

# DATA ENCRYPTION AND CRYPTO POLITICS

## 1 SECURITY REQUIREMENTS

- Confidentiality  
Protection from disclosure to unauthorized persons
- Integrity  
Maintaining data consistency
- Authentication  
Assurance of identity of person or originator of data
- Non-reproduction  
Originator of communications can't deny it later
- Availability  
Legitimate users have access when they need it
- Access control  
Unauthorized users are kept out
- These are often combined  
User authentication used for access control purposes

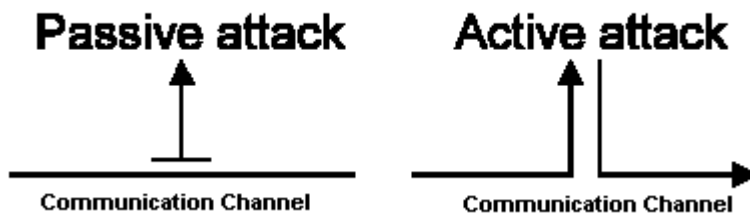
## 2 ATTACK TYPES

### 2.1 PASSIVE ATTACK

It is classified as the as the observation of the communications or data. In this the attacker's intention is to know the data, which is passing through the communication channel. He is not having intention to change the data.

### 2.2 ACTIVE ATTACK

It is compared to be more harmful than passive attack since the received data will not be same as the data, which is sent. The attacker will receive that data and will actively modify communications or data.



### **3 WHY WE NEED DATA ENCRYPTION?**

Often there has been a need to protect information from 'prying eyes'. In the electronic age, information that could otherwise benefit or educate a group or individual can also be used against such groups or individuals. Industrial espionage among highly competitive businesses often requires that extensive security measures be put into place. And, those who wish to exercise their personal freedom, outside of the oppressive nature of governments, may also wish to encrypt certain information to avoid suffering the penalties of going against the wishes of those who attempt to control.

Although password protection and physical security measures can be implemented to limit access to the computer data, hackers and criminals still manage to gain access. Important data such as credit card accounts, bank records, etc, should therefore be stored in an encrypted format to foil hackers who break into computers. In addition, when this sort of data is transmitted over the Internet, it should be *encrypted*. Data Encryption is used to prevent the attacker from altering the message unbeknownst to the receiver, and to prevent the sender from later denying that he/she sent a particular message on a particular date and time.

When Julius Caesar sent messages to his generals, he didn't trust his messengers. So he replaced every A in his messages with a D, every B with an E, and so on through the alphabet. Only someone who knew the "shift by 3" rule could decipher his messages.

### **4 TYPES OF ENCRYPTION**

Traditionally, several methods can be used to encrypt data streams, all of which can easily be implemented through software, but not so easily decrypted when either the original or its encrypted data stream are unavailable. (When both source and encrypted data are available, code breaking becomes much simpler, though it is not necessarily easy). The best encryption methods have little effect on system performance, and may contain other benefits (such as data compression) built in.

Usually some reversible mathematical operations, which can be easily done by a computer, are used for encryption. For eg : XOR operation and Permutation is used in encryption. In a permutation step, the order of the bits in the bit sequence is rearranged.

Cryptographic strength is measured in the time and resources it would require to recover the plaintext. The result of *strong cryptography* is cipher text that is very difficult to decipher without possession of the appropriate decoding tool. How difficult? Given all of today's computing power and available time—even a billion computers doing a billion checks a second—it is not possible to decipher the result of strong cryptography before the end of the universe.

## **4.1 ENCRYPTION USING TABLES**

The simplest of all of the methods, the 'translation table', is one of the simplest way of encryption. Each 'chunk' of data (usually 1 byte) is used as an offset within a 'translation table', and the resulting 'translated' value from within the table is then written into the output stream. The encryption and decryption programs would each use a table that translates to and from the encrypted data. In fact, the 80x86 CPU's even have an instruction 'XLAT' that lends itself to this purpose at the hardware level. While this method is very simple and fast, the down side is that once the translation table is known, the code is broken. Further, such a method is relatively straightforward for code breakers to decipher - such code methods have been used for years, even before the advent of the computer. Still, for general "unreadability" of encoded data, without adverse effects on performance, the 'translation table' method lends itself well.

