DATA TRANSMISSION IN BIOMETRICS OVER THE INTERNET

Johann Siau and Aladdin .M. Ariyaeeinia

University of Hertfordshire, UK
J.Siau@herts.ac.uk, A.M.Ariyaeeinia@herts.ac.uk.

ABSTRACT

Reducing the bandwidth consumption on the network traffic has been a major concern over the past many years. With the advent of biometrics over the Internet, this issue will become more apparent as the performance of such system relies heavily on the network performance, reliability and security.

This paper presents investigations conducted into the transmission of data over the network for the purpose of biometrics-based recognition. For the benefit of this study, an appropriate client/server architecture has been designed and implemented in software. The paper also presents discussions on such fundamental issues as whether to transmit raw biometrics data and/or biometrics features, the implementation of encryption/decryption algorithm and its effect to the relative performance of a network is discussed and compared in this paper.

1. INTRODUCTION

The predecessor of the Internet [1] was called Advanced Research Projects Agency Network (ARPANET) [1]. This was created by the US government in the 1960's as a defense network with no single point of failure. ARPANET was then expanded to other countries to become 'a system of interconnected networks' and subsequently became known as the Internet.

The limitation of Internet is that it was not originally designed with security in mind. Rather, it was designed as a means of sharing information. As the Internet and its use evolved, there appeared many security implications and bandwidth issues that posed threats to any system relying on the Internet as a communication medium. With the advent of biometric identity verification over the Internet, it is felt that the issues of security and network performance need to be tackled more effectively [2].

In general, the network performance varies widely with the geographical location of the clients, server type, and network resources. There is variation in the response time from session to session even if the connection is made to the same server. This is because in each session, the data packets may travel through a different route [3]. There is a difference in the performance of the dial-up Internet service, Integrated Subscriber Digital Network (ISDN), Asymmetric Digital Subscriber Line (ADSL), Cable Modem and Leased Line as they all have a different bandwidth and response time. This will undoubtedly affect the performance of a biometric verification system in terms of speed, reliability, and the quality of service.

The following sections detail an analysis carried out to determine the right balance in the transmission method for the purpose of implementing applications involving biometric verification. Due to the time limitation, these tests were conducted in different geographical locations within the UK However, most of the Local Area Network (LAN) tests were carried out in the premises of the University of Hertfordshire.

2. BIOMETRICS APPLIED

The raw biometric data can have different sizes depending on its type. For instance, voice or face biometric datasets are considerably larger than that of fingerprint. In any case, the data contains the identity of an individual and should be treated with utmost care. Therefore, it is necessary to have an appropriate architecture and method of transmission in order to provide a high level of protection against uncertainties.

2.1 Client/Server Architecture

An effective client/server structure realisation for biometrics on the Internet has recently been proposed by the authors [2]. This realisation (Figure 1) consists of 3 distinct components, each performing a specific task. The client part consists of users (clients) requesting appropriate services from the server. A main role of the server is to respond to these requests. However, from time to time, itself becomes a client to the central database and requests services from it.

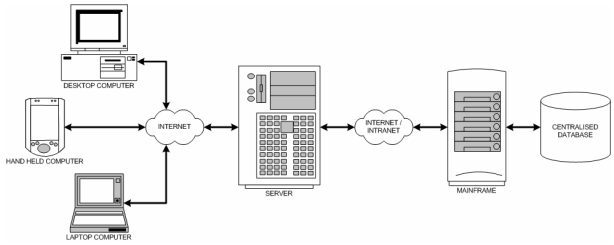


Figure 1: **Client/Server Architecture**

The modular nature of the proposed structure is also necessary for performing software updating effectively. For example, the client module dynamically obtains information relevant to its process, and the updates to its software are provided by the server. As a result, it is ensured that the client software will

always be up-to-date, and modifications or improvements can be gradually rolled in.

In order to maintain data integrity, the transmission channel needs to be secured and encrypted. This will ensure that data sent from the client to the server and vice versa will be of no use to others even though the system is breached.

Figure 2 illustrates the operation of the proposed system in terms of its enrolment and verification processes. It should be noted that although the system is ideally suited to Speaker Verification, it could also be adapted to suit other types of biometrics. The operation can be described as follows.

