, including Phishing and Pharming attacks as well as Session Hijacking and Man-in-the-Middle attacks. While it is true that through the use of TLS the User/Claimant should be able to correctly authenticate the Verifier it has been found that this is not always the case. [33]

In a TLS connection the web application itself, such as the web browser, handles the verification of the certificate and usually will only notify the User if something is wrong. It is the User's responsibility to ensure that the certificate being authenticated is valid.

The generic phishing attack where the Attacker sends a fraudulent email linking to a fraudulent Verifier is a possible attack. In this example the User clicks the link to the fraudulent website and there is no certificate to be verified, in which case the User, who did not check the certificate themselves, has no idea they are interacting with a fraudulent site. The fraudulent site may even have a legitimate certificate that was issued to it by a known and trusted Root CA; however, in this case the certificate would contain information about the certificate owner which would not match the information contained in the certificate belonging to the actual authentic website.

In the generic Pharming attack, similar to the generic phishing attack above, the website may or may not have a certificate. The attack is possible because the User does not manually check the authenticity of the certificate.

Session Hijacking and Man-In-The-Middle attacks are possible if the User does not verify the certificate received during the TLS handshake. In these attacks an Attacker can sit between the actual verifier and the User and act as a relay point for information between the two: the Attacker will establish a TLS session with the User and a TLS session with the web service. Once the authentication process has finished the Attacker can choose to hijack the session at any time and/or view/modify the contents of the transactions.

In the general web application the User/Claimant's End Device is the keyboard attached to their personal computer while the personal computer and the web application operating on it form the Data Processing Unit. Additionally, in the general web application, the <credential, token> pair is very often a <username, password> pair.

The attacks and threats on these entities are similar to those discussed in the Skype evaluation. The Attacker's objective is to somehow intercept the <username, password> pair at some point between the End Device (keyboard) and the transmission out of the Data Processing Unit (web application) and resubmit this pair to be falsely authenticated at a later time. Various attacks can be performed to accomplish the Attacker's objective such as an over-the-shoulder attack or a keyboard logger.

Many web applications also incorporate a "Remember Me" option which will save the <username, password> pair on the User's personal computer; the saved data may o may not be encrypted. An Attacker can reverse engineer the web application and determine how to recover the <username, password> pair.

Table 3.7 summarizes the threat and attack resistance of the entities and transmission lines outside the control of the authentication system. It can be observed that there is no resistance to any of the threats on the authentication process, and thus no confidence can be had that the Claimant possesses the claimed identity.

Table 3.7 - Web-Based Authentication Threat Resistance for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	Resistance	Location of Attack
Eavesdropping	No	Transmission between keyboard and Web Application
Replay	No	Attacker can replay data captured during transmission between keyboard and Web Application
Session Hijacking	No	• An Attacker can take over the User's personal computer after the authentication phase is complete and interact with the web application through the TLS connection • Fraudulent verifier operates between the User/Claimant and the Web Service and hijacks the session at an appropriate time
Man in the Middle	No	• An Attacker can take over the User's personal computer after the authentication phase is complete and interact with the web application through the TLS connection • Fraudulent verifier operates between the User/Claimant and the Web Service
Verifier Impersonation	No	Phishing and Pharming attacks
End Device Attacks	No	• Attacker can perform an "over-the-shoulder-attack" • Keystroke logger
Data Processing Unit Attacks	No	"Remember Me" feature allowing to <username, password> recovery
Token Threats	No	guessing or dictionary attacks

3.3.3.2 Mutually Authenticated TLS

Figure 3.11 illustrates the authentication process that occurs in a Mutually Authenticated TLS connection. It can be seen that Figure 3.11 differs greatly from the GUAP, due entirely to the reliance on a mutually-trusted third party for authentication. In the Mutually Authenticated TLS connection both the Server and Client accept each other's identities if the certificates exchanged during the TLS handshake are verified as valid.

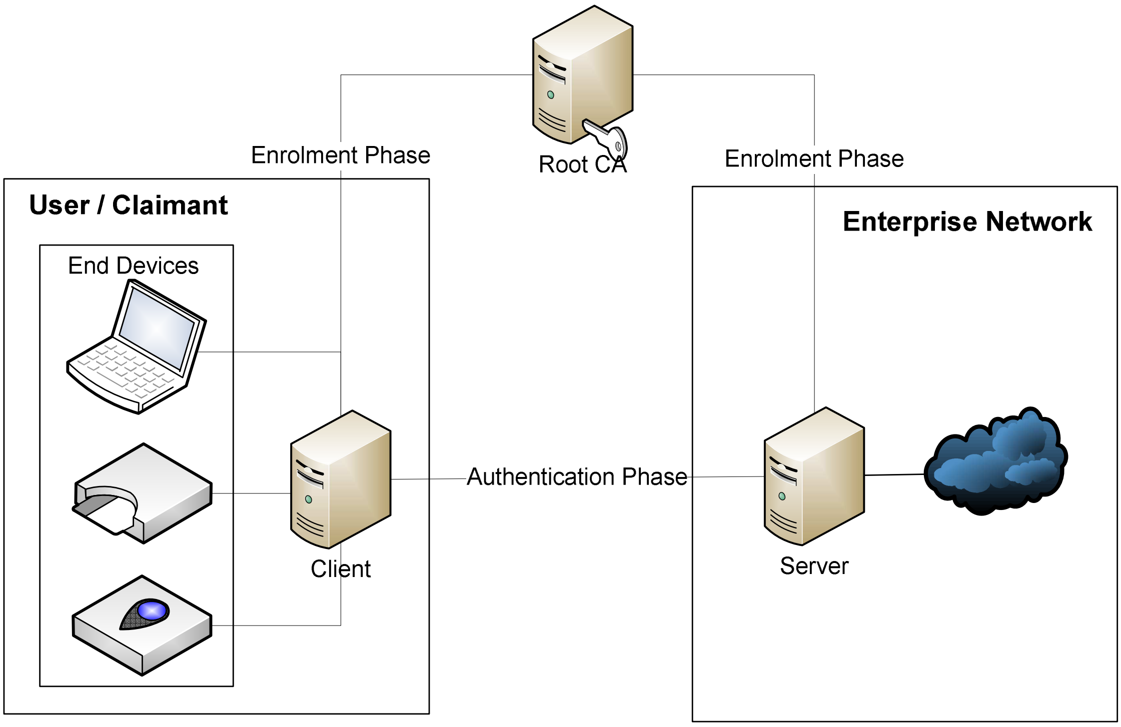


Figure 3.11 - Mutually Authenticated TLS

The Enrolment Phase, seen in Figure 3.11, consists of the Root CA issuing a certificate to both the Client and the Server. It is up to the Root CA to verify the identities of both during certificate issuance. An additional enrolment phase that is not shown is when the User registers for a service offered by the enterprise and makes their identity known to the enterprise.