A modification to the 'translation table' uses 2 or more tables, based on the position of the bytes within the data stream, or on the data stream itself. Decoding becomes more complex, since you have to reverse the same process reliably. But, by the use of more than one translation table, especially when implemented in a 'pseudo-random' order, this adaptation makes code breaking relatively difficult. An example of this method might use translation table 'A' on all of the 'even' bytes, and translation table 'B' on all of the 'odd' bytes. Unless a potential code breaker knows that there are exactly 2 tables, even with both source and encrypted data available the deciphering process is relatively difficult

## **4.2 DATA REPOSITIONING**

Similar to using a translation table, 'data repositioning' lends itself to use by a computer, but takes considerably more time to accomplish. A buffer of data is read from the input, then the order of the bytes (or other 'chunk' size) is rearranged, and written 'out of order'. The decryption program then reads this back in, and puts them back 'in order'. Often such a method is best used in combination with one or more of the other encryption methods mentioned here, making it even more difficult for code breakers to determine how to decipher your encrypted data. As an example, consider an anagram. The letters are all there, but the order has been changed. Some anagrams are easier than others to decipher, but a well-written anagram is a brainteaser nonetheless, especially if it's intentionally misleading.

## **4.3 SINGLE KEY ENCRYPTION**

Data that can be read and understood without any special measures is called *plaintext* or *clear text*. The method of disguising plaintext in such a way as to hide its substance is called *encryption*. Encrypting plaintext results in unreadable gibberish called *ciphertext*.

You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting ciphertext to its original plaintext is called *decryption*.



Single key encryption alone as a means for transmitting secure data can be quite expensive simply due to the difficulty of secure key distribution. For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier, the Bat Phone, or some other secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key.

From DES to Captain Midnight's Secret Decoder Ring, the persistent problem with conventional encryption is *key distribution*: how do you get the key to the recipient without someone intercepting it?