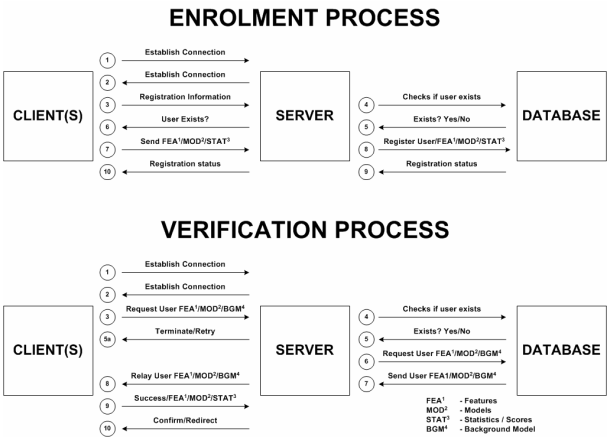


Figure 2: **Proposed Client/Server Architecture**

The database acts as the central storage area for all biometric data and also as a server to the main server. Each server has its unique identifier that allows its connection to the database. All communications between the server and database is secured and encrypted. Distributed/different servers from different geographical location can therefore connect to the central database through a fast network link.

During the enrolment process, the client initially establishes a connection with the server. This is known as the handshaking process in which the client and server establish the identity of both machines for that particular session. The encryption key (section 2.3) is also exchanged at this time. The registration information is then sent to the server. Once a confirmation is obtained from the server that the user does not exist in the system, the client is prompted to send the biometric features, models and statistics over to the server to be enrolled. These are encrypted before transmission. The server then forwards this information to the database and thus enrolling the user to the system.

When a user returns to verify his/her identity, the client machine establishes a connection with the server, whereby during the handshaking process, a different key will be allocated to secure the connection for the session. The client then requests the server to provide data files associated with the user. Server then requests the relevant information from the central database and relays the data back to the client. Client machine uses this information to perform a verification test. If the test result is positive, the statistics regarding the success of the verification is sent back to the server to be stored into the central database.

Depending on the level of security required, the function of the client machine, and the location of the client machine, some operations can be adapted to optimize the performance-to-security ratio appropriately. For example, when a home PC is used, the data files can be stored on the local computer for later use. This will result in reducing the amount of data transfer necessary between the client and the server. However, when the client uses a station which is not registered as his/her own, then the data files provided by the server will need to be removed from the client station after each process is completed in order to improve the security measures.

2.2 Data Format

As in most client/server architectures, a set of instructions is needed to allow communications between the client software and the server software. The instructions for the system, follows a format similar to that shown in Fig 3. The start tag contains one of control, data, or key tags as appropriate for the correct operation of the system.

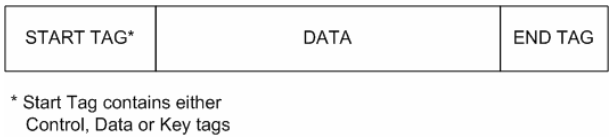


Figure 3: **Data Format Tags**

It is worth noting that the biometric information transferred across should be in the form of characteristic features rather than raw data. This will significantly reduce the size of the data to be transferred. Moreover, with this approach, the load on the server can be significantly reduced by performing considerable parts of the processing on the client machine.

2.3 Data Security

The transmission of data over the network requires some form of security measure. Sensitive data such as biometrics needs to be encrypted to prevent others from misusing it. Therefore, the link between the client and server architecture has to be indefinitely secure throughout the entire process to prevent access or attacks from unknown source.

To secure the link between the client and the server effectively, the data transmitted between them needs to be in encrypted form. Encryption is a process of disguising/ciphering a message which hides its contents by representing it in a different form. For the purpose of decryption, the exact key used for the encryption process will be needed to restore the original message. Without knowing the key, it will be practically impossible to access the message contents. This process is summarized in Figure 4.

A well known algorithm for encrypting and decrypting messages is Blowfish [4]. This algorithm is in the public domain and is considered for the purpose of this study. A main advantage of Blowfish is that it is significantly faster than Data Encryption Standard (DES) [5]. A description of Blowfish is presented in the following section.