The authentication phase occurs entirely within the TLS protocol. The Client authenticates the Server's public-key certificate and the Server verifies the Client's public key certificate; verification of the certificates is performed using the public key of the Root CA.

Assuming the TLS protocol is properly implemented on the both the Client and Server the resulting weaknesses are in the components and transmission lines inside the entities labeled as the User/Claimant and the Enterprise Network in Figure 3.11. An additional weakness is the Root CA itself: it must maintain the proper secrecy of its private key in order for the TLS scheme to operate; the remainder of this analysis assumes the correct operation of the Root CA.

Authentication in this process is based on the "something you have" factor. The Client and Server both possess a public key certificate and a corresponding private key. As long as the Client and Server can keep their private key private the TLS authentication is genuine, assuming that public key cryptography is secure and that the Root CA has not compromised its own private key. If an Attacker's objective is to pass authentication at the server as a legitimate user his or her task would be to obtain the certificate and private key belonging to the User; if an Attacker's objective was to fraudulently represent the enterprise the Attacker would have to obtain the certificate and private key belonging to the server. Therefore the confidence in the output of the authentication process depends on the confidence in the secure storage of the private key.

Due to the reliance on the third party the only threat on the inside entities and transmission lines is threat of a Verifier Impersonation attack that is the result of an Attacker gaining the Enterprise's certificate and private key. This attack can be prevented by the Enterprise by ensuring that various checks and measures are in place to ensure the secure storage of their private key. Providing that an Enterprise is able to have confidence in the secure storage of its private key Table 3.8 illustrates that the Enterprise can have complete confidence in its inside entities and transmission lines.

Table 3.8 - Mutually Authenticated TLS Threat Resistance for Inside Entities and Transmission Lines

Authentication Process Attacks/Threats	**Resistance**
Eavesdropping	Yes
Replay	Yes
Session Hijacking	Yes
Man in the Middle	Yes
RA Attacks	Yes
CSP Attacks	Yes
Verifier Attacks	Yes

It can be observed in Figure 3.11 that the entities outside the control of the Enterprise network are the User/Claimant entity and the transmission line from that entity through the internet to the Enterprise network. In order to have confidence that the User/Claimant entity and the transmission lines can not compromise the confidence in the authentication output the enterprise must have confidence that the User's certificate and private key cannot be obtained by an Attacker at any point before, after or during the authentication process.

As in the one-way authenticated TLS the transmission between the User/Claimant and Enterprise Network is encrypted and secure after the TLS Handshake. However, the TLS handshake is susceptible to the same Verifier Impersonation attacks, Session Hijacking and Man-in-the-Middle attacks if the certificates are not manually verified.

While the Enterprise can assume that the User would want to keep his or her private key secret, particularly in a privacy-critical system, the Enterprise has no assurances that it is indeed kept secret. If the trusted third party were to periodically perform evaluations of the User's key storage mechanisms and pass the evaluations and the Enterprise had confidence in the third party's ability to evaluate the secrecy of the private key than there may be a level of assurance available to the Enterprise. However, without such a system, the Enterprise can have no reasonable confidence in the outside entities and transmission lines, as summarized in Table 3.9.

Table 3.9 - Mutually Authenticated TLS Threat Resistance for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	**Resistance**	**Location of Attack**
Eavesdropping	No	Transmission between key storage and web browser
Replay	No	Attacker can reuse private key captured during eavesdropping attack
Session Hijacking	No	• An Attacker can take over the User's personal computer after the authentication phase is complete and interact with the web application through the TLS connection • Fraudulent verifier operates between the User/Claimant and the Web Service and hijacks the session at an appropriate time
Man in the Middle	No	• An Attacker can take over the User's personal computer after the authentication phase is complete and interact with the web application through the TLS connection • Fraudulent verifier operates between the User/Claimant and the Web Service
Verifier Impersonation	No	Phishing and Pharming attacks
End Device Attacks	No	Process that locates stored private keys
Data Processing Unit Attacks	No	Virus exploiting a fault in the web browser recovers private key
Token Threats	No	An Attacker in possession of both private key and certificate will be falsely authenticated

3.3.4 Conclusions

The above analysis assumed the correct operation and security of the Root CA and the TLS protocol is always properly implemented. These assumptions allowed the evaluation of the entities both inside and outside the control of the authentication system.

The analysis found that with properly implemented security considerations that the authentication system could have confidence in the entities and transmission lines inside its control. However, it was found that in both mutually-authenticated TLS and one-way authenticated TLS the authentication system could have no reasonable confidence in the entities and transmission lines outside of its control. Resulting in the conclusion that web based authentication can only have a 50% confidence level in the output of the authentication system. Table 3.10 summarizes the findings and conclusions regarding web-based authentication.

Table 3.10 - Confidence Level in Web-Based Authentication

TLS Scheme	Confidence in Inside Entities and Transmission Lines	Confidence in Outside Entities and Transmission Lines	Authentication Confidence Level (%)
Mutually-Authenticated	Confident	Not Confident	50%
One-Way Authenticated	Confident	Not Confident	50%

3.3.5 Recommendations

The lack of confidence in the outside entities and transmission lines arose from the susceptibility of the authentication system to every attack and weakness considered, despite the establishment of an encrypted channel between the User and the Web Service through TLS. This susceptibility indicates that in order for the authentication system to have confidence in the outside entities and transmission lines rather than adding more security and encryption the entities and transmissions it should be assumed that the Attacker is able to completely capture any data submitted to the Verifier.

With the assumption that the any data submitted to the Verifier by a user could be replayed by an attacker the verification cannot rely solely on either the static attributes of the “something you know” or “something you have” factors and should instead verify a token based on both static and dynamic attributes. Keeping everything else the same in the system but changing the verifier to one that verifies both static and dynamic attributes of a token many of these threats to the outside entities and transmission lines could be removed. To completely remove the threat of the session hijacking and man-in-the-middle attacks the system should also incorporate periodic re-authentication in addition to the verification of dynamic attributes. With these improvements Table 3.11 highlights the new threat resistance in the outside entities and transmission lines.

It is recommended that a text-dependent voice biometric verifier is used in the improved system. Voice biometrics are recommended in this situation as it can easily incorporate both the static and dynamic attributes desired through the secret passphrase being spoken and the way that it is spoken by the user. Additionally, the use of voice biometrics will require little or no additional hardware to the user as the only requirement to use a voice verification system is the presence of a microphone.