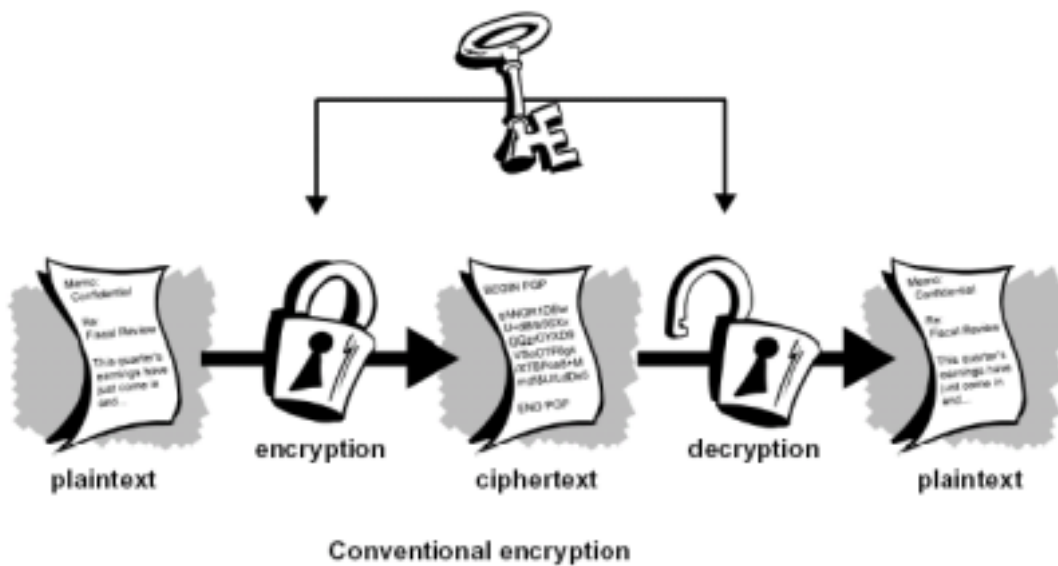


Figure 2 Data Encryption Seminar

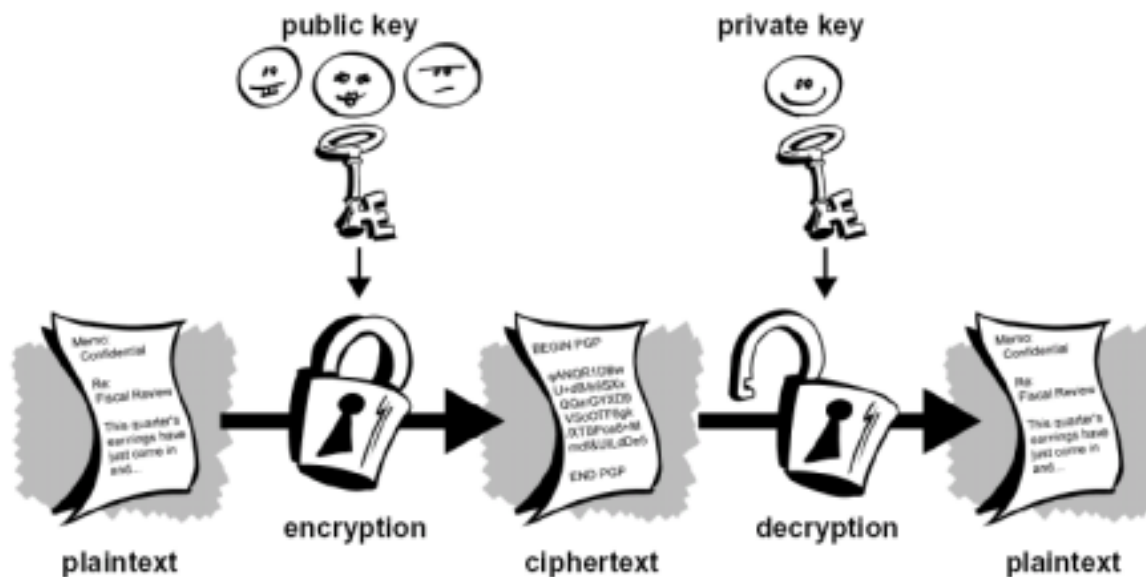
Advantage of Single key Encryption :- Conventional encryption has benefits. Problem of communicating a large message in secret reduced to communicating a small key in secret. It is very fast. It is especially useful for encrypting data that is not *going* anywhere.

However, conventional encryption alone as a means for transmitting secure data can be quite expensive simply due to the difficulty of secure key distribution. So the greatest problem of single key encryption is key distribution.

#### 4.4 TWO KEY ENCRYPTION

The problems of key distribution are solved by *public key cryptography*, the concept of which was introduced by Whitfield Diffie and Martin Hellman in 1975.

Public key cryptography is an asymmetric scheme that uses a *pair* of keys for encryption: a *public key*, which encrypts data, and a corresponding *private*, or *secret key* for decryption. You publish your public key to the world while keeping your private key secret. Anyone with a copy of your public key can then encrypt information that only you can read. Even people you have never met.



Two Key Encryption

Figure 3 Data Encryption Seminar

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Some examples of public-key cryptosystems are Elgamal (named for its inventor, Taher Elgamal), RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard

Adleman), Diffie-Hellman (named, you guessed it, for its inventors), and DSA, the Digital Signature Algorithm (invented by David Kravitz).

There are few operations in mathematics that are truly 'irreversible'. In nearly all cases, if an operation is performed on 'a', resulting in 'b', you can perform an equivalent operation on 'b' to get 'a'. In some cases you may get the absolute value (such as a square root), or the operation may be undefined (such as dividing by zero). However, in the case of 'undefined' operations, it may be possible to base a key on an algorithm such that an operation like division by zero would PREVENT a public key from being translated into a private key. As such, only 'trial and error' would remain, which would require a significant amount of processing time to create the private key from the public key.