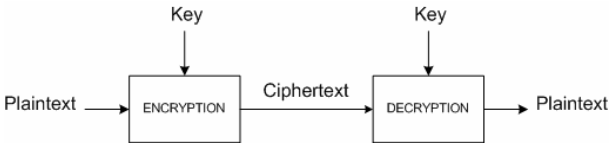


Figure 4: **Encryption/Decryption Process**

2.4 Blowfish

Blowfish is a 64-bit block cipher, and the algorithm consists of two parts. These are a key-expansion part and a data-encryption part. Key expansion converts a key of at most 448 bits into several sub-key arrays in total of 4168 bytes. The plain data is then encrypted via a 16-round Feistel network, where each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

Blowfish uses a large number of subkeys for encryption or decryption and these keys must be pre-computed before any of the above processes can be carried out. The generation of the subkeys involves two arrays consisting of

Eighteen 32-bit P-arrays subkeys, $P_1 \cdots P_{18}$

and

Four 32-bit S-boxes with 256 entries each.

The calculation of the subkeys is detailed in Schneier's paper [4]. In general, generating the subkeys is a computationally expensive process and requires a total of 521 iterations. However, these keys can then be stored and reused.

3. THE ANALYSIS

The most common connection to the Internet is normally via a dial-up service which ideally offers a maximum transmission speed of 56kbit/s. However, cable/ADSL services are becoming more and more available in the UK. In an ideal situation, these offer services with transmission speeds of up to 1Mbit/s downstream (receiving data) and 512kbit/s upstream (sending data). However, the most common transmission speeds of these for receiving and sending data are 512kbit/s and 256kbit/s respectively. It should also be noted that these transmission rates may vary considerably during a given connection.

3.1 Theoretical Approach

The basic approach in calculating the time taken to transmit a file from one location to another via the Internet can be done using the following equation

$$T_s = \frac{Fsz \times 8}{Cnx} \quad (1)$$

where, T_s is the time taken in seconds

Fsz is the file size in bytes and

Cnx is the connection speed in bits/sec

The above equation assumes an ideal situation where the connection to the Internet and to the destination servers is achieved at the maximum throughput. This however, is not the actual case on a day-to-day basis.

A comparison of the calculated transmission time for different file sizes and different connection types is presented in Table 1.

Connection	File Size (bytes)						
	87k	130k	173k	216k	259k	302k	345k
Dial-up 56k	12.43	18.57	24.71	30.86	37.00	43.14	49.29
Cable/DSL 512k	1.36	2.03	2.70	3.38	4.05	4.72	5.39
Cable/DSL 1M	0.68	1.02	1.35	1.69	2.02	2.36	2.70
LAN 10M	0.07	0.10	0.14	0.17	0.20	0.24	0.27
LAN 100M	0.01	0.01	0.01	0.02	0.02	0.02	0.03
LAN 1G	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Time (s)						

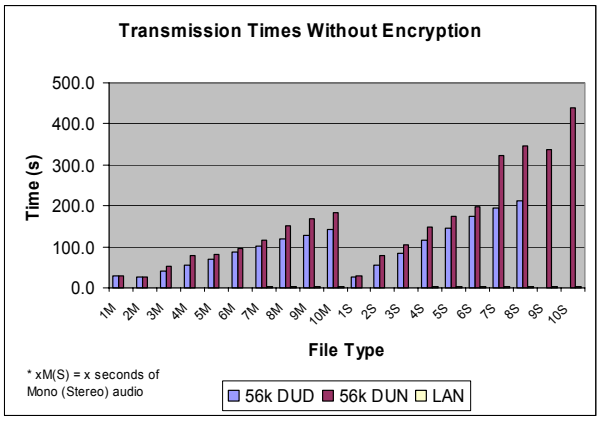
File Size (bytes)							
388k	431k	517k	603k	690k	776k	862k	1024k
55.43	61.57	73.86	86.14	98.57	110.86	123.14	146.29
6.06	6.73	8.08	9.42	10.78	12.13	13.47	16.00
3.03	3.37	4.04	4.71	5.39	6.06	6.73	8.00
0.30	0.34	0.40	0.47	0.54	0.61	0.67	0.80
0.03	0.03	0.04	0.05	0.05	0.06	0.07	0.08
0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01
Time (s)							

Table 1: **Time/File Size/Connection table**

As observed in this table, even in an ideal situation, the use of a dial-up connection involves relatively long transmission time.