Table 3.11 – Improved Web-Based Authentication Threat Resistance for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	**Resistance**	**Improvement to previous attack**
Eavesdropping	Yes	While an Attacker may be able to record the biometric between the sensor and the application on the user's PC due to the verification of dynamic attributes it cannot be replayed successfully
Replay	Yes	See above
Session Hijacking	Yes	Prevented by periodical re-authentication
Man in the Middle	Yes	Prevented by Periodical re-authentication
Verifier Impersonation	Yes	Phishing and Pharming attacks prevented since any data captured from a legitimate user can not be used successfully by an Attacker.
End Device Attacks	Yes	While an Attacker may be able to record the biometric between the sensor and the application on the User's PC due to the verification of dynamic attributes it cannot be replayed successfully
Data Processing Unit Attacks	Yes	Remember me feature removed
Token Threats	Yes	Reduced through liveness detection and spoofing resistance within the Verifier and the verification of both static and dynamic attributes

Chapter 4

Conclusions, Recommendations and Future Work

4.1 Overview

Chapter 1 introduced the notion of User Authentication, defined the GUAP and explored the factors of authentication while Chapter 2 explored the GUAP including its weaknesses and threat and the weaknesses and threats associated with the factors used in authentication tokens. Chapter 2 went on to define measures through resistance to attacks and threats on the entities and transmissions forming GUAP to establish the confidence level in the output of the authentication system. Chapter 3 analyzed several authentication systems and using the measures established in Chapter 2 assigned a confidence level to the output of these authentication systems, and as necessary, made recommendations as to how to improve the confidence level. This chapter will first summarize the findings of Chapter 3 and highlight the failings of the analyzed authentication systems before describing and analyzing an example authentication system that meets the requirements for a 100% Confidence Level in the output of the authentication process.

4.2 Conclusions

Table 4.1 summarizes the authentication confidence level of the analyzed systems as compared to the ideal authentication system. It can be observed that, with the exception of the Financial Call Center without Agent Controls, the analyzed authentication systems all received a confidence level of 50%; The Financial Call Center without Agent Controls received a confidence level of 25% due to its inherent trust in the internal agents.

Table 4.1 - Confidence Levels of Analyzed Authentication Systems Summary

Authentication System	Authentication Confidence Level (%)	Confidence in Inside Entities and Transmission Lines	Confidence in Outside Entities and Transmission Lines
Ideal	100%	Confident	Confident
Skype	50%	Confident	Not Confident
Financial Call Centers – without Agent Controls	25%	Moderately Confident	Not Confident
Financial Call Centers – with Agent Controls	50%	Confident	Not Confident
One-Way Authenticated TLS Web-Based Authentication	50%	Confident	Not Confident
Mutually-Authenticated TLS Web-Based Authentication	50%	Confident	Not Confident

It can be further observed in Table 4.1 that none of the authentication systems have any confidence in the entities and transmission lines outside of their control. Table 4.2 summarizes the resistance to attacks and threats on the outside entities and transmission lines of the analyzed systems.

Table 4.2 - Threat Resistance Summary for Outside Entities and Transmission Lines

Authentication Process Attacks and Threats	Threat Resistance			
	Ideal	Skype	Financial Call Centers (Both)	Web-Based Authentication (Both)
Eavesdropping	Yes	No	No	No
Replay	Yes	No	No	No
Session Hijacking	Yes	Yes	No	No
Man in the Middle	Yes	Yes	No	No
Verifier Impersonation	Yes	No	No	No
End Device Attacks	Yes	No	No	No
Data Processing Unit Attacks	Yes	No	No	No
Token Threats	Yes	No	No	No

All of the analyzed authentication systems have very little, if any, resistance to the attacks and threats on the entities and transmission lines outside of their control. Some of the systems, Skype and Web-Based Authentication, have even taken steps to ensure the establishment of a secured channel before authentication information is exchanged; however, these steps have not protected the authentication process. The inability of the authentication systems to provide adequate threat resistance arises from the ability of an attacker to somehow, at some point during the process, capture the <credential, token> pair and submit the captured pair in a separate authentication session to be falsely authenticated as the owner of the credential.

All of the analyzed authentication systems use a token based on some form or combination of the “something you know” or “something you have” authentication factors and verify only the static attributes of the authentication token. By authenticating only the static attributes in the token the authentication systems merely verify that the Claimant is in possession of the information that comprises the token and if the Claimant can demonstrate as such he or she will be authenticated as possessing the claimed identity. Thus, once an attacker is able to possess the static information contained in the token, or the token itself, the attacker is able to fraudulently pass the authentication process and be falsely authenticated as a legitimate user.

4.3 Recommendations

As previously mentioned the examined authentication systems failed from an inability to operate correctly once the token itself or the static information comprising the token was compromised. The examined systems would verify that a claimant was in possession of some static information that comprised the authentication token and that once an attacker had that information in his or her possession was able to successfully masquerade as a legitimate user.

The security of the examined authentication systems assumed that the static information comprising the authentication token was not obtainable by an attacker once it was out of the direct control of the system. However, it was determined that there exists feasible points of attack in the portions of the authentication process where an attacker can capture the <credential, token> pair itself or at the very least the static information comprising the token. It is therefore recommended that to have complete confidence in the output of the authentication process the design of an authentication system should assume that the entire <credential, token> pair can be obtained by a determined attacker. An ideal authentication system, which has complete confidence in its output, must be able to correctly authenticate a claimant in the face of a compromised token.

Once the assumption is made that the token, or the authenticating information contained within it, can be compromised outside of the direct control of the authentication system it becomes necessary to incorporate dynamic information into the token. Verification of dynamic attributes add an ever changing structure to the information contained within the token meaning that if an attacker was able to capture the token and resubmit it, the Verifier would recognize the token as a previously submitted token and not authenticate the attacker. This ever changing structure was not possible in the examined systems which would verify only static information stemming form the "something you know" and "something you have" authentication factors; dynamic attributes, on the other hand, are more readily available in the "something you are" authentication factor.

Recommendations for improvements to the examined systems were to incorporate the verification of dynamic attributes into the systems with minimal changes to the authentication process. In each recommendation, verification using voice biometrics was recommended as the source for the dynamic attributes. Voice biometrics was recommended based on its many strengths discussed in Chapter 1. The remainder of this section will expand on these recommendations by describing an authentication system which meets the requirements for an ideal system which has complete confidence in the correctness of it output.

4.3.1 An Ideal Authentication System

In this section an authentication system that meets the requirements to have complete confidence in its output will be described. This system will first be described in terms of the entities that comprise the system, the authentication factors used in the authentication token and the steps involved in the enrolment and the authentication phases. A complete analysis of the system will then be performed in order to demonstrate its ability to resist the identified attacks and threats to the authentication process and have complete confidence in its output.

Figure 4.1 illustrates the entities involved in the proposed authentication system. It can be noted the major difference between Figure 4.1 and the GUAP, first illustrated in Figure 1.10, is that the end device in the proposed system consists of a microphone. The proposed system utilizes an authentication token based on voice biometrics and thus the microphone will be the only device the user will interact with in the authentication process.

