Cryptography

John Slankas
John_Slankas@ncsu.edu

Computer Science
NC STATE UNIVERSITY

Why Use Cryptography?

- Authentication
- Non-repudiation
- Confidentiality
- Integrity

What not to do …

COMMON WEAKNESSES

CWE-319: Cleartext Transmission of Sensitive Information

CWE-319: Cleartext Transmission of Sensitive Information

Summary			
Weakness Prevalence	Medium	Consequences	Data loss
Remediation Cost	Medium	Ease of Detection	Easy
Attack Frequency	Sometimes	Attacker Awareness	High

- Condition: Your software sends sensitive information across a network, such as private data or authentication credentials, that information crosses many different nodes in transit to its final destination.
- Consequence: Attackers can sniff this data right off the wire

CWE-327: Broken or Risky Cryptographic Algorithm

CWE-327: Use of a Broken or Risky Cryptographic Algorithm

Summary			
Weakness Prevalence	High	Consequences	Data loss Security bypass
Remediation Cost	Medium to High	Ease of Detection	Moderate
Attack Frequency	Rarely	Attacker Awareness	Medium

- Condition: The use of a non-standard algorithm is dangerous because a determined attacker may be able to break the algorithm and compromise whatever data has been protected.
- Consequences: Well-known techniques may exist to break the algorithm.

http://cwe.mitre.org/data/definitions/327.html

CWE-256: Plaintext Storage of a Password

```
...
Properties prop = new Properties();
prop.load(new FileInputStream("config.properties"));
String password = prop.getProperty("password");
DriverManager.getConnection(url, usr, password);
...
```

Anyone who can get to config.properties can get password

Anyone who can get to registry key can get password

```
...
String password = regKey.GetValue(passKey).toString();
NetworkCredential netCred = new NetworkCredential(username,password,domain);
...
```

http://cwe.mitre.org/data/definitions/256.html

Computer Science
NC STATE UNIVERSITY

CWE-321: Use of Hard-coded Cryptographic Key

The use of a hard-coded cryptographic key significantly increases the possibility that encrypted data may be recovered.

```
String key = "Bar12345Bar12345123456789OABCDEF";
String cipherText = encrypt(message, iv, key);
```

http://cwe.mitre.org/data/definitions/256.html

Computer Science
NC STATE UNIVERSITY

Kerckhoff's Cryptography Principles

- The system should be, if not theoretically *unbreakable*, unbreakable in practice.
- The design of a system *should not require secrecy*, and compromise of the system should not inconvenience the correspondents.
- The key should be memorable without notes and should be easily changeable.
- The cryptograms should be transmittable by telegraph.
- The apparatus or documents should be portable and operable by a single person
- The *system should be easy*, neither requiring knowledge of a long list of rules nor involving mental strain

Computer Science
NC STATE UNIVERSITY

Encryption

- **Encryption** is the process of encoding a message/data so its meaning is not obvious
- **Decryption** is the reverse process, transforming an encrypted message/data back into its normal, original form
- **Plaintext**: original message/data
- **Ciphertext:** encrypted message/data

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

Encryption

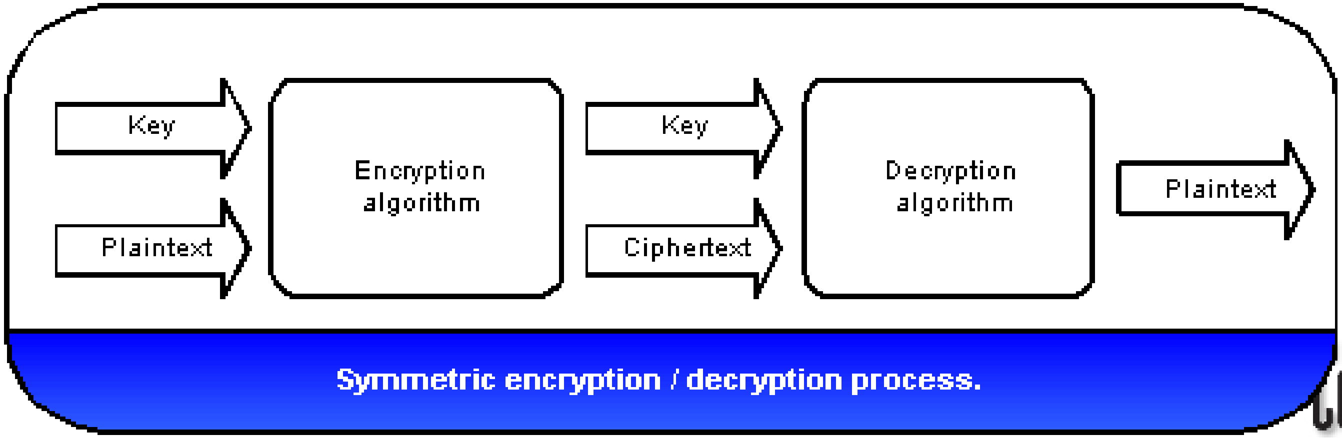
- **Algorithm:** set of rules for encryption and decryption
- **Key:** key specifies the particular transformation of plaintext into ciphertext, or visa versa during decryption

