NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

3010 Lecture Week 11

RSA, PKC & Crypto Failures

Dr Tay Kian Boon

Number 2 on Top 10 OWASP

A02:2021 – Cryptographic Failures

OWASP Top 10 Vulnerabilities

- OWASP a professional org that monitors security failures.
- (www.owasp.org)
- They maintain website: top 10 failures every few years.
- At Number 2:

- Cryptographic Failures

Cryptographic Failures

- Shifting up one position to #2, (previously *Sensitive Data Exposure*)
- Focus is on failures related to cryptography (or lack thereof).
- Often lead to exposure of sensitive data.
- Cryptography must be implemented correctly in order for data to be safe.
- Many points of possible failure in doing it correctly.

Cryptographic Failures

- Notable Common Weakness Enumerations (CWEs) included are
  - *CWE-259: Use of Hard-coded Password*,
  - *CWE-327: Broken or Risky Crypto Algorithm &*
  - *CWE-331 Insufficient Entropy*.
    - (normally link to weak random number generation for keys & IVs etc) – will share this later

Cryptography Overview

Main Crypto Ingredients

- Strong Crypto Algorithms
  - Use to protect data (can be voice, video etc…)
- Secure Hash Functions
  - Use in Digital signatures & ensuring data integrity
- Strong Random Number Generators
  - Use to generate unbiased random keys for crypto algorithm use

Crypto Algorithms

For A to talk to B securely, both need to use

- Same (strong) Crypto Algorithms (private key cryptosystem)
  - Eg AES Encryption algorithm
- Same (strong) Crypto Algorithms (public key cryptosystem)
  - Eg RSA encryption algorithm
- Keys used
  - Need to be generated from crypto secure RNG such as ISAAC, FORTUNA,…
- Protocol to send message must be robust, no leakage, no weakened entropy

Private (symmetric) Key Crypto

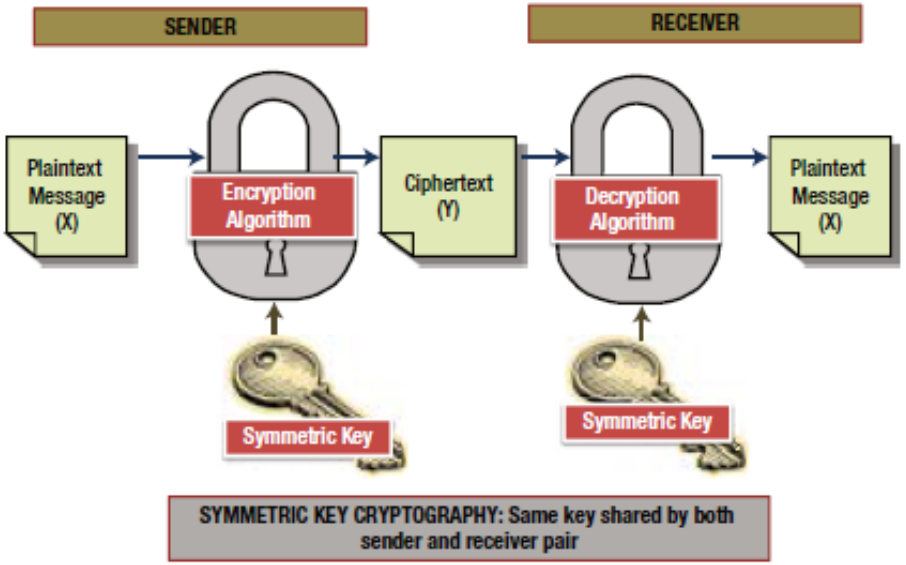

Figure 8.2 Symmetric Key Cryptography

Private (symmetric) Key Crypto

- Private (symmetric) key algorithms such as AES are good.
- Suffer from some serious drawbacks on its own:

1. How to agree on new key change (how to solve this?)
2. How to manage keys – eg storage, expiry date etc
3. How to send encrypted message to someone you don't know?
    1. Basis of ecommerce!

Answer: Public Key Crypto! 2 Keys: Public and Private!

Public Key Cryptography

Public key cryptography uses a pair of keys for encryption and decryption. A ***public key*** is used to encrypt the data and a ***private key*** is used to decrypt the data. Using the public key, anyone can encrypt the data, but they cannot decrypt the data. In this approach, both sender and receiver have the ability to generate both keys (using a computer system) together. However, only the public key is made known to the other party, who can download this key even from a web server; the private key is not known to anyone. It is not sent to the other party, hence the problem of distribution of the key never arises. In case of intrusion or any other problems, the system can generate a private key, and a corresponding public key that can be published again. The algorithms that generate keys are related to each other mathematically in such a way that knowledge of one key does not permit anyone to determine the other key easily.

Figure 8-5 illustrates how the confidentiality of a message is ensured through asymmetric key cryptography (alternatively known as public key cryptography).

Public Key crypto

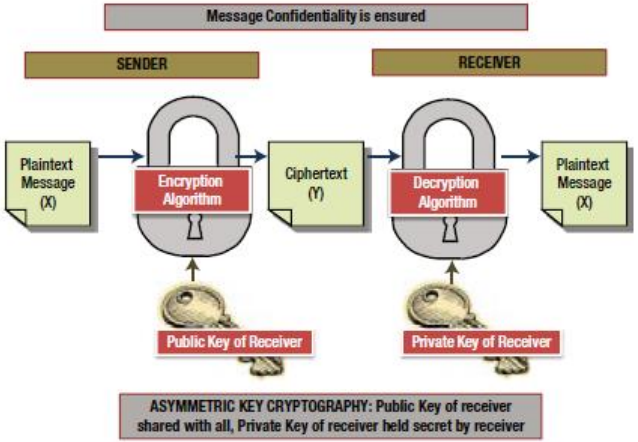

Figure 8-5. *Public Key Cryptography – How