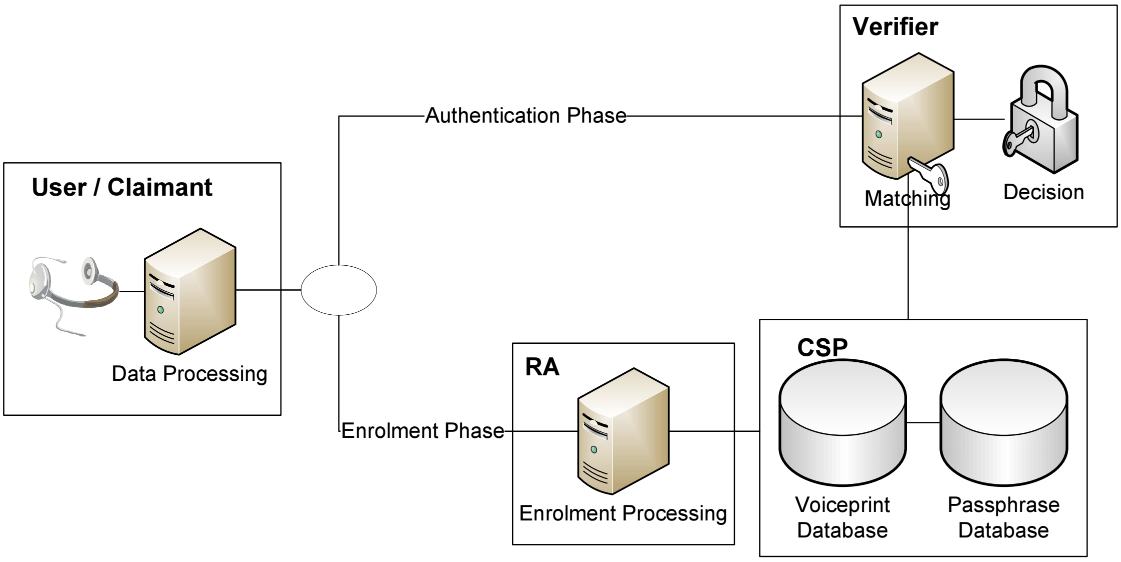


Figure 4.1 - Ideal Authentication System Process

4.3.1.1 Enrolment Phase

Figure 4.2 illustrates the enrolment phase of the proposed system. Once the enrolment phase has been initialized by a system administrator and a user id has been created, the user will provide voice samples through their microphone and the attached Data Processing Unit over a secured channel to the Registration Authority which will compute a voice print from the samples before storing the voice print in the database. The complexity of establishing a voice print, such as the number of phrases and sounds the user must speak before a voice print is created, depends on the nature of the voice biometric verification system employed.

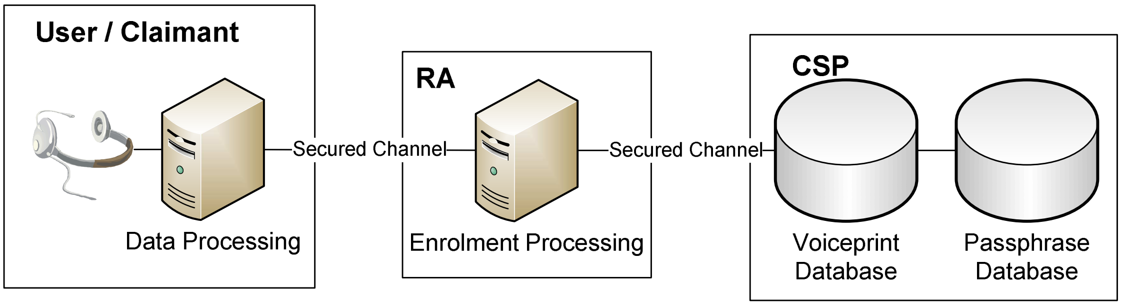


Figure 4.2 - Enrolment Phase

In the proposed authentication system enrolment, however, is not complete once a voiceprint for the new user has been established. Immediately following the establishment of the voiceprint the user will be asked to create and speak an identifying phrase that the user will use in the authentication phase for identification. This phrase will be the user’s secret passphrase, however, instead of typing it through a keyboard the user will speak it when authenticating to the system. The use of this passphrase and the voiceprint combines both the "something you know" and "something you are" authentication factors into the authentication token making the proposed system a two-factor authentication system.

Once the user's voiceprint and their secret passphrase have respectively been recorded in the voiceprint database and the passphrase database the enrolment process is complete. The user now possesses a credential (an assigned user id) and a token (their voice and a secret passphrase) which will be used in the authentication process.

4.3.1.2 Authentication Phase

Figure 4.3 illustrates the authentication phase of the proposed system. In the authentication process the Claimant will make an identity claim by submitting their user id to the system and speaking their secret passphrase through the microphone. The Data Processing Unit will then digitize the analog voice sample and transmit the voice sample over a secured channel to the Verifier. The Verifier will then perform a matching operation on the voice sample; matching the voice sample against both the template voiceprint and the template passphrase and compute a matching score. If the matching score is inside the acceptable range the Claimant will be authenticated as possessing the claimed identity.