$$C = E(K,P)$$
$$P = D(K,C)$$

Computer Science
NC STATE UNIVERSITY

Symmetric Algorithms

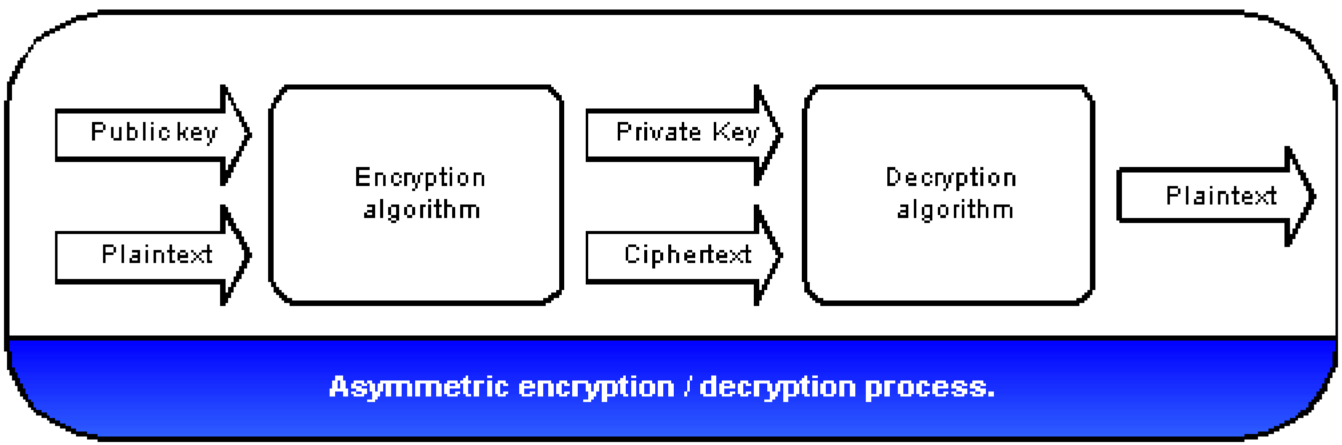
- Same key to encrypt and decrypt data.
- Generally works fast
- Typical uses: secrecy and integrity of data, messages, files
- Examples:
  - DEA (Data Encryption Algorithm) which is specified within the DES (Data Encryption Standard). (Obsolete)
  - Triple DES (Encrypt, Decrypt, Encrypt) 2 or 3 keys
  - AES (Advanced Encryption Standard) – US Standard
  - Skipjack, Blowfish, IDEA



http://www.spamlaws.com/encryption-algorithms.html
http://www.networksorcery.com/enp/data/encryption.htm

Asymmetric Algorithms

- Public-key cryptography
- One key encrypts data, while the other key decrypts
  (can reverse the keys)
- Keys are typically referred to as public and private
- Generally much slower than symmetric algorithms
- Typical uses: Key exchange, authentication



http://www.spamlaws.com/encryption-algorithms.html
http://www.networksorcery.com/enp/data/encryption.htm

Hash Algorithms

- Takes an arbitrarily long data and produces a fixed-size result
- Uses
  - Message integrity
  - Efficient digital signatures
  - Password verification
- Examples: MD5, SHA-1, SHA-256

Cryptography Engineering, Ferguson et al, Wiley, 2010

Hash Algorithm Properties

- Easy to compute
- One way: infeasible to regenerate message
- Change in Message -> Change in Hash
- Infeasible to find two different messages with the same hash

Computer Science
NC STATE UNIVERSITY

Choosing an Encryption Algorithm

- There is no right answer
- How long must the data be protected?
- How long must the encryption/decryption process take?
- Use standard algorithms
- Keys:
  - Longer key: stronger encryption
  - Longer key: more processing time
- Combine symmetric and asymmetric algorithms to get the best of both options

http://technet.microsoft.com/en-us/library/ms345262.aspx

For What Security Property?