In the case of the RSA encryption algorithm, it uses very large prime numbers to generate the public key and the private key. Although it would be possible to factor out the public key to get the private key (a trivial matter once the 2 prime factors are known), the numbers are so large as to make it very impractical to do so. The encryption algorithm itself is ALSO very slow, which makes it impractical to use RSA to encrypt large data sets. What PGP does (and most other RSA-based encryption schemes do) is encrypt a symmetrical key using the public key, then the remainder of the data is encrypted with a faster algorithm using the symmetrical key. The symmetrical itself key is randomly generated, so that the only way to get it would be by using the private key to decrypt the RSA-encrypted symmetrical key.



## 5 KEYS

A key is a value that works with a cryptographic algorithm to produce a specific cipher text. Keys are basically really, really, really big numbers. Key size is measured in bits; the number representing a 1024-bit key is darn huge. In public key cryptography, the bigger the key, the more secure the ciphertext. However, public key size and conventional cryptography's secret key size are totally unrelated. A conventional 80-bit key has the equivalent strength of a 1024-bit public key. A conventional 128-bit key is equivalent to a 3000-bit public key. Again, the bigger the key, the more secure, but the algorithms used for each type of cryptography are very different and thus comparison is like that of apples to oranges.

While the public and private keys are mathematically related, it's very difficult to derive the private key given only the public key; however, deriving the private key is always possible given enough time and computing power. This makes it very important to pick keys of the right size; large enough to be secure, but small enough to be applied fairly quickly. Additionally, you need to consider who might be trying to read your files, how determined they are, how much time they have, and what their resources might be. Larger keys will be cryptographically secure for a longer period of time.

In the case of the RSA encryption algorithm, it uses very large prime numbers to generate the public key and the private key. Although it would be possible to factor out the public key to get the private key (a trivial matter once the 2 prime factors are known), the numbers are so large as to make it very impractical to do so. The encryption algorithm itself is ALSO very slow, which makes it impractical to use RSA to encrypt large data sets.

Just for explanatory purpose here is an example of RSA key pair generation and use.

|  |  |
|--|--|
| $n, e = \text{public key, } n = \text{product of two primes } p \text{ and } q$ $d = \text{private key}$ $\text{Encryption: } C = M^e \bmod n$ $\text{Decryption: } M = C^d \bmod n$ $p, q = 5, 7$ $n = p \times q$ $= 35$ $e = 5$ $d = e^{-1} \bmod ((p-1)(q-1))$ $= 5$ | $\text{Message } M = 4$ $\text{Encryption: } C = 4^5 \bmod 35$ $= 9$ $\text{Decryption: } M = 9^5 \bmod 35$ $= 59049 \bmod 35$ $= 4$ |
|--|--|

Data Encryption seminar RSA eg Fig 6 mv

What you want to encrypt needs to be hidden for many years, you might want to use a very large key. Of course, who knows how long it will take to determine your key using tomorrow's faster, more efficient computers? There was a time when a 56-bit symmetric key was considered extremely safe. Keys are stored in encrypted form. PGP stores the keys in two files on your hard disk; one for public keys and one for private keys. These files are called *key rings*. As you use PGP, you will typically add the public keys of your recipients to your public keyring. Your private keys are stored on your private keyring. If you lose your private keyring, you will be unable to decrypt any information encrypted to keys on that ring.