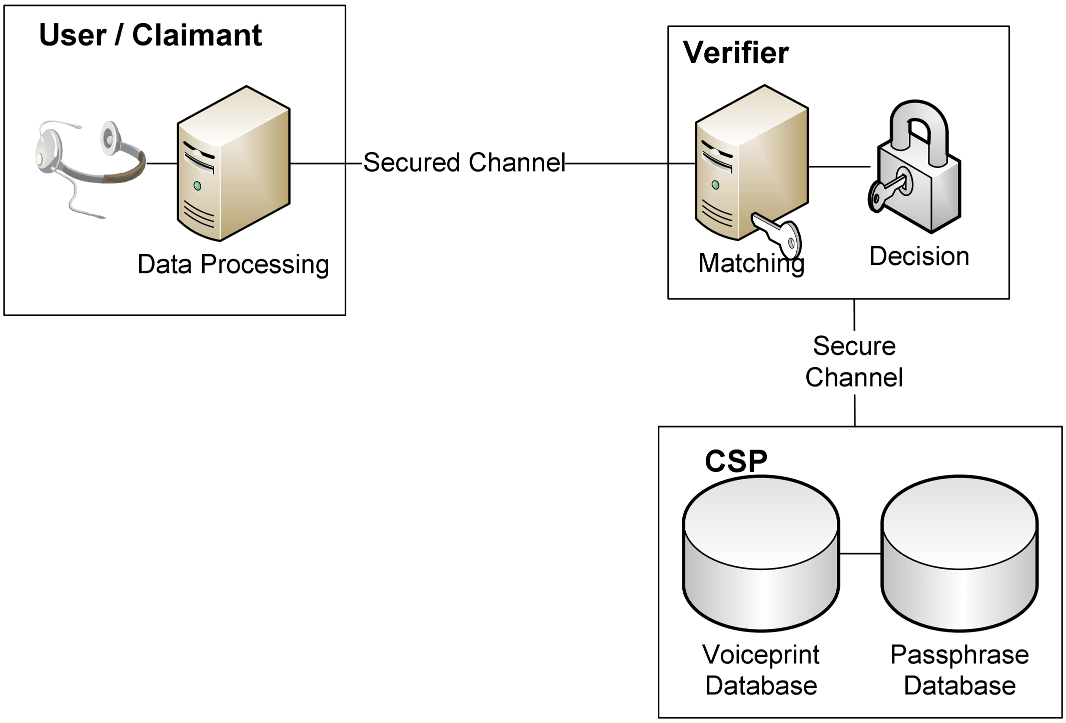


Figure 4.3 - Authentication Phase

The verification that occurs in the proposed authentication system verifies both static and dynamic attributes of the authentication token. The token in this system is the digitized sample of the claimant (user) speaking their secret passphrase. The voice biometric verification will first perform checks on the voice sample to resist spoof attempts through previously discussed methods before performing verification on both static and dynamic attributes of the token. The system will verify the voice sample based on the following static attributes:

- correctness of the passphrase
    - Does the submitted passphrase match the template passphrase?
- correctness of the voice
    - Are the physical traits of the voice sample produced from the voice defined by the template voice print?

If the static attributes are verified as correct, the system will then verify the voice sample based on the following dynamic attributes:

- the waveform of the voice sample does not match a previously submitted sample
    - It is extremely improbable that the submitted sample is identical to a previously submitted sample if it had been spoken for this authentication session. If the current sample matches a previous sample then it is likely that the sample was not spoken by the user during this session.
- unique voice attributes are different than a previous session
    - Human traits such as emotional state and comfort level which can be extracted from the human voice [24] will change between authentication sessions

By verifying the above static attributes the system is able to identify and verify the identity of the claimant and the verification would be sufficient if the integrity of the token was assured. However, as it must be assumed that the token can be compromised, the verification of the dynamic attributes above ensure that captured authenticating information can not be used to pass authentication in a different authentication session.

4.3.1.3 Analysis

Following the same procedure as the examined authentication systems the below analysis of the proposed authentication system first examines the entities and transmission lines that are inside the control of the authentication system before examining the entities and transmission lines outside the control of the authentication system. Following this analysis final summarizing remarks about the proposed authentication system will be made.

4.3.1.3.1 Analysis of the Inside Entities and Transmission Lines

The entities inside the control of the proposed authentication system are labelled in Figure 4.1 as the RA, CSP and the Verifier while the transmission lines inside the control of the system are the transmissions between those entities. An example implementation to minimize threats and attacks on these entities is to combine them into a single logical unit, as illustrated in Figure 4.4.

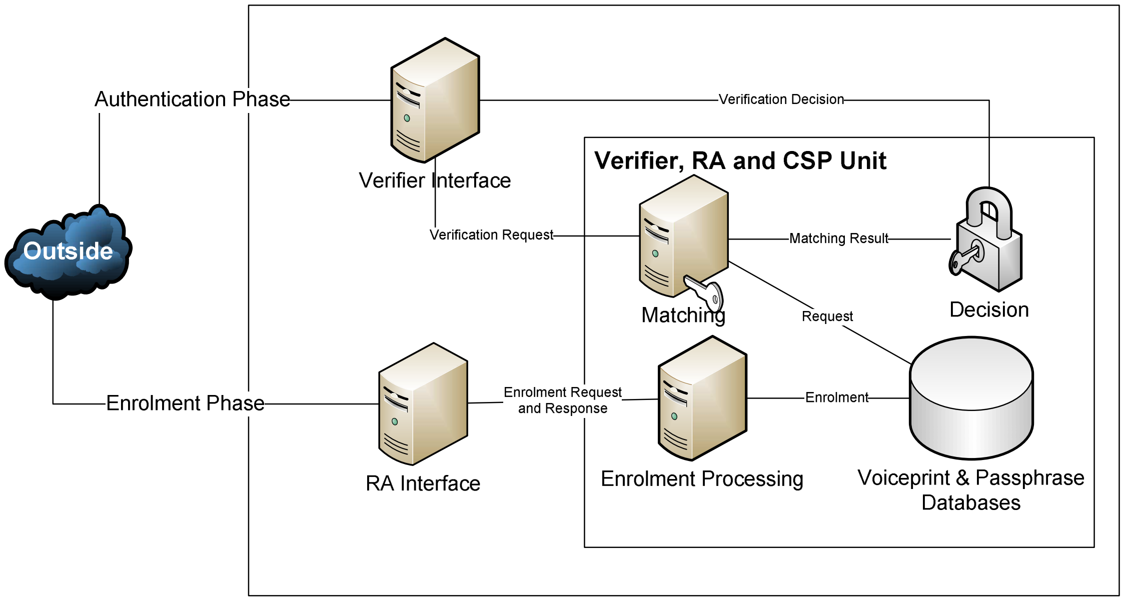


Figure 4.4 - Proposed Authentication System - Inside Entities and Transmission Lines Implementation

The combined unit ensures that template information is never available outside of the unit. A user will register in the system through the RA interface which will communicate with the actual RA inside the combined unit; a claimant will request verification through the Verifier Interface, which will forward the submitted <credential, token> pair to the actual Verifier inside the combined unit.

The combined unit, through the use of internal encryption and authentication mechanisms is able to ensure that communications are performed only with the system installed interfaces. Additionally, the only information sent out of the combined unit is an enrolment response (successful or unsuccessful) and a verification decision (successful or unsuccessful).

The two interfaces ensure that information submitted by the User/Claimant is valid, meaning that it does not contain malicious code, in addition to ensuring that no outside, unauthorized connections are possible to the combined unit. Table 4.3 illustrates that the proposed system can be implemented in such a way as to have complete confidence in its inside entities and transmission lines.

Table 4.3 - Proposed Authentication System Threat Resistance for Inside Entities and Transmission Lines

Authentication Process Attacks/Threats	Resistance
Eavesdropping	Yes
Replay	Yes
Session Hijacking	Yes
Man in the Middle	Yes
RA Attacks	Yes
CSP Attacks	Yes
Verifier Attacks	Yes

4.3.1.3.2 Analysis of the Outside Entities and Transmission Lines

The entities and transmission lines outside the direct control of the proposed authentication system consist of the entity labelled as the User/Claimant in Figure 4.1 and the transmission out of that entity. The components forming the User/Claimant entity are the microphone that the user uses to provide voice samples and the Data Processing Unit which digitizes the voice samples and communicates with the next entity in the process.

