

# **Cryptography**

## **Open Source Security Tools for Information Technology Professionals**

**School of Professional Studies (SPS)**

**The City University of New York (CUNY)**

Aron Trauring  
Adjunct Professor  
CEO, Zoteca

Class 13 December 19th, 2005

## Related Terms

- Cryptanalysis — the science of studying and breaking the secrecy of encryption processes, compromising authentication schemes and reverse-engineering protocols
- Cryptology — the study of cryptography and cryptanalysis

# Goal of Cryptography

To allow the transmission of confidential information over insecure channels without unauthorized disclosure, since obtaining the information is too work-intensive or time-consuming to be worthwhile to the attacker.

## Need for Privacy

- Competitive edge
- Reduce vulnerability
- Protect democracy

In his landmark book, *Privacy and Freedom*, Alan Westin emphasizes the relationship between access and privacy in democratic governments. Indeed, he defines democracy and authoritarianism in terms of information policy. Authoritarian governments are identified by ready government access to information about the activities of citizens and by extensive limitations on the ability of citizens to obtain information about the government. In contrast, democratic governments are marked by significant restrictions on the ability of government to acquire information about its citizens and

by ready access by citizens to information about the activities of government. Rather than being inexorably in conflict, access and privacy are both intertwined with democratic accountability.

TRANSPARENCY – THE MECHANISMS: OPEN GOVERNMENT  
AND ACCOUNTABILITY [http://usinfo.state.gov/journals/  
itdhr/0800/ijde/vaughn.htm](http://usinfo.state.gov/journals/itdhr/0800/ijde/vaughn.htm)

## Basic Definitions

- Plaintext — readable data
- Ciphertext — unreadable data
- Encryption (encrypt) — process of transforming plaintext to ciphertext
- Decryption (decrypt) — reverse of encryption
- Cryptosystem — a system that provides encryption and decryption

# Cryptosystem Components — Algorithms (Ciphers)

- Set of rules that dictate how enciphering and deciphering take place
- Rules are applied in a specific sequence
- Ancient algorithms — atbash, Caesar cipher (monoalphabetic substitution ciphers)
- World War II — German Enigma (machine with rotors)
- Modern algorithms are complex mathematical formulas
- Internal mechanism of algorithm is *not* secret

# Cryptosystem Components — Key (Cryptovvariable)

- Secret part of the cryptosystem
- Caesar cipher — how many letters to shift
- Enigma — initial setting of rotors and how the operators shifted the rotors between letters
- Modern keys are large sequence of random bits
- Keyspace — range of values that can be used to construct the key
- Larger the keyspace harder to figure out
- Encryption algorithm should use the entire keyspace and choose key as randomly as possible

# Cryptosystem Components — Protocols and Software

- Modern systems transmit of telecommunication channels
- Protocols — how the information is transmitted
- Software — tools for implementing cryptosystem

# Kerckhoff's Principle

- Paper published in 1883
- Only secrecy involved in a cryptosystem should be the key
- Algorithms should be publicly known so that vulnerabilities, weaknesses and flaws can be studied and disclosed
- Researchers support Kerckhoff's principle. Governments oppose it, so keep algorithms secret.

# NSA — National Security Agency

- As big as a Fortune 50 company
- Listen in on communications in the “interest of national security”
- Conducts research in cryptology to develop secure algorithms and break other cryptosystems

# Cryptosystem Strength

- How hard it is to figure out the secret part of the cryptosystem
- Function of power of algorithm, length of key and key secrecy
- Brute force attack — trying every possible key to break encryption
- Work factor — estimate of effort and resources it would take an attacker to penetrate a cryptosystem using brute force attack
- Flaws in algorithm can make easier to break
- Not using entire key space or non-random choice of keys also weaken cryptosystem
- Not protecting the key is a fatal flaw of many cryptosystems