An example of the PGP (Pretty Good Privacy) is given below. Just for an example the for the mail Id mathew\_vv@sify.com the private key will be like the following (PGP DES Algorithm 1024 bit)

```
•0i0<2sA00 N003>00A0^$Z0AžjZPv0000h0i+K0,,>"0 0+Cmi0t">60µd00A%yc`6`°1@
-]AQ%0$0,%U0AX0^ež†5-šZw`æ`ē`u0]0,-ā00.,'A2Hrr"mo y ň/#3]DÁV`Dwš)µ0
0-Ā0,'ē†#DF -0ē'ē0J6HIāµ0^\">00Éigxy00àç@6ā"5Yi'Āød}pæz'0K?IZZ0Rý0&α
«%B%0;TCX0±NĀ~JpPæi«00á8lY$0|U`¥α000p?ZÜáf$ðæ00L00[DiiçĀαoxm+^†0s-0`A,
ž00Ū0Y>±`0}50-ŕ"003Ā0..UµN`9+*0EYα00Ā£sxo/ĀYiEM;0aŋ] ŸY=ē`nL`l0
0\\335]KŸ00000æ^000±ā`ē:IĒ0r0Z^sèiŪiŪē2.2<ēā: ]`00;13i0`$Mathew
Varghese <mathew_vv@sify.com>% N0000 000<2sA00000000 0 0|iV!ú0gw00
úK007~DĀ0ŕŵi0ž/çE%` ē 3-3áčlt0E00,-,EOA00P0<2sA00 úS090ŋŕn`K8
Ū0$VĒ0é0@U000ĀĒĒC@"-+Ld†I00fŸæé00;0P00Ÿ_ŪvRO==00i"áw`Yiy_0toĀŪv%ŸfĀG0aif|É
fŸ00ú»Ÿ`Ac0^0[ž0 ¥r:00žF X+ā0`ŸXNI0i`0ā`06k3l80E
ž»U0E,]804Ÿ|0WCĒ00lP3!,µ*Ÿ<á±)@000|„š0r0†Ā00Ē0)zĒ•0Ū-Ÿ<00
P>0F00=fŕ]ĀŸ0|Ÿ%K"0&°«4ĀĀUé20; 00Ÿ0Ū0±0^I00IM0Ē
áŪĀéi0Ÿ`è\|nwtC0Ÿ0Ā0%Ī#00{00ŪŸE]8Æz0ŪŪ00É%Ÿ:
]0%G0$qtw0Ÿ00x@0|u0;0ŋs000/V 00r0i„0{#8`Ÿ†ç6«000Ÿ%;jĀ`vLfè8ŋ•1*`Ÿäfs"´0
6E;00C/0;ž4ó æ0ž@LH0Bµn4Āé4rŷ"æĒ0|iĒ.
4r0.81%0Āè0c&^cxš040{Ā7Ūf+SII`xĀq0G05u03sα:ŸŪ'...L-00i_Ū"00d0i)x\š0điNR?`
çJ]PM(0^0iP_x- X000Ÿ000<ewĪKp0E`F`{N00
0R+jf(i0i7"j0ē»0~0w0Ēm0b0%~Ÿ000Ā...0%0g f~"!pR}Ā% F0000 000<2sA 0
0|iV!ú0gw`Ī` 0{000P4Ÿ7ā0i000±Dp\ A-0ŪĀ1R,0xú.<00 i0x0
```

## 6 BLOCK CIPHERING SYSTEMS

Consider an example of Block encryption through a channel which is not secure (By highly unbreakable encryption algorithm !!!)

Original text *Block encryption if block width is known !!!!!*

Deposit \$10,000 in acct. number 12-3456-789012-3

Intercepted encrypted form

H2nx/GHE KqvldSbq GQHbrUt5 tYf6K7ug S4CrMTvH 7eMPZcE2

Second intercepted message

H2nx/GHE KqvldSbq GQHbrUt5 tYf6K7ug Pts21lGb a8oaNWpj

Cut and paste blocks with account information

H2nx/GHE KqvldSbq GQHbrUt5 tYf6K7ug S4CrMTvH a8oaNWpj

Decrypted message will contain the attacker's account —  
without them knowing the encryption key *mvv fig 5*

### ➤ Triple DES (3DES)

Encrypt + decrypt + encrypt with 2 (112 bits) or 3 (168 bits) DES keys  
By late 1998, banking auditors were requiring the use of 3DES  
rather than DES