In Figure 4.2 and Figure 4.3 the transmission out of the User/Claimant entity to the RA and Verifier entities respectively is labelled as a secured channel. This is an open channel that has been secured through the correct use of cryptographic tools as described in Section 1.2.6. Since the transmission between the User/Claimant entity and the authentication system is performed over a secured channel this transmission can be considered secure against eavesdropping, replay, man-in-the-middle and session hijacking attacks from the definition of a secured channel.

There are a variety of attacks that can be attempted inside of the User/Claimant entity. The transmission between the microphone and the Data Processing Unit is vulnerable to any of the transmission-based attacks and the Data Processing Unit is also vulnerable. An attacker can install a listening device on the transmission line or inside the Data Processing Unit and capture the <credential, token> pair as it is submitted by a legitimate user.

Phishing and Pharming attacks are also feasible. In the proposed system it is still the user's responsibility to ensure that they are correcting to the actual authentication server and not a fraudulent server before providing authenticating information. It is feasible that an attacker utilizing previously described Phishing or Pharming techniques can manage to obtain a <credential, token> pair.

However, due to the nature of the verification of the token a captured <credential, token> pair does not compromise the authentication system. The verification of the token relies on the verification of both static and dynamic attributes of the token: while a compromised token may compromise those static attributes it will not compromise the dynamic attributes.

For example, if an attacker were to capture the <credential, token> pair (the user id and a voice sample) during an authentication session and resubmit the pair the verifier will find that:

- The voice pattern is correct
- The passphrase is correct
- The voice sample waveform matches a previously submitted sample

Since the submitted sample was found to match a previously submitted sample the authentication system will reject the authentication claim and require the legitimate User to choose a new passphrase.

An interesting session hijacking or man-in-the-middle attack is possible through software operating on the Data Processing Unit. In this attack the attacker installs software on the Data Processing Unit which will allow the attacker to take control of the entity at some point. The attacker then waits until immediately following the authentication process (after the legitimate user is authenticated) before taking control of the user's system and hijacking the session; the attacker can then at as the legitimate user through the user's system.

The described session hijacking attack can be prevented through periodic re-authentication in the proposed system. In periodic re-authentication either at random times or before changes are committed by the user the system will re-authenticate the user. In systems where the user interacts with the system through a voice channel (such as a call center) re-authentication can be performed unobtrusively using the user's voice, perhaps even without the user's knowledge, to verify the user's identity.

With the described hijacking attack prevented through the use of periodic re-authentication and the ability of the system to operate correctly in the face of compromised tokens Table 4.4 illustrates the threat resistance of the outside entities and transmission lines.

Table 4.4 - Proposed Authentication System Threat Resistance for Outside Entities and Transmission Lines

Authentication Process Attacks/Threats	Resistance
Eavesdropping	Yes
Replay	Yes
Session Hijacking	Yes
Man in the Middle	Yes
Verifier Impersonation	Yes
End Device Attacks	Yes
Data Processing Unit Attacks	Yes
Token Threats	Yes

4.3.1.3.3 Analysis Summary

Using the information in Table 4.3 and Table 4.4, Table 4.5 illustrates the confidence level that the proposed system as in both the entities and transmission lines inside and outside of its direct control. It can be observed in Table 4.5 that the proposed authentication system meets the requirements for a 100% Confidence Level in the output of the authentication process.

Table 4.5 - Confidence Level of the Proposed Authentication System

Confidence in Inside Entities and Transmission Lines	Confidence in Outside Entities and Transmission Lines	Authentication Confidence Level (%)
Confident	Confident	100%

4.4 Future Work

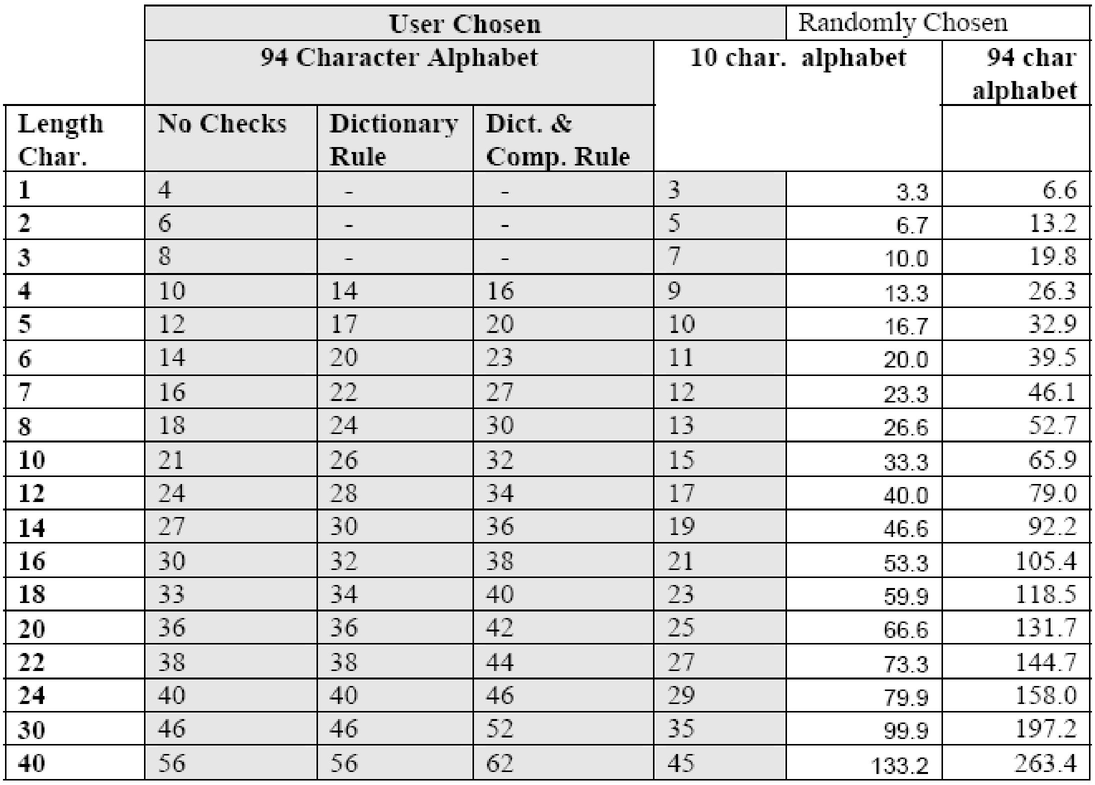
Future work will focus on the continued development of a secure and authenticated VoIP system where the user authentication in the system incorporates the findings of this thesis. Namely, the VoIP system in development will employ a voice biometric system which incorporates both static and dynamic attributes into the authentication token. It is thought that this approach will ensure that the authentication process will be able to resist attacks and operate correctly in the face of a compromised authentication token.