# Low-Tech Cryptography — One-Time Pad

- Gilbert Vernam — 1917
- Encode plaintext message in bits and XOR with pad that has random bits
- Pad must be as long as message
- Recipient must receive uncompromised pad along with message
- Pad must be truly random
- Can prove that not subject to mathematical attack

# Low-Tech Cryptography — Running Ciphers

- Algorithm is an agreed upon set of objects, e.g. books
- Key — e.g. book page, line count, and column count in running order of books
- Simple yet hard to break

# Low-Tech Cryptography — Concealment Ciphers

- Message within a message
- Algorithm — pick out words from message using key
- Key — e.g. every third word
- Much easier to break since can try multiple alternatives

# Low-Tech Cryptography — Steganography

- Hiding data in another media type so that the very existence of the data is concealed
- Can be in graphics or sound file or in unused part of hard disk
- Algorithm — pull bit from each byte in file
- Key — put significant data in least significant bit

# Cryptosystem Services

- Modern cryptosystems have multiple uses and cryptography techniques may vary accordingly)
- Confidentiality — Denies unauthorized parties access to data (e.g. military systems)
- Authenticity — Validates the source of the message to ensure that the sender is properly identified (e.g. legal systems)
- Integrity — provides assurance that the message was not modified, accidentally or intentionally (e.g. financial systems)
- Nonrepudiation — Establishes that a particular sender has sent the message so that they can't deny it at a later date (e.g. legal systems)

# Symmetric Cryptography

- Sender and receiver use same key for encryption and decryption
- Can provide confidentiality and integrity

# Symmetric Cryptography — Advantages

- Relatively fast
- Hard to break if use a large key size

## Symmetric Cryptography — Disadvantages

- Must get key to other person out-of-band i.e. not through the unsecured transmission channel
- Each pair of users need a unique, so as number of users increase, key management becomes complicated —  $N(N-1)/2$  keys required
- Can't provide authentication and non-repudiation (since two people have same key)

# Symmetric Cryptography — Cipher Types Definitions

- Substitution ciphers — replaces bits, characters or blocks of characters with different bits, characters or blocks
- Transposition ciphers — rearranges the bits, characters or blocks of characters to hide original meaning
- Frequency analysis — using well known facts about language can break simple ciphers (e.g. e most frequently occurring letter)

# Symmetric Cryptography — Cipher Types Process

- Modern symmetric algorithms use complex mathematical formulas to do both substitution and transposition
- Algorithms indicate how substitution and transposition *can* take place
- The key tells algorithm exactly how these processes *will* take place and in what order
- Assume algorithm has 16 different sets of rules that determine substitution and transposition
- If always done in same order easy to break
- Key randomly chooses which set of rules, in what order rules are to be applied and with what values to apply rules

## Symmetric Cryptography — Cipher Attributes

- Purpose of ciphers is making the relationship between the key and ciphertext as complex as possible so that the key can't be recovered from the ciphertext
- Confusion and diffusion are key attributes of ciphers
- Confusion is provided by substitution algorithms — bits are substituted for other bits making it difficult to backtrack
- Diffusion is provided by transposition algorithms — bits are scrambled around or diffused through original message
- Confusion — Ciphertext value should depend on several parts of key but this mapping seems to be random
- Diffusion — Single plaintext bit should change many ciphertext values (best if one bit changes about half the other bits)

# Symmetric Cryptography — Cipher Chunks

- Block ciphers — work on discrete fixed sizes of bits i.e. blocks of data
- Stream ciphers — performs operations on each bit one at a time

# Symmetric Cryptography — Block Ciphers

- Message broken into fixed-sized parts
- Each part is passed through a series of substitution and transposition algorithms
- Key determines what functions are applied and in what order
- Key ensures the process is random
- Don't require as much processing power and are usually implemented in software

# Symmetric Cryptography — Stream Ciphers

- Keystream generator — a “random” stream of bits which is XOR’ed with original message
- Similar to one-time pads
- Key ensures the randomness of the keystream generator