- **RC2**  
Companion to RC4, 1024 bit key  
RSADSI trade secret, reverse-engineered and posted to the net in 1996  
RC2 and RC4 have special status for US exportability
- **IDEA**  
Developed as PES (proposed encryption standard), adapted to resist differential cryptanalysis as IPES, then IDEA  
Gained popularity via PGP, 128 bit key  
Patented
- **BLOWFISH**  
Optimized for high-speed execution on 32-bit processors  
48 bit key, relatively slow key setup
- **CAST-128**  
Used in PGP 5.x, 128 bit key
- **SKIPJACK**  
Classified algorithm originally designed for Clipper, declassified in 1998  
Very efficient to implement using minimal resources (e.g. smart cards)  
32 rounds, breakable with 31 rounds  
80 bit key, inadequate for long-term security
- **GOST**  
GOST 28147, Russian answer to DES  
32 rounds, 256 bit key  
Incompletely specified
- **AES**  
Advanced Encryption Standard, replacement for DES  
128 bit block size, 128/192/256 bit key

Many, many others

No good reason not to use one of the above, proven algorithms

## **RELATIVE PERFORMANCE**

**Fast :-** RC4, Blowfish, CAST-128, AES, Skipjack, DES

**Slow :-** IDEA, RC2, 3DES, GOST

**Typical speeds :-**

RC4 = Tens of MB/second

3DES = MB/second

## **Recommendations**

For performance, use Blowfish

For job security, use 3DES

**Length of key**

512 bit key is marginal

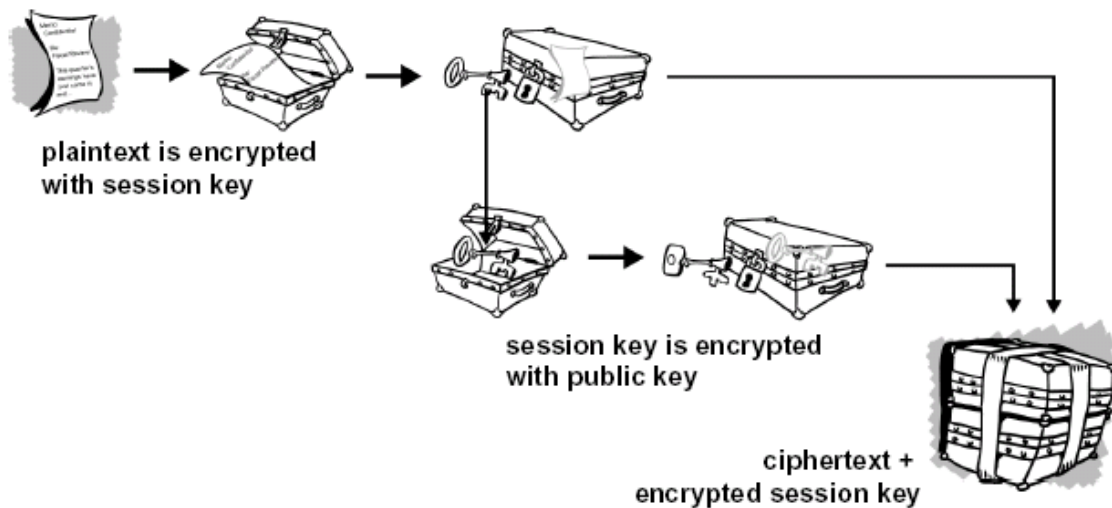
1024 bit key is recommended minimum size

2048 bit key is better for long-term security

**7 DATA ENCRYPTION SOFTWARES, PGP**

PGP combines some of the best features of both conventional and public key cryptography. PGP is a *hybrid cryptosystem*. When a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security.

Most cryptanalysis techniques exploit patterns found in the plaintext to crack the cipher. Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to cryptanalysis. (Files that are too short to compress or which don't compress well aren't compressed.) PGP then creates a *session key*, which is a one-time-only secret key. This key is a random number generated from the random movements of your mouse and the keystrokes you type. This session key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is cipher text. Once the data is encrypted, the session key is then encrypted to the recipient's public key. This public key-encrypted session key is transmitted along with the cipher text to the recipient.