Appendix A
Password Guessing Entropy

Table A.2 was created by NIST [1] and contains rough estimates "of the average entropy of user chosen passwords as a function of length." The logic used by NIST [1] in Table A.1 is as follows for user-selected passwords drawn from the full keyboard alphabet:

- The entropy of the first character is taken to be 4 bits;
- The entropy of the next 7 characters are 2 bits per character; this is roughly consistent with Shannon's estimate that "when statistical effects extending over not more than 8 letters are considered the entropy is roughly 2.3 bits per character;"
- For the 9th through the 20th character the entropy is taken to be 1.5 bits per character;
- For characters 21 and above the entropy is taken to be 1 bit per character;
- A "bonus" of 6 bits of entropy is assigned for a composition rule that requires both upper case and non-alphabetic characters. This forces the use of these characters, but in many cases thee characters will occur only at the beginning or the end of the password, and it reduces the total search space somewhat, so the benefit is probably modest and nearly independent of the length of the password;
- A bonus of up to 6 bits of entropy is added for an extensive dictionary check. If the Attacker knows the dictionary, he can avoid testing those passwords, and will in any event, be able to guess much of the dictionary, which will, however, be the most likely selected passwords in the absence of a dictionary rule. The assumption is that most of the guessing entropy benefits for a dictionary test accrue to relatively short passwords, because any long password that can be remembered must necessarily be a "pass-phrase" composed of dictionary words, so the bonus declines to zero at 20 characters. For user selected PINs the assumption of Table A.1 is that such pins are subjected at least to a rule that prevents selection of all the same digit, or runs of digits (e.g., "1234" or "76543"). This column of Table A.1 is at best a very crude estimate, and experience with password crackers suggests, for example, that users will often preferentially select simple number patterns and recent dates, for example their year of birth.

Table A.1 – Estimated Password Guessing Entropy in bits vs. Password Length [1]

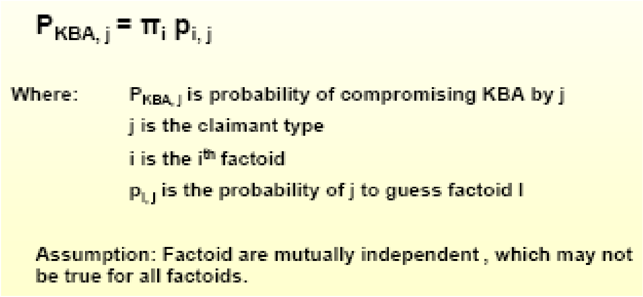


	User Chosen				Randomly Chosen	
	94 Character Alphabet			10 char. alphabet		94 char alphabet
Length Char.	**No Checks**	**Dictionary Rule**	**Dict. & Comp. Rule**			
1	4	-	-	3	3.3	6.6
2	6	-	-	5	6.7	13.2
3	8	-	-	7	10.0	19.8
4	10	14	16	9	13.3	26.3
5	12	17	20	10	16.7	32.9
6	14	20	23	11	20.0	39.5
7	16	22	27	12	23.3	46.1
8	18	24	30	13	26.6	52.7
10	21	26	32	15	33.3	65.9
12	24	28	34	17	40.0	79.0
14	27	30	36	19	46.6	92.2
16	30	32	38	21	53.3	105.4
18	33	34	40	23	59.9	118.5
20	36	36	42	25	66.6	131.7
22	38	38	44	27	73.3	144.7
24	40	40	46	29	79.9	158.0
30	46	46	52	35	99.9	197.2
40	56	56	62	45	133.2	263.4

Appendix B

KBA Guessing Entropy

The guessability of KBA depends on who the guesser is and what the factoid is. Figure B.1 contains a formula for calculating the probability of compromising KBA and Table B.1 which contains actual values was obtained from [22].



$$P_{KBA,\,j} = \pi_i \, p_{i,\,j}$$

Where: $P_{KBA,\,j}$ is probability of compromising KBA by j

j is the claimant type

i is the i^{th} factoid

$p_{i,\,j}$ is the probability of j to guess factoid I

Assumption: Factoid are mutually independent , which may not be true for all factoids.

Figure B.1 – Guessability of KBA [22]

Table B.1 – KBA Guessability Metrics ($p_{i,\,j}$) [22]

	Spouse	Family	Friend	Employer	Professional	Others
Date of Birth	1	1	1	1	1	1 in 18250 or 2-14
Place of Birth	1	1	?	1	1	?
Credit Card	1	10-8 = 2-24	10-8 = 2-24	10-8 = 2-24	10-8 = 2-24	10-8 = 2-24
Home Address	1	1	1	1	1	1
Phone Number	1	1	1	1	1	1
Cell Phone	1	1	1	1 in 16,000 = 2-14	1	1 in 16,000 = 2-14
Mother's Maiden Name	1	1	?	?	?	?
AGI	1	1 in 10,000 = 2-13	1 in 10,000 = 2-13	1 in 10,000 = 2-13	1	1 in 100,000 = 2-17

Tax	1	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1	1 in 10,000 = 2-13
Social Security Number	1	2-15	2-15	1	1	2-15
Bank Statement Balance	1	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1 in 1,000 = 2-10
Credit Card Balance	1	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1 in 1,000 = 2-10	1 in 1,000 = 2-10

Notes to Table B.1:

- Last Pay Stub Information is not a good candidate since the information may not be easily available to the Verifier.
- W2 information could include other information such as state income tax. Gross income, social security tax, Medicare tax are not good candidates since they are guessable from gross income. Even retirement plan deduction will have very low entropy.
- Year of Birth = someone can be assumed to be between 20 and 70 years of age
- Date of Birth = year * days = 50 * 365 = 18,250
- Credit Card guessing = Middle 8 digit
- Home Address Anyone may get from phone book
- Phone Number listed
- Cell Phone Number knows the areas code: No more than 16 exchanges
- Some may know AGI to nearest 10,000. Once that is known, tax may be guessed within 1,000
- AGI Not known but less than 100,000
- Tax not known but less than 10,000
- Social Security Number -- first three digits are based on place of issuance, which are well known and can be guessed based on the place of birth assuming SSN is obtained at or around time of birth. Assumes one check digit. Could be more. Thus entropy more akin to 5 digits = 105 = 215.
- Bank Statement and Credit Card based on balance of up to $1,000

List of Acronyms

AES	Advanced Encryption Standard
CA	Certificate Authority
CSP	Credential Service Provider
DES	Data Encryption Standard
DL	Discrete Logarithm
DNS	Domain Name Service
DPU	Data Processing Unit
EC	Elliptic Curve
ECDLP	Elliptic Curve Discrete Logarithm Problem
EER	Equal Error Rate
FAR	False Accept Rate
FRR	False Reject Rate
FTE	Failure to Enroll
GUAP	Generalized User Authentication Process
HTTP	Hypertext Transfer Protocol
KBA	Knowledge Based Authentication
OSI	Open Systems Interconnection
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PSTN	Public Switched Telephone Network
RA	Registration Authority
RSA	Rivest, Shamir and Adleman Public-Key Cryptosystem
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TTP	Trusted Third Party