# Symmetric Cryptography — Requirements of Effective Stream Ciphers

- Long Periods of no repeating patterns within keystream values
- Statistically unpredictable keystream
- Statistically unbiased keystream (as many 0s as 1s)
- A keystream not linearly related to key
- High level of randomness required along with need to encrypt one bit at a time require great deal of processing power
- Often implemented in silicon since software may be too slow

# Symmetric Cryptography — Initialization Vectors

- Random values that are used with algorithms to ensure that patterns are not created during the encryption process
- If not used two identical plaintext values encrypted with same key will create same ciphertext
- Patterns can be used to make it easier to break key

# Symmetric Cryptography — Cryptography Notation

- Cipher-w/r/b
- Cipher — cipher name
- w — Word (block) size, in bits, which can be 16, 32 or 64 bits in length
- r — Rounds (how many times the cryptographic functions are applied)
- b — Bytes in key, from 0 to 255 (e.g. 16 bytes = 128 bit)
- Configured for specific implementations — give developers flexibility

# Symmetric Cryptography — Modes of Operation

- Symmetric block ciphers often have different modes of operation
- Mode specifies how the cipher will operate
- Different modes are used for different situations with different requirements
- Need to understand modes so know which one to use for which purpose

# Symmetric Cryptography — Electronic Code Book Mode (ECB)

- Block entered into algorithm with key and ciphertext produced
- For given block of plaintext and given key, same block and key always produce same ciphertext
- Advantage — very easy and fast
- Disadvantage — patterns may arise so makes it easier to break
- Good for encrypting small amounts of data or where need to encrypt blocks of data independently (e.g. in a database)

# Symmetric Cryptography — Cipher Block Chaining Mode (CBC)

- Uses not only key and block of plaintext, but also result of previous block
- XOR previous block with new block before encrypting new block
- Each block is thereby dependent on all blocks before it (in a chain)
- First block is XORed with an initialization vector to ensure no patterns at start
- Advantage — removes patterns
- Disadvantage — slower, can't be used for encrypting independent data (e.g. database)

# Symmetric Cryptography — Cipher Feedback Mode (CFB)

- Block cipher which emulates stream cipher
- Useful when sending small blocks of data, e.g. in a client-server situation
- Key and IV are used to generate an e.g. 8-bit keystream
- Keystream is XOR-ed with plaintext to create 8-bit ciphertext
- Ciphertext is sent to destination and used with key to create next keystream
- Similar to CBC except with smaller blocks

# Symmetric Cryptography — Output Feedback Mode (OFB)

- Use keystream instead of ciphertext to generate next keystream
- Less likely to cause errors (encryption process may corrupt text)

# Symmetric Cryptography — Counter Mode (CM)

- Uses a new IV instead of ciphertext or keystream
- IV is incremented for each new block
- Since no chaining can encrypt blocks in parallel — hence faster than CBC, CFB or OFB
- Useful in encrypted networks where packets may arrive out of order — no chaining means can decrypt before get all packets

# Symmetric Cryptography Algorithms — Data Encryption Standard (DES)

- National Institute of Standards and Technology (NIST) began a search in '60s
- Settled on IBM's Lucifer algorithm in 1974
- Original key was 128, but NSA shortened it to 56 bits
- Data Encryption Algorithm became national standard in 1977 and ANSI standard in 1978
- Symmetric block algorithm — uses 64 bit blocks that go through 16 rounds of transposition and substitution functions
- DES was used throughout industry and government and became embedded in many products

- In 1986 NSA announced it would no longer use DES after January, 1998
- In 1998, the EFF used a \$250,000 grid computer that broke DES using brute force in three days

# Symmetric Cryptography Algorithms — Triple DES (3DES)

- 2DES has an attack that reduces it to equivalent of DES and is so equally useless
- 3DES was a quick fix until a new standard could be developed
- Uses 48 rounds but thereby takes three times as long
- DES-EEE3 — uses 3 different keys and encrypts three times
- DES-EDE3 — uses 3 different keys and encrypts, decrypts and encrypts (decryption just introduces randomness)
- DES-EEE2 — uses only two keys, first and third encryption use same key