3.2 Experimental Result

Experiments were conducted at different times using two types of common Internet connections with the file size varying from 4kb to 900kb. The files used were generated from white noise. These audio files were of 1 to 10 seconds in length. The two types of connection used were a 56k dial-up connection service and a Local Area Network (LAN). The results of this experimental study are given in Figure 5. As it is observed, the transmission time in practice is significantly longer than that suggested theoretically.



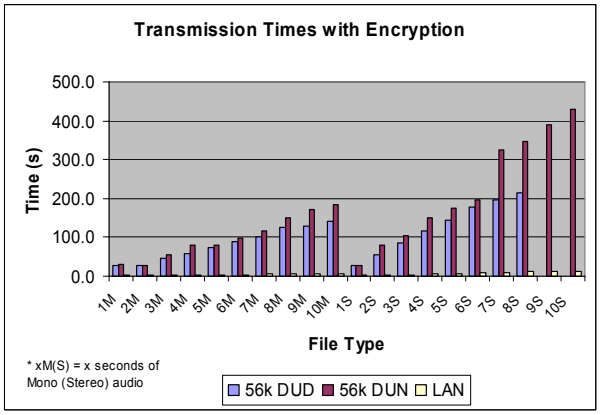


Figure 5: **Test results**

The results in Figure 5 clearly indicate that the verification over the Internet is unfavourably influenced by the performance of the network. To minimize this, it seems advantageous to compress data before its transmission.

The next set of experiments was based on the transmission of audio models rather than raw data. The previous set of white noise files was pre-processed and the features were extracted using LPCC-12. These were used to generate audio models based on a VQ with a codebook size of 64. The results of this study are presented in Table 2. As observed, due to the use of VQ, considerable reduction in the file size is achieved. This in turn has resulted in significant reduction in transmission time.

LPCC12VQ64	File Size (bytes)	Time (s)	
		Without Encryption	With Encryption
56kDUP	4k	1.9	2.3
56kDUN	4k	2.6	2.7
LAN	4k	0.1	0.17

Table 2: **Transmission of Features**

As part of this study, a second set of experiments was conducted based on the encryption of VQ files using the Blowfish algorithm. The results of this investigation are also shown in Table 2. It is seen that there is a slight increase in the overall transmission time in this case. This is due to the initial processing time needed to prepare the data prior to transmission, and the time taken to decrypt the data at the receiver. The resultant increase in the overall transmission time is negligible and often not noticeable.

These experimental results indicate the difficulties introduced by the transmission of raw data over the Internet, especially when the file sizes are too large. The results presented were based on the use of audio files. It should be noted that image-based biometric data files are of considerably larger sizes. The transmission of such raw files over the Internet may result in unacceptably long delays in the verification process.

4. CONCLUSION

A client/server architecture for biometric verification over the Internet has been proposed and described. The discussions have included an analysis of the transmission of biometric data. Based on the experimental investigations, it is shown that, in practice, it may not be feasible to transmit raw biometrics data over the Internet, as this involves unacceptably long delays in the process. Through a set of experiments, it has been demonstrated that the transmission of data models (or features) instead of raw material will significantly reduce the transmission time. It is also argued that the client-server link should be made secure by encrypting the data before its transmission. It is shown that the increase in the overall transmission time due to this process is relatively small.

5. REFERENCES

1. J. Abbate, “Inventing the Internet”, Inside Technology.2000.

2. Johann Siau, Aladdin M. Ariyaeeinia, “Biometrics over the Internet”, COST275 Technical Meeting, INST Paris, April 2002 (http://www.fub.it/cost275)

3. David M. Piscitello, Bellcore and A. Lyman Chapin, BBN, “Introduction to Routing” Connexions Magazine Volume 7, No. 9, Sept 1993.

4. B. Schneier, “Fast Software Encryption”, Cambridge Security Workshop Proceedings, 1993, Springer-Verlag, 1994, pp. 191-204.

5. National Bureau of Standards, Data Encryption Standard, U.S. Department of Commerce, FIPS Publication 46, Jan 1977.