List of Cryptographic Functions

Encryption Key	**e**
Decryption Key	**d**
Encryption function	$E_e(m) = c$
Decryption function	$D_d(c) = m$
Hash function	$h(x) = y$
Digital Signature signing function	$S_A(m)$
Digital Signature verification function	$V_A(m,s)$
Digital Signature signing function in a public-key cryptosystem	$S_A(m) = D_{d_A}(m)$
Digital Signature verification function in a public-key cryptosystem	$V_A(m,s) = \begin{cases} true, if: E_{e_A}(s) = m \\ false, otherwise \end{cases}$
Certificate function combining an ID and key into a well-known structure	$f(ID_{Bob}, e_{Bob})$
Certificate generation function	$CERT = D_{d_{CA}}(f(ID_{Bob}, e_{Bob}))$
Certificate verification function	$V_{e_{CA}}(ID_A e_A, CERT_A) = \begin{cases} true, if: E_{e_{CA}}(CERT_A) = f(ID_A, e_A) \\ false, otherwise \end{cases}$

Bibliography

[1] A. Menezes et al. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN 978-0849385230.

[2] C.E. Shannon. "Communication Theory of Secrecy Systems." *Bell System Technical Journal*, Vol. 28, pp. 656-715, 1949.

[3] W. Diffie and M. Hellman. "New Directions in Cryptography." *IEEE Transactions on Information Theory*, Vol 22, pp. 472-492, 1976.

[4] G.B. Agnew, R. Mullin, I. Onyszchuk, and S. Vanstone. "An Implementation for a fast public key cryptosystem." *Journal of Cryptology*, vol. 3. pp. 63-79, 1991.

[5] Niels Ferguson and Bruce Schneier. *Practical Cryptography: Paperback Edition*. Wiley Publishing Inc, 2003. ISBN 0-471-22357-3.

[6] G.B. Agnew, R.C. Mullin, S.A. Vanstone. "An Implementation of Elliptic Curve Cryptosystems Over $F_{2^{155}}$." *IEEE Journal on Selected Areas in Communications*, Vol. 11, no. 5. pp 804- 813. June 1993.

[7] R. Rivest, A. Shamir, L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM*, Vol. 21, pp. 120-126, 1978.

[8] T. ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms." *IEEE Transactions on Information Theory*, Vol. 31, pp. 469-472, 1985.

[9] D. Hankerson, A. Menezes and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlang, New York, Inc. 2004. ISBN 0-387-95273-X

[10] NIST Special Publication 800-63, *Electronic Authentication Guidance*. NIST, February, 2008.

[11] C. Pfleeger and S. Pfleeger. *Security in Computing: Fourth Edition.* Prentice-Hall, 2007. ISBN 0-13-239077-9.

[12] B. Lawler. "Models of KBA." Presented at *KBA Symposium*. Gaithersburg, MD, February, 2004. Available: http://csrc.nist.gov/archive/kba/agenda.html

[13] Yau Wei Yun. "The '123' of Biometric Technology." *Privacy Standards Technical Committee*, 2004.

[14] Simon Liu and Mark Silverman. "A Practical Guide to Biometric Security Technology*." IT Professional*, Volume 3, Issue 1, pp 27-32, 2001.

[15] D.A Reynolds. *Automatic Speaker Recognition: Current Approaches and Future Trends.* MIT Lincoln Laboratory, Lexington, MA, 2001.

[16] C. Summerfield. "Biometric Basics: The Fundamentals of Speaker Verification." Presented at the *Voice Biometrics Conference*, Washington, D.C. May 2007.

[17] J.A. Markowitz. “Voice Biometrics.” *Communications of the ACM,* Volume 43, Issue 9. September, 2000.

[18] D. Miller. “Biometric Basics: The Fundamentals of Speaker Verification.” Presented at the *Voice Biometrics Conference*, Washington, D.C. May 2007.

[19] C. Giordano and F. Mackenzie. “Bell’s Voice Identification Service: Making Privacy Protection More Convenient.” Presented at the *Voice Biometrics Conference*, Washington, D.C. May 2007.

[20] R. Morris and K. Thompson. “Password Security: A Case History.” *Communications of the ACM*, Vol. 22, no. 11, pp. 594-597, Nov. 1979.

[21] E. Spafford. “Observations on reusable password choices.” In *Proceedings of the 3rd Security Symposium*. Usenix, September 1992.

[22] S. Chokhani, (2004, February). “KBA Metrics.” Presented at *KBA Symposium*. Gaithersburg, MD, February, 2004. Available: http://csrc.nist.gov/archive/kba/agenda.html

[23] Q. Xiao. “Security Issues in Biometric Authentication.” In *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*. West Point, New York, 2005.

[24] Kanevsky et al. *Conversational Data Mining*. U.S. Patent 6,665,644 B1, Dec. 16, 2003.

[25] OMB Memorandum M-03-22, *OMB Guidance for Implementing the Privacy Provisions of the E-Government Act of 2002*, September 26, 2003. Available: http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf

[26] S. Kerner. “VoIP to Hit 12.1 Million US Households by 2009.” *The ClickZ Network*, Oct. 7, 2004. [Online] Available: http://www.clickz.com/showPage.html?page=3418651

[27] “VoIP Usage Increases, But US Businesses Not Ditching Traditional Phones According to In-Stat.” *Business Wire*, April 11, 2007. Available: http://findarticles.com/p/articles/mi_m0EIN/is_2007_April_11/ai_n27198771

[28] “Skype Official Website.” *Skype Technologies,* S.A., Luxembourg, 2008. Available: http://www.skype.com/

[29] S. Basset and H. Schulzrinne. “An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol.” *Columbia University*, NY, September 15, 2004. Available: http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2004/cucs-039-04.pdf

[30] Tom Berson. “Skype Security Evaluation.” *Anagram Laboratories*, Palo Alto, Ca, October 18. 2005. Available: http://www.anagram.com/berson/abskyeval.html

[31] IETF, *RFC 2818*, “HTTP Over TLS.” May 2000. [Online} Available: http://tools.ietf.org/html/rfc2818

[32] IETF, *RFC 5246*, “The Transport Layer Security (TLS) Protocol Version 1.2.” August 2008. [Online] Available: http://tools.ietf.org/html/rfc5246

[33] B. Schneier. *Secrets & Lies: Digital Security in a Networked World: Paperback Edition.* Wiley Publishing Inc, 2004. ISBN 978-04711453802.