Network Security Issues Case Study : Secure Talk

Mark Arnold
Yuliang Zheng

The Centre for Computer Security Research
Department of Computer Science
The University of Wollongong
Wollongong, NSW 2522, AUSTRALIA
E-mail: {arnold,yuliang}@cs.uow.edu.au

7 July 1994

Abstract

As technology rapidly changes and computers become ever more prevalent network security with the associated problems of authentication, privacy and data integrity become more of a focus. The authentication problem involves or establishes that an identity of a party usually another user is indeed genuine. These problems of security involve issues such as channel monitoring as well as interception and modification of data by unauthorized parties. Along with various other solutions to the above problems, a case study looking at the Secure Talk program together with an examination of some other alternatives is included.
Secure Talk is a means of achieving secure communication through authentication and encryption and as a possible implementation of some of the ideas proposed to solve the authentication, privacy and data integrity problems.
Finally a discussion of the issues is included with regard to the problems and solutions presented with various suggestions being made based on this examination.

1 Introduction

The use of Computing facilities and utilization of network facilities has become very popular among researchers and academics, the powerful, sophisticated access that is provided to specialized data and personal communication has increased the pace of scientific research enormously.

Theoretically people at any site a chain of networked machines or institutions with access to hardware and network media could potentially monitor or store this data for later examination, and as a result this poses various threats to privacy and security of data over the network. However, the sheer volume and general 'noise' inherent to data makes these scenarios highly improbable and impractical to examine the enormous volume of data, and monitor every transmission.

Communications can be vulnerable to 'traffic analysis', a technique or process where the content of messages is not uncovered but information regarding the source and destination addresses can effectively suggest its meaning. For example, knowing that certain people are on certain mailing lists exposes their interests.

In this paper a presentation is made of various threats to a computer network, which is followed by a look at some solutions. A case study looking at the Secure Talk Communication Package is presented, followed by some other schemes for authentication and security. The final three sections deal with a discussion of the problems and solutions, as well as suggestions and finally a conclusion.

In section 2, various possible threats to security are introduced, these being 2.1 authentication and the various forms of data security 2.2. Section 3 looks at ways which these problems

can be solved leading to one possible solution in the area of communication. Section 3 also is a case study looking at this solution, the Secure Talk Package. In the following subsections 4.1, 4.2, 4.3 alternative solutions are examined these being the PGP system for public key encryption, Kerberos, and Kuperee both authentication schemes. Finally sections 5, 6, and 7 provide discussion looking at the benefits as well as the drawbacks on implementing network authentication and security, followed by suggestions with regard to governmental policy decisions and some concluding remarks are made.

2 Security Problems

There are a number of areas of concern when accessing, communicating or transferring data over a network, when wishing to do so users need to be satisfied that certain criteria hold. These criteria being based upon authentication, privacy and data integrity.

The Authentication criteria involves ensuring that the other parties identity is genuine, this is of importance for without authentication sensitive or private data could not be transferred or communicated across a network with the complete assurance that its destination is the intended receiver.

Authentication attempts to foil one of the goals of an attacker, this being to transfer or communicate with a receiver posing as an authorized party or transmitter.

The Privacy criteria is that the data transfer or communication is not subject to examination by others. In the situations where this criteria is breached an attacker will be able to have access to the contents of other users data as transmissions are made via the network.

The Security aspects or criteria of a network involve the looking at the problems of channel monitoring or traffic analysis as well as the added problems of interception and modification of data.

The privacy and security aspects are other attempts to combat typical goals of an attacker, these being determining the contents of transmitted data and the modification of data while the data is still seen as originating from its creator.

An attacker can do a number of things while having access to a communication channel, these may include

1. Blocking flow of information within the channel.
2. Record information to be replayed during another transmission at a later date.
3. Alter or tamper with the transmission, inserting, deleting or modifying data.