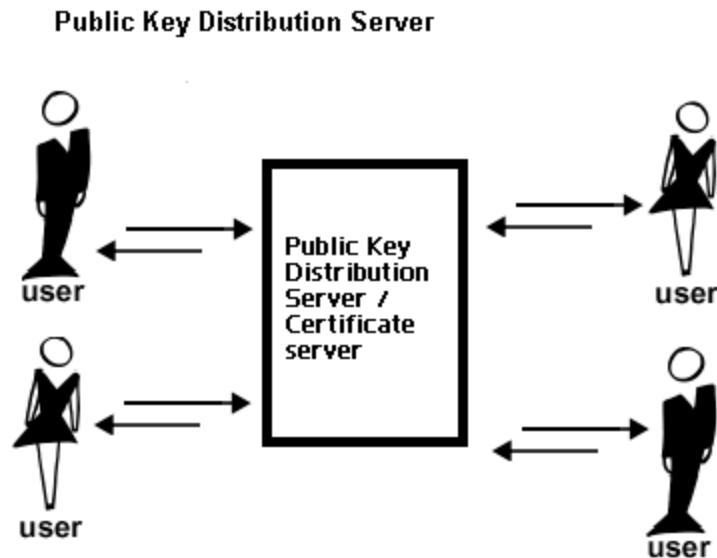
*How PGP encryption works,*

*Data Encryption Seminar Fig 7*

Decryption works in the reverse. The recipient's copy of PGP uses his or her private key to recover the temporary session key, which PGP then uses to decrypt the conventionally-encrypted cipher text.

## 8 PUBLIC KEY INFRASTRUCTURES

A PKI contains the certificate storage facilities of a certificate server, but also provides certificate management facilities (the ability to issue, revoke, store, retrieve, and trust certificates). The main feature of a PKI is the introduction of what is known as a *Certification Authority*, or *CA*, which is a human entity—a person, group, department, company, or other association—that an organization has authorized to issue certificates to its computer users. (A CA's role is analogous to a country's government's Passport Office.)



Data Encryption seminar Figure No. 8

A CA creates certificates and digitally signs them using the CA's private key. Because of its role in creating certificates, the CA is the central component of a PKI. Using the CA's public key, anyone wanting to verify a certificate's authenticity verifies the issuing CA's digital signature, and hence, the integrity of the contents of the certificate (most importantly, the public key and the identity of the certificate holder).

## **9 CRYPTO POLITICS**

In God we trust. All others we monitor

— NSA motto

### **History Of Crypto Politics**

- 1977 NSA tried to block NSF funding of crypto research  
Attempt to intimidate IEEE over security conference
- 1978 NSA uses Invention Secrecy Act to classify crypto patents
- 1979 Bobby Ray Inman's "The sky is falling" speech: NSA should control crypto research
- 1982 NSA blocked NBS request for public-key equivalent of DES
- 1984 NSDD-145 moves control of computer security from NBS to NSA (NSA memo calls NSDD-145 "NSA-engineered")
- 1986 NSDD-145 extended to allow NSA jurisdiction over private databases (Dialog, CompuServe) NSA tries to decertify DES CCEP (Commercial COMSEC Endorsement Program) using NSA-designed tamperproof hardware (eg Blacker)
- 1987 Computer Security Act moved control of crypto back to NBS
- 1988 NSA tries to block publication of Khufu block cipher
- 1989 NSA/NIST memorandum of understanding moves control of crypto back to the NSA
- 1990 NSA designs signature-only PKC for NIST, begins work on Clipper
- 1991 NIST announces DSS and NSA-designed SHS  
Industry reaction was almost universally negative
- 1994 PGP publishes its first product for data encryption
- 1995 Network Associates Takes over PGP
- 1995 U.S government makes a new policy decision of one point of public key algorithms
- 2000 RSA patent renewed till 2005

### **Federal Controls in Digital Telephony**

Law Enforcement Requirements for the Surveillance of Electronic Communications, 1992

- Real-time, full-time monitoring capability
- Intercepts undetectable to all parties (including service providers)
- Multiple simultaneous intercepts possible
- Decoding or decryption of all communications
- Supplementary information provided is: – Directory number, associated directory number, line equipment number, call type/bearer capability, service profile identifier, PBX directory number, PBX station identifier, electronic serial number (ESN), mobile identification number (MIN), terminal equipment identifier, and service site information (for cell phone tracking)