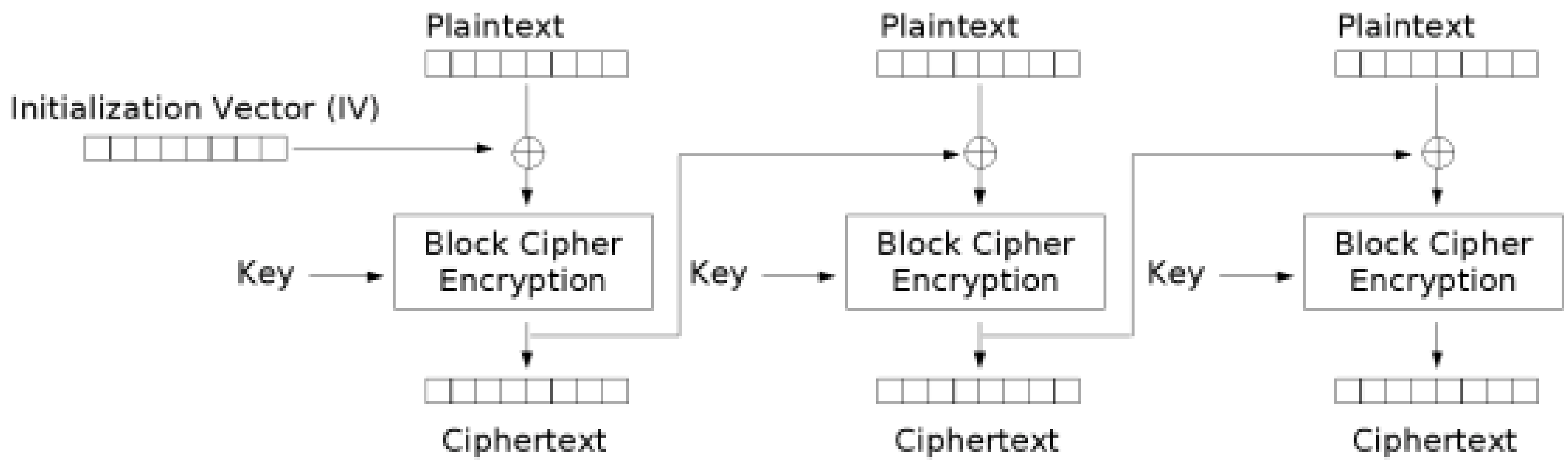
Algorithm	Confidentiality	Authentication	Integrity	Key Management
Symmetric encryption algorithms	Yes	No	No	Yes
Public-key encryption algorithms	Yes	No	No	Yes
Digital signature algorithms	No	Yes	Yes	No
Key-agreement algorithms	Yes	Optional	No	Yes
One-way hash functions	No	No	Yes	No
Message authentication codes	No	Yes	Yes	No

Applied Cryptography, 2nd Ed, Bruce Schneier, 1996, Table 10.1

Using Encryption Wisely

- Use standard algorithms
- Educate yourself
  - What are the strengths and weaknesses?
  - Block cipher modes? ECB / CBC / OFB / CFB / CTR
    - Security / Efficiency / Fault-tolerance
  - Weak keys? Semi-weak keys?
- Stay current
  - DES
  - MD5, SHA1
- Protect the keys
- Rotate the keys
- Verify certificates
- Abstraction – make it easy to use

Sample Code



Cipher Block Chaining (CBC) mode encryption

Computer Science
NC STATE UNIVERSITY

Cryptography Attacks

- Ciphertext only
- Known plaintext/ciphertext pairs
- Man in the Middle
- Replaying message
- Side-channel attacks
  - Timing how long something takes to occur
  - Examining memory

Don't assume you are safe. Be paranoid

Cryptography in Applications

- Passwords
  - Hash
  - Don’t forget the salt…
- Digital signatures
- Protect confidential data
- Transfer users from one authenticated session to another
- URL / Data parameters
- Many more uses ….

SSL / TLS

- SSL (Secure Sockets Layer)
  - Standard security protocol for communications over a network (ie, the Internet)
  - Provides endpoint authentication (client optional)
  - Encrypts all communication
  - Uses both asymmetric and symetric algorithms
- For web applications with SSL, use SSL for the entire session (login to logout).
  - Why?

At the Application Layer

- Use standard APIs
  - Java Cryptographic Extension (JCE)
  - Cryptographic Application Programming Interfaces (Windows)
  - Mac OS X Security Services
- Database layer

Encryption Functions - MySQL

Table 11.17. Encryption Functions

Name	Description
AES_DECRYPT()	Decrypt using AES
AES_ENCRYPT()	Encrypt using AES
COMPRESS()(v4.1.1)	Return result as a binary string
DECODE()	Decodes a string encrypted using ENCODE()
DES_DECRYPT()	Decrypt a string
DES_ENCRYPT()	Encrypt a string
ENCODE()	Encode a string
ENCRYPT()	Encrypt a string
MD5()	Calculate MD5 checksum
OLD_PASSWORD()(v4.1)	Return the value of the old (pre-4.1) implementation of PASSWORD
PASSWORD()	Calculate and return a password string
SHA1(), SHA()	Calculate an SHA-1 160-bit checksum
UNCOMPRESS()(v4.1.1)	Uncompress a string compressed
UNCOMPRESSED_LENGTH()(v4.1.1)	Return the length of a string before compression

http://dev.mysql.com/doc/refman/5.1/en/encryption-functions.html

Password / Key Storage

- Wrong answer: hard-coded
- Configuration file
  - Proper permissions
  - Encrypted in configuration files
    - Hides values, still need a decryption key
- Human operator
- Monitor use of keys/passwords