Due to the fact that the data can be subject to these attacks assessment needs to be made of the transmission to determine if

1. The message genuinely originated from its sender.
2. The message has not been altered by an interloper.

Channel monitoring involves the attacker looking at ways which a channel can be scrutinized so communication or data can be accessed with its source and destination noted. The interception and modification of data stem from the attacker being able to access and in cases modify data so that it may become incorrect or it may be intercepted and removed hence it will not reach its destination.

The combined problems of Authenticity and Security pose many ways in which an attacker can gain access to the data or at least obtain information pertaining to its source and destination. These are the main areas where security of a network may be compromised and open to attack or unauthorized access.

2.1 Authentication

With the remote access available when using current computing technology the challenge of providing protection and providing user identity takes on many forms. Through successful authentication a party is guaranteed that the person at the destination is in fact the user who is wished to be communicated with.

The authenticity problem takes on another form as well as simply user authentication, this being message authentication. User authentication can include checking physical features of the user such as fingerprints or voice identification, and can also be done by checking information known to the user an example being a password. When user authentication is checked in this way it is also message authentication.

Authentication requires some information which is secret to each user such as a password or a secret key and the appropriately generated public

key, hence these methods are based on public key cryptosystems.

The use of public key cryptosystems to solve the problem of authentication brings up an added problem of key distribution and certification. The problem of key distribution is making sure that the other parties public key is correct and has not been substituted for a public key of an impostor, by nature a public key is obtainable by all users connected to the computer network and as a result is open to tampering or modification.

To rectify the public key distribution problem a third party needs to be involved which both communicating parties are able to trust. Through the trusting of the third party the users are able to trust each other.

When a user needs a public key of another user a request is sent to the trusted third party, usually a server, and the public key is obtained. When a user creates a private and public key pair the server must be notified to keep a record of the correct public key. This means that the public keys obtained are really that of the other party in the communication.

2.2 Data Security

This encompasses many threats where an unknown party can monitor a channel by some means which may lead to the problems of Interception and Modification of data.

When a foreign party is able to monitor a communication or data channel security is severely compromised as data transmissions are available for scrutiny by the unauthorized party, this scrutiny revealing the source and destination of the transmissions.

Channel monitoring can take many forms ranging from a hardware wire taps to code which continually monitor serial ports on a computer to detect data transfer or communications. This type of action may be hard to detect as no data is altered or intercepted as the only side affect may be small deterioration if at all in the transfer rate of the data which may be impossible to detect.

By monitoring the transfer of data across an insecure channel if the source and destination can be determined it may be possible to imply or speculate about the nature of the data.

When an intruder is able to monitor and intercept data on a network this is a major threat to security. The data may be classified as secret, confidential, or simply of a private nature, contents of this data therefore should not be generally accessible to unauthorized persons. While it may be difficult to detect if data has been intercepted and even harder to stop this interception due to the size and number of channels in a modern network, mechanisms or transformations such as encryption need to be developed to make the data is meaningless to all parties other than the intended destination.

As with the threat of channel monitoring this has the added threat that an attacker can access data either removing it hence it will never reach its destination or the interception and access to unauthorized data which would otherwise not be possible. To detect this the party from which the data has been sent should receive confirmation when the data reaches its destination through some form of transfer protocol or handshake prior to or on the completion of the data transfer. This would assure that the data at least reached its intended destination, but cannot assure that other users did not have access to the transmission.

If an attacker is able to intercept data on a network and is able to tamper with this data before it reaches its final destination this poses a serious problem to the integrity and privacy of this data. Data could be modified so as to change its meaning or value by alteration of a few bits or could be substituted for totally new data in either case dire consequences could result. The receiver needs a way to determine if the data has been altered so as to make a decision about its usability. A method is to sign the transmission and then encrypt the data to make it unintelligible the attacker now is unable to make sense of the data due to the fact that it is encrypted, and if this data is modified it can be detected through verification of the signature. If the signature indicates that the data had been tampered with it could be re-sent at a later date the attacker being none the wiser.