Black Box encryption equipment **Clipper** was introduced to the all the Digital Transmission of AT&T Network

80% of Americans opposed it Of over 300 submissions, only 2 were supportive  
Clipper adopted as Escrowed Encryption Standard (EES), FIPS 185, in February 1994

The legal machinations required to get this adopted fill a 200- page law journal article

No one bought Clipper

AT&T shut down its product line

FOIA'd documents obtained later showed that the government had a secret key escrow policy which was the exact opposite of the publicly claimed Clipper policy

### **Export controls are highly effective in ensuring that the masses have no real security**

The majority of all crypto in use worldwide is crippled or broken

- 77% of Thawte users are using weak encryption
- 60% of them are in the US
- For most of its existence, Verisign issued weak (512-bit) keys to users outside *and inside* the US

Practical example of export control effects is demonstrated by CIA hacking into European parliament computers in 1996 (Sunday Times): “includes details of the private medical and financial records of many MEPs and officials, and discussion documents on confidential issues, including trade, tariff and quota agreements. The breach came to light when officials believed that American negotiators had been given advance warning of confidential European Union positions in last year’s trade negotiations” “They were able to exploit the fact that parts of the system were manufactured by two American firms”

### **French and Russian Crypto Controls**

French controls are based on the “decret de 18 avril 1939”

On a scale of 1 to 8, encryption is rated 2

- Netscape is the second most dangerous weapon type recognized by the French government

Modified constantly over the years, “decret 86-250 du 18 fev 1986” explicitly mentions encryption software, “loi 90-1170 du 29 December 1990” requires approval for encryption use from the Prime Minister

- “If you don’t tell us you’re using PGP, none will bother you. If you ask us for permission to use it, we will refuse”

— J. Vincent-Carrefour, head of the SCSSI

### **NSA targets private individuals**

NSA maintained 1,056 pages of files on Princess Diana (Washington Post)

NSA produced 39 internal publications on Diana

Information was collected over a period of years

“NSA systematically intercepts international communications,  
both voice and cable”

— NSA Director Lt.General Lew Allen testifying before  
Congress

### **Indian laws**

India does not have many laws on these systems. Indian telegraph act came into existence in 1865 and it got amended from time to time. All sort of communications are managed by TRAI (Telephone Regulatory Authority of India).

## **BIBLIOGRAPHY**

**PGP!** <http://www.pgp.com/> or [www.nai.com/pgp](http://www.nai.com/pgp)  
A GREAT Enigma article, how the code was broken by Polish scientists  
[www.members.aol.com/nbrass/1enigma.htm](http://www.members.aol.com/nbrass/1enigma.htm)  
**RSA Encryption Systems** [www.rsa.com](http://www.rsa.com)  
<http://www.tutorialfind.com/tutorials/networking/security/>  
<http://www.cs.uiowa.edu/~jones/compress/>  
<http://members.ozemail.com.au/~firstpr/crypto/#PKAF>  
<http://homepages.paradise.net.nz/fisiihoi/tutorials/text/encryption.htm>  
<http://www.acda.gov/factshee/exptcon/wasse.htm>  
<http://www.bxa.doc.gov/encstart.htm>  
<http://www.bxa.doc.gov/Encryption/EncrpolycyUpdate.htm>  
<file:///F:/educational/Data%20encryption%20seminar/pkafcom.htm#update>  
<http://www.gil.com.au/~gtaylor/pgp.html>  
<http://www.counterpane.com/>  
<http://www.erols.com/gwmoore>

Thanks to Peter Gutmann ,University of Auckland  
<http://www.cs.auckland.ac.nz/~pgut001> for his notes on various aspects of  
cryptography.

Send your Comments and Suggestions to [mathew@valiyaparambil.com](mailto:mathew@valiyaparambil.com)  
Website [www.valiyaparambil.com/mathew.html](http://www.valiyaparambil.com/mathew.html)