- DES-EDE2 — uses only two keys, first and third encryption use same key
- Used in `openssh` for session key

# Symmetric Cryptography Algorithms — Advanced Encryption Standard (AES)

- NIST announced search in January, 1997
- Rijndael — created by Joan Daemen and Vincent Rijmen
- Three key and block sizes
- 128-bit key and block goes through 10 rounds
- 192-bit key and block goes through 12 rounds
- 256-bit key and block goes through 14 rounds
- Used to encrypt sensitive (but non-classified) U.S. government data

# Symmetric Cryptography Algorithms — International Data Encryption Algorithm (IDEA)

- Divides 64-bit blocks into 16 smaller blocks
- Each small block has 8 rounds performed on it
- 128-bit key
- Faster than DES in software yet more secure
- No known attacks
- Patented so restricted use

# Symmetric Cryptography Algorithms — Blowfish

- Designed by Bruce Schneier
- 64-bit blocks
- Key up to 448 bits
- 16 rounds
- Option in `openssh` for session key

# Symmetric Cryptography Algorithms — RC4

- Designed by Ron Rivest of RSA in 1987
- Variable key size
- Stream algorithm — very fast, simple and efficient
- Source code released so no longer a trade secret
- RC4 name is trademarked so often implemented as ARC4

# Symmetric Cryptography Algorithms — RC5

- Block cipher designed by Ron Rivest
- Blocks of 32, 64 or 128 bits
- Key up to 2048 bits
- RC6 is a variant designed for AES competition

# Asymmetric Cryptography — Public Key Cryptosystems

- Public key, private key pairs
- Can't use same key for encryption and decryption
- If use private key for encryption also provides authentication — only owner has private key
- Secure message format — encrypt using other person's public key so only s/he can decrypt
- Open message format — encrypt using your private key for authentication (but then anyone can decrypt)

# Asymmetric Cryptography — Advantages

- Secure key distribution — no problem publishing public key
- Scales well — only need to distribute one public key no matter how many people converse with
- Can be used for authentication and nonrepudiation

# Asymmetric Cryptography — Disadvantages

- Works much more slowly than symmetric algorithms since need to compensate for public key

# Asymmetric Cryptography — Digital Signatures

- Zero knowledge proof — prove who I am without giving up important information
- Signature generation — using private key can encrypt and generate signature (open message format)
- Signature verification — using public key can verify signature (since only owner of private key can generate)
- Provides authentication and non-repudiation

# Hybrid Cryptosystems

- Want to combine the advantages of symmetric and asymmetric systems
- Use asymmetric for key distribution and authentication
- Use symmetric for confidentiality

# Hybrid Cryptosystems — Distributing a Symmetric Key

- A encrypts message with symmetric key  $S$
- A encrypts symmetric key  $S$  with B's public key  $P$
- A sends ciphertext and encrypted key to B
- Only B can decrypt the key (as owner of private key — secure message format)
- B uses decrypted  $S$  to decrypt message

## Hybrid Cryptosystems — Session Keys

- If reuse symmetric keys over and over likely to be compromised
- Best if can use a symmetric key only once
- By using public-key authentication to create a “session” can exchange a symmetric key for that session only
- Used in OpenSSH — authentication with asymmetric keys, encryption of data with session key

# Asymmetric Cryptography — One Way Functions

- Easier to compute in one direction than in another
- Easy to break a glass — very hard to put it back together
- When encrypt using a public key message encoded with a one-way function
- Function supplies a trapdoor — mechanism for decoding
- Trapdoor only useful if know what it is and how to take advantage of it — the private key provides this information