3 Secure Talk

The simplest form of securing transmissions over an insecure channel, is to simply use a classical transformation to to encrypt the data with a common secret key known only to the communicating users. A form of checksum could also be appended to the transmission to detect modification. This would make the transmission

unintelligible to all but the sender and receiver, however this method is impractical due to the large number of users needing distinct keys and a secure means is needed by which to distribute these keys, so other solutions need to be sought.

An alternative to the single key transformation as a solution to the authentication and security problems is based on public key cryptosystems. This is where each party has a private and public key pair, the public key being a function of the private or secret key. Typically to send data to another party it is encrypted with the appropriate public for the user and from then on the data can only be retrieved by use of the secret key known only to the intended receiver.

There are various methods or algorithms upon which authentication can be based which surround the notion of public key cryptosystems and utilize cryptographically strong tools such as one way and universal hashing functions as well as digital signature methods. When used in authentication a signature is usually created which involves a message and the users secret key this signature can then be verified at the other end of the transmission by the use of the public key generated from the secret key and the message that was sent as part of the transmission.

The program Secure Talk is an attempt to solve the problems of authentication and security, by modification and addition to the UNIX talk program, which enables users to communicate using a split screen display.

Users type from the keyboard this appears on the users screen and in the respective half of the screen on the other users terminal.

A typical talk session can be seen figure 3

The general process of authentication adopted involves digest or fingerprint comparison. This provides a means by which users can be sure the person they are talking to is not an importer and provides a secure means for this communication through data encryption. Provided authentication is successful and the option is selected to protect the data an attacker would not be able to make sense of any transmissions intercepted as the parties communicate.

One of the important features of Secure Talk is that the communicating parties are provided with two levels of security, these being :

- Authentication Only
- Authentication and Security

This is an important feature in that if the system is adopted where by the authentication only option is available to users a limited amount of protection is provided. At the same time the transmissions would be available for scrutiny by a law enforcement or governmental body should the need arise.

3.0.1 The Authentication Process

Authentication is achieved through the use of two algorithms used to ensure that the users identity are genuine, the security problem being solved by encryption of the communicating data. Authentication is split into two parts an enciphering and deciphering algorithm, one algorithm in executed by either party depending on who is authenticating who. The following are the algorithms used

---

Algorithm 1 $E_{owh}(x_B, y_A, p, g, m)$

1. $x \in_R [1, p-1]$.
2. $r = y_A^{x_B+x} \bmod p$.
3. $z = G(r)_{[1 \cdots (P+\ell)]}$.
4. $t = h(m \mid\mid r)$.
5. $c_1 = g^x \bmod p$.
6. $c_2 = z \oplus (m \mid\mid t)$.
7. output (c_1, c_2).

end

The enciphering algorithm involves the following steps,

1. A random number x is generated in the range 1 to $p-1$.
2. The variable r is generated by calculating y_A the other persons public key raised to the sum of x_B our private key plus x, *mod* p is then applied so r is in the range 0 to $p-1$. A random bit-string z is generated using a cryptographically strong random string generator $G()$ in this instance STRANDOM [4] a generator based on HAVAL [6] is used with input r to generate a string equal in length to the message and appended digest.
3. The digest or tag t is created by hashing the message appended with r.
4. Now the first cipher-text c_1 is generated by raising g to the power x and applying *mod* p.

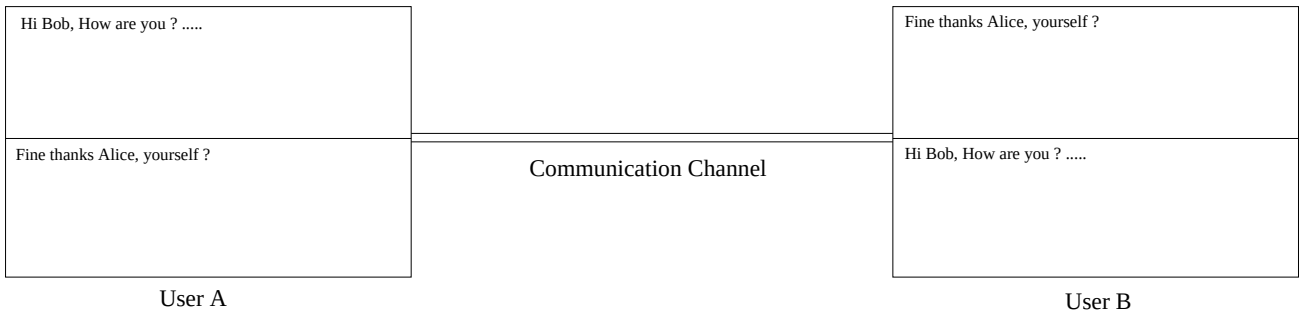

Figure 1: Typical Talk System

5. The second cipher-text $c_2$is finally generated by exclusive-ORing the string z and the message and appended tag, this effectively encrypts the message and appended tag. The cipher-texts c_1 and c_2 are output or sent to the other party for deciphering and the validation of the users identity.

Algorithm 2

$D_{owh}(x_A, y_B, p, g, c_1, c_2)$

1. $r' = (y_B \cdot c_1)^{x_A} \bmod p$.
2. $z' = G(r')_{[1\cdots(P+\ell)]}$.
3. $w = z' \oplus c_2$.
4. $m' = w_{[1\cdots P]}$.
5. $t' = w_{[(P+1)\cdots(P+\ell)]}$.
6. if $h(m' \mid\mid r') = t'$ then
   output (m')
   else
   output $(\emptyset)$.

end

The deciphering algorithm has the following steps to complete the authentication process.

1. The variable r' is calculated by the following process, y_B the other users public key is multiplied by c_1 the first cipher-text received from the enciphering process this product is then raised to the power of x_A or our private key and finally *mod* p is applied to get r in the range 1 to $p-1$.

2. z' a random bit string is then generated using $G()$ with the input r. If authentication is going to be successful r' and z' will be the same as r and z as in the enciphering process.

3. The second cipher-text is decrypted by exclusive-ORing c_2 with z' to reveal the message and appended digest.

4. The original message is extracted and stored in m'.

5. The precomputed digest is obtained from c_2.

6. Finally a new digest by hashing m' the original message appended with r', this digest is then compared to the precomputed digest, if equal then authentication has been successful otherwise the other parties identity is not genuine.

These algorithms originally used as presented in [7] were modified to increase the security of the process and can be found in [5].

In the Talking Securely system the message generated contains a 64 bit classical key to be used in the encryption and decryption of the conversation if necessary as well as some plaintext including a time stamp generated at the creation time of the message.

The algorithms require the users to have both a private and public key pairs to proceed with this process, various other strengths in the above algorithms lie in the hashing function h, and the pseudo random string generator g.

3.0.2 The Security Features

The various data security problems are overcome by encrypting the communication data prior to it being sent between users and it is appropriately decrypted upon being received. The encryption is done using the classical key approach where a part of the common classical key is sent to both parties as part of the message in the authentication process. Each party will generate a component of the key which is inserted

in the message this is then upon obtaining the other parties component combined by exclusive-OR. Once this key is generated and set, a random bit-string is generated, this is done using STRANDOM [4] where the key is used to seed the generator. The data is then exclusive-OR'd with the random string to encrypt. The random string generated is of a common length at both ends of the connection. As the data is encrypted the string becomes used up and a new different random string will be generated using the last 2 words of the previous string as the seed for the next string. Care needs to be taken to ensure that both ends of the encryption are synchronized so the data communicated is decrypted to the appropriate plain-text, and an agreement has to also be made as to the length of the random bitstring generated.

Using this simple process encryption and decryption are the same process and while provided the bit-string is a random one the process will produce secure encrypted data with little cost in terms of speed. This will not enable any party listening to the channel to make sense of the communication as it will be encrypted and any effort to tamper with the data will only be local to the letter or word.