# Asymmetric Cryptography — Diffie-Hellman Algorithm

- Whitfield Diffie and Martin Hellman
- First PKA system
- Can only be used for key exchange
- T sends E her public key  $x$
- E sends T her public key  $y$
- T computes  $K = X^{**}y \text{ mod } n$
- E computes  $K = Y^{**}x \text{ mod } n$
- The  $K$ s will be the same and can be used as a symmetric key

# Asymmetric Cryptography — RSA

- Ron Rivest, Adi Shamir and Leonard Adelman
- Can be used for standard encryption for confidentiality
- Can be used for key exchange protocol
- Can be used for digital signatures
- One way function — based on factoring a large integer into two prime factors

# Asymmetric Cryptography — RSA Algorithm

- Choose two large prime numbers  $p$  and  $q$
- Generate the product of these two numbers  $n = p * q$
- Choose a random number  $e$  to be the encryption key.  $e$  must be relatively prime with  $(p-1) (q-1)$
- Compute the decryption key  $d = e^{-1} \text{ mod } ((p-1) (q-1))$
- The public key is  $(n, e)$
- The private key is  $d$
- To encrypt message  $m$  with public key use  $C = m^e \text{ mod } n$
- To decrypt use  $M = c^d \text{ mod } n$

# Asymmetric Cryptography — Other Algorithms

- El Gamal — uses logarithms in a finite field. Slowest of all.
- Elliptic Curve Cryptosystem — most efficient so used in limited devices (e.g. cell phones)
- LUC — Lucas sequences
- Can all be used for encryption, key exchange and digital signatures
- Message Integrity
- Parity bits and cyclic redundancy checks are used for detecting unintended modifications
- If message intercepted can be changed and parity corrected

- Hashing algorithms are used to detect intentional and unintentional changes

# One Way Hash

- A function that takes a variable length message and calculates a fixed length value
- The fixed length value is called message digest, fingerprint or hash value
- Hash value is appended to message
- Recipient receives message and recalculates hash
- Compares two values to see they are the same
- If message is changed and altered can change hash as well
- If publish hash then can be used to verify integrity

- Often used to verify integrity of distributed code to ensure it hasn't been compromised

# Hash Message Authentication Code (HMAC)

- Sender concatenates a symmetric key with the message
- The result is put through a hashing algorithm
- Receiver repeats same steps and compares hash values
- Interceptor needs symmetric key to alter hash

## CBC-MAC

- Sender encrypts message with a symmetric block algorithm in CBC mode
- The last block is used as the MAC
- The plaintext message and the appended MAC are sent to the receiver
- Receiver repeats same steps and compares last block

# Good Hash Algorithms

- Compute over the entire message
- One-way functions so that message not disclosed by hash value
- Given a message and a hash value, computing another message with the same hash would be impossible (collision free)
- Should be resistant to attacks

# Good Hash Algorithms — Message Digest (MD) Family

- Created by Ron Rivest — generates a 128-bit hash
- MD2 — very slow
- MD4 — relatively fast
- MD5 — more complex than MD4, harder to break
- HAVAL — variable length modification of MD5

# Good Hash Algorithms — Secure Hash Algorithm Family

- Designed by NIST and NSA to be used with Digital Signature Standard
- SHA-1 — 160 bit
- Newer: SHA-256, SHA-384 and SHA-512

# Asymmetric Cryptography — Digital Signature Standard

- NIST standard from 1991 (DSS)
- Message is hashed using SHA – ensures integrity of message
- Private key is used to sign hash — authenticates through digital signature
- DSA — PKA which can be used only for digital signatures
- Can also use RSA to digitally sign

## Public Key Infrastructure (PKI)

- Programs, data formats, procedures, communication protocols, security policies and public key cryptographic mechanisms
- Working in a comprehensive manner to enable a wide number of dispersed people to communicate securely
- PKI establishes a level of trust
- Provides authentication, non-repudiation, integrity and confidentiality for exchanged messages
- Uses hybrid cryptosystems