3.0.3 Cryptographic Tools : Haval

One of the important tools in ensuring the strength of the public key crypto-system used the Secure Talk package is the one way hashing algorithm Haval [6]. This provides a secure way of processing the message to generate a unique digest. Haval is unique in that it can produce variable length output as well as three different security levels depending on the number of times the input is processed. Haval uses some of the ideas behind the design of MD5 (from the MD family of algorithms). It also utilizes five boolean functions which are considered desirable for a one way hashing algorithms described in [3]. When Haval is used in Secure Talk the length of the digest produced is 256 and the pass level is set to 5, giving maximum security.

Haval initially starts to hash a message by padding the message to a multiple of 1024 bits, this produces a number of blocks $B_{n-1}B_{n-2}\cdots B_0$. Haval then starts from block B_0 and an 8 bit constant string D_0 which is taken from the fraction part of $\pi = 3.1415...$, and processes the message in a block-by-block way as diagramitically illustrated in 2. As can be seen the message is compressed by repeatedly calculating

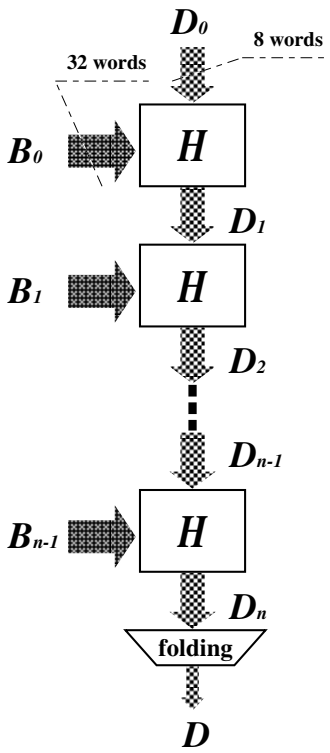

Figure 2: Hashing a Message by HAVAL

$$D_{i+1} = H(D_i, B_i)$$

where i ranges from 0 to $n-1$. The routine H is known as the updating algorithm of HAVAL. Prior to outputting the digest HAVAL adjusts the final 256 bit string D_n into the string of specified length. This adjusted string is output as the digest of the message M.

4 A trusted third party for public key distribution

One solution to the authentication problem is the usage of a trusted third party. One of the major problems in a public key cryptosystem is the distribution of public keys. The problems of storage and generation of a secret key are relatively trivial in comparison. A secret key of substantial size can be generated from a small password which is remembered by the user, through the use of a random number generator or similar tool. The secret key can be stored in a file in the users home directory if necessary. For security and verification this file can be signed

by the use of a secure hashing function and encrypted once again using the password as the key to some encryption scheme. Thus the user is not greatly inconvenienced by having to enter a small password when compared with the benefit from secure authentication.

This is the approach adopted by the Secure Talk Package in the storage of users secret keys when the initial secret key is created from the users password this key is hashed and then encrypted using a second random bitstring, the first bitstring being the secret key. If the user wishes the key generated from the user input can be checked with the key stored in the file for verification purposes. Various approaches along similar ideas to this were explored for the storage of public keys, however as the keys are public the public key files are open to attack rendering the keys useless.

The main problem is the public key and its distribution, which poses the question of how to make sure that the other communicating party is able to get a correct copy of the public key ie. one which has not been tampered with. As the public key file is publicly accessible by nature it can also be tampered with, removed, or substituted with a different users public key, this would foil the authentication process.

When two legitimate parties try to communicate there is no way of knowing if either parties public key is correct. There could be various attempts to make the file tamper-proof such as hashing the file, however merely hashing the public key and appending the digest to the end of the file provides little protection as the whole file can be replaced or the key changed and a new correct tag appended by a foreign party. One way to protect the file from modification is to sign the file by hashing the file combined with some information private to the user such as the private key or part of it, however this can not be verified by other parties as they do not know the secret component used to create the signature, and as the parties have no common secret data this problem is a circular one, which is unsolvable. As neither of the parties can assume the other has a genuine identity no trust has been established, hence the problem of public key distribution cannot be solved without the inclusion of a third party which is trusted by each user. Through this third party each user can trust that the public key belonging to the other party is valid. One example of this is a server which issues certificates with public keys through which users initially communicate. Through trusting the server the parties are able to establish some trust between each other. The situation in figure 4 is needed to rectify this problem.

The assumption here is that the public key server has a public key p_{ks} which is known to all parties, this is a realistic assumption as it is easier and more efficient than storing the public keys of all other users. When a request is made for a users public key to the public key server, it will return a pair this will consist of $(y_b, Cert_b)$ where :

- y_b is the public key belonging to user B.
- $Cert_b$ is the certificate associated with y_b. This certificate will be the signature made by the Key Server on y_b

Upon receiving the pair $(y_b, Cert_b)$, the user who requested the key will check the validity of $(y_b, Cert_b)$ using the Servers public key p_{ks}.

4.1 Pretty Good Privacy (PGP)

This is a public key crypto-system or encryption package which can be used to allow communication between parties with both Authentication and Privacy without secure channels [8].

Encryption is used as the primary means of obtaining privacy, transmissions should be encrypted with the intended receivers public key however this is comparatively slower when compared to conventional single key encryption. To overcome the speed problem of public key encryption the signed plaintext to be transmitted is encrypted using a single key algorithm with a temporary random 'session key'. This 'session key' is then enciphered using the receivers public key and sent along with the encrypted data to its destination, where the 'session key' is decrypted using the private key of the receiver which is then used to recover the original communication.

Authentication is achieved by creating a signature using the senders private key to encrypt the communication. This signature is then appended to the the data, upon the data reaching its destination the recipient can check the digital signature by using the senders public key and the 'session key' to decrypt the transmission, and also to verify the signature. From this verification the messages source as well as checking if the data has been tampered with during its transmission can be done. Only the original

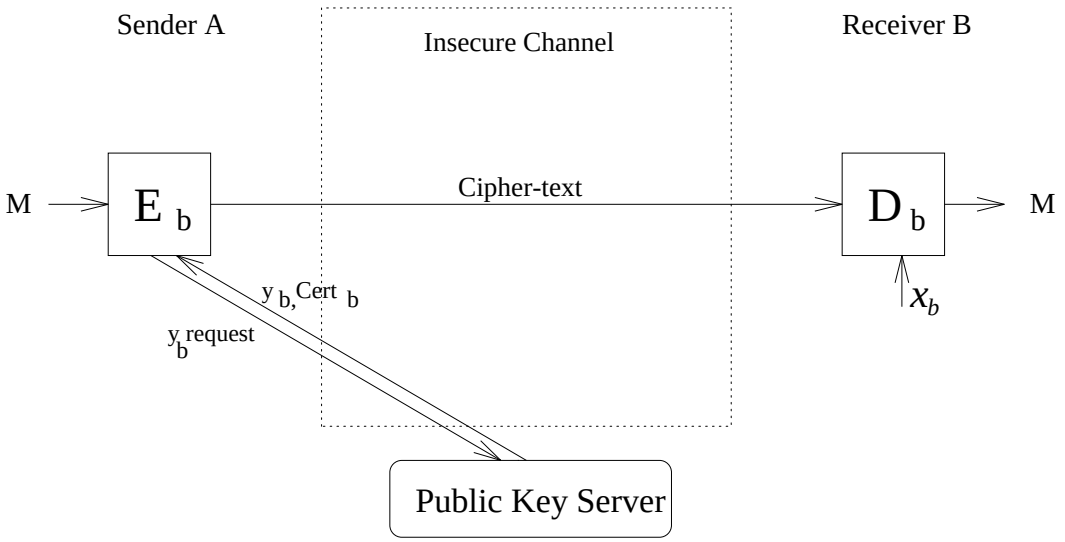

Figure 3: Public Key Distribution

sender is able to change the message and signature, as this is the only party that has knowledge of the secret key that made the signature.

The public keys are kept in individual 'key certificates' which include information such as the owners *userid* the time when the public/private key pair was created and the public key material itself.

Secret key certificates contain information of the same nature as public key certificates except pertaining to the users secret key

To create a signature a 'message digest' is a obtained from a cryptographically strong one-way hash function, this digest being 128 bits in length. This compactly 'represents' the message and is used to detect changes. The message digest gets encrypted using the secret key to form a signature.

Documents are signed by attaching signature certificates. These certificates contain an identification number of the key that was used to sign it, the signature, and a time-stamp of when the signature was made. The receiver looks up the senders public key which is used to verify the signature.

Encrypted files are prefixed by the key ID of the public key used to encrypt them. The receiver uses this key ID message prefix to look up the secret key needed to decrypt the message. The receivers software automatically looks up the necessary secret decryption key in the receivers secret key ring.

Two types of key rings are the principal method of storing and managing public and secret keys. Rather than keep individual keys in separate key files, they are collected in key rings to facilitate the automatic lookup of keys either by key ID or by user ID. Each user keeps his own pair of key rings.

The PGP system provides the user with the option of variable key lengths up to around one thousand bits. The problem of key distribution is solved by having a various servers on the network where public keys can be obtained, upon creating a private and public key pair the user simply informs a server of the key and it is added to the public key file, the user only needs to notify one server as the servers are all networked.

4.2 Kerberos

Kerberos is a network authentication system for use on physically insecure networks [2]. It allows parties communicating over networks to prove their identity to each other while preventing eavesdropping or replay attacks. It also provides a method of detecting data modification and unauthorized access using cryptography systems such as DES.

Kerberos works by providing parties with tickets that they can use to identify themselves to other parties and secret keys for secure communication.

Kerberos provides for mutual authentication and secure communication between principals on an open network by manufacturing secret keys for any request and providing a mechanism for these secret keys to be safely propa-

gated through the network. Kerberos does not however provide for authorization or accounting, although applications that wish to can use their secret keys to perform those functions securely.

There is one drawback to the Kerberos authentication package being that it is not available outside the United States. This is due to the fact that the encryption system, DES (Data Encryption Standard) is not exprotable, therefore Kerberos can not be exported in its original form. A system known as 'Bones' has been developed this provides Kerberos authentication without using encryption hence no security whatsoever is provided. The primary use of 'Bones' is that it allows software expecting Kerberos to be present to function.

4.3 Kuperee

Kuperee [1] is a recently developed protocol which implements Kerberos type authentication using the ideas of public key cryptography. The Kerberos authentication service adopts a private key approach where the user and server share a secret key which is used for authentication, when the user makes a request to the server. The secret key is also shared between other trusted authorities such as the Ticket Granting Server and the Key Distribution Server, from this use of private key cryptography the system is open to attack on either of the trusted parties, compromising security greatly. If any one of the trusted parties is compromised an attacker may be able to pose as another user through obtaining a private key from this compromised authority.

To overcome this threat a public key approach has been adopted, where the user and the server or trusted authorities share a publicly-known key this requires less trust to be placed by the user in these authorities. Other advantages include enabling the users to perform one-to-one secure communication, with Kerberos this was not possible as the users had to establish a common key via a key exchange protocol before communication could commence. The public key based algorithms of the Kuperee system are those of [1], these algorithms are attractive due to their resistance to adaptively chosen ciphertext attacks.

5 Discussion

After examination of the above problems and solutions it can be seen that there is a trade off between providing security and authentication to users and convenience.

At one end of the spectrum there is a network that has no protection, in this situation the users are not inconvenienced by time consuming authentication routine and small delays in encryption or decryption of data communications. The downside to this situation being the many unprotected avenues of attack open to an opponent of a network.