## PKI — How?

- Each person who wants to participate in a PKI has to have a digital certificate
- Digital certificate contains public key and other identifying information
- The certificate is created and signed by a trusted third party — the certificate authority (CA)
- When the CA signs the digital certificate it binds the individual's identity to the public key
- CA takes liability for the authenticity of the individual
- Based on third parties trusting CA

## PKI — Certificate Authority

- Responsible for creating and handing out certificates
- Maintaining certificates
- Revoking if necessary (CRL — Certificate Revocation List)
- Can be revoked if private key compromised or wrongly issued in first place
- Online Certificate Status Protocol (OCSP) — a protocol to work with CRL to actually check current status
- CAs can be within an organization
- If cross organization must either have trusted third party or do cross certification

# PKI — Certificates

- Standard X.509 version 4
- Version number
- Serial number — unique number
- Algorithm information – used to sign certificate
- Issuer
- Validity dates
- Subject — name of owner
- Public key of owner

- ID of issuing CA
- ID of subject
- Optional information

## PKI — Key Management

- Critical part of cryptosystem since poor management of keys can compromise whole system
- Key should be long enough to provide necessary level of protection
- Keys should be stored and transmitted through secure means
- Keys should be extremely random and key algorithm should use entire keyspace
- Keys lifetimes should be related to sensitivity of data — more sensitive the shorter lifetime
- The more a key is used the shorter its lifetime should be

- Keys should be backed up or escrowed in case of loss or emergency
- Key recovery should involve more than one person from different departments
- Keys should be properly destroyed when their lifetime expires

## Web of Trust

- Alternative to large infrastructure
- A and B trust each other
- A trusts C and D and signs their keys
- B can now trust C and D

# Cryptography Attacks

- Active attacks — altering messages, modifying system files and masquerading as another individual
- Passive attacks — gathering information for an active attack

# Cryptography Attacks — Ciphertext Only Attack

- Use sniffing to gather several ciphertexts all encrypted with same system
- Goal is to discover key
- Extremely difficult to carry out successfully

# Cryptography Attacks — Known Plaintext Attack

- Attacker has plaintext and ciphertext
- Goal is to discover key
- Messages often start and end with same text
- Use reverse engineering, frequency analysis and brute force to figure out rest

# Cryptography Attacks — Chosen Plaintext Attack

- Attacker tricks someone into encrypting a known message
- More helpful than known plaintext since can develop message to reveal more information
- Requires good social engineering

# Cryptography Attacks — Chosen Ciphertext Attack

- Attacker chooses ciphertext to decrypt and has access to resultant plaintext
- Very difficult to engineer but obviously very useful

# Cryptography Attacks — Differential Cryptanalysis

- Takes two blocks of plaintext and follows changes as they go through the different rules
- The difference between them is used to map probability for possible keys
- Used to break DES

## Side Channel Attacks

- Instead of trying to directly break key look at surrounding environment
- e.g measure power consumption, radiation emissions and the time it takes for certain processing
- Used to reverse engineer the process to uncover key
- Used to break RSA private keys
- GNU Privacy Guard (GnuPG)
- Based on OpenPGP standard
- GPL equivalent of Pretty Good Privacy (PGP)

# GnuPG Key Pairs

- `gpg --gen-key`
- Default key size is 1024. Can be up to 2048.
- Name and email address is how your key is indexed in public key servers
- Move mouse around as generates key as GPG uses random signals from keyboard and mouse
- Keep a backup copy of your keys

# GnuPG Revocation Certificate

- If lose keys or someone gets access to your private key
- Can use to revoke from public key servers
- `gpg -output <filename> -gen-revoke <userid>`
- user is email address of
- Keep in separate place from private key

# GnuPG Public Server Publication

- `gpg --keyserver <server> --send-key <user>`

# GnuPG Encryption

- `gpg --output <file.gpg> --encrypt --recipient <friends email> <filename>`
- Need friend's public key

# GnuPG Decryption

- `gpg --output <file.gpg> --decrypt <filename>`
- Need private key and pass phrase