At the other end of the scale users can be provided with authentication and security services, an amount of inconvenience may be incurred due to users having to remember passwords and the time delays incurred through authentication processes and encryption-decryption routines. In this scenario an attackers access to insecure channels will provide little or no meaningful information and in gathering this information the intruder has a reasonable chance of detection if data is tampered with.

For general users most of the time, complex security measures are not needed due to the nature of their data, but these type of transmissions should have a minimal amount of protection attack from idle opponents of a computer system.

It needs to be justified that the additional calculations required in achieving the security measures are necessary, this being determined by the nature of the data to be transmitted. In addition to this a mechanism may be needed where by a user can determine the level of protection sought on transmissions such as Electronic Mail, remote login sessions, and general network file transfers. By this mechanism the user can decide the measures needed and justify the time and other inconveniences such as key generation, management and distribution.

6 Suggestions

With current speeds of both hardware and the forward march of technology, it is paramount to provide some security features appropriate to the needs of the data being transferred. The time taken for authentication and data encryption will continue to decrease to a point were this will no longer be an issue or consideration. Currently a minimal level of security should be

provided giving users their right to privacy and security of data. Simple but effective authentication schemes such as PGP provide a useful system in which users can authenticate data transfers such as electronic mail messages and alike.

With the current growth of public key servers it can be seen that these are an effective way of providing security and authentication to the user.

All the current ventures into encryption and authentication suggest that general security principles should be part of network and computing infrastructure. Other general consumer items such as the motor car in response to theft and security problems has evolved so that security has become an issue at the production level, manufacturers now install theft resistant door locks, alarms and other devices into cars on the production line as part of the design. The same general principles should be taken into consideration when computers and networks are built so as to provide the user with at least a minimal level of security and authentication, rather than researches and developers having to develop their own systems long after networks have come into use.

7 Conclusion

We have presented the problems of security involving issues such as authentication, privacy and data integrity, and looked at ways in which these problems can be solved. For the problem of authentication public key cryptography and the use of public key servers provides a secure solution to this dilemma. These problems of security involving the issues of channel monitoring as well as interception and modification of data can also be averted through the use of encryption, which may also involve the use of public key cryptosystems or variations on this theme.

The Secure Talk program was presented together with an examination of some other alternatives in their to combat the authentication and data integrity problems.

Finally the discussion of the issues with regard to computer security included the problems and solutions, looking at some of the popular current implementations, together with suggestions based on this discussion.

Acknowledgment

The Authors of this paper have been supported by the Australian Research Council under reference number A49232172 and the University of Wollongong Research Program Grant.

We would like to thank the other members of Software Project Group 18 at Wollongong University, these being Anthony Boyd, Susan Dalton, Flora Lo, Adam Millard and Shalini Shah for the original development and implementation of the Talking Securely System.

References

[1] T. Hardjono, Y. Zheng, and J. Seberry. Kuperee : An approach to kereberos-type authentication using public-keys, October 1993.

[2] Burt Kaliski. Kerberos, osf/dce, and public key, 1993. RSA Data Security Conference.

[3] J. Seberry and X.-M. Zhang. Highly nonlinear 0-1 balanced boolean functions satisfying strict avalanche criterion, 1993.

[4] Y. Zheng. STRANDOM — a cryptographically strong pseudo-random number generator based on HAVAL, July 1993. (source code in the programming language C).

[5] Y. Zheng. Improved public key cryptosystems secure against chosen ciphertext attacks, February 1994.

[6] Y. Zheng, J. Pieprzyk, and J. Seberry. HAVAL — a one-way hashing algorithm with variable length of output. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - Proceedings of AusCrypto'92,* Lecture Notes in Computer Science, Vol. 718, pages 83–104. Springer-Verlag, 1993.

[7] Y. Zheng and J. Seberry. Immunizing public key cryptosystems against chosen ciphertext attacks. *Special Issue on Secure Communications, IEEE Journal on Selected Areas on Communications*, 11(5):715–724, June 1993.

[8] Philip Zimmermann. Pretty good privacy : Public key encryption for the masses, June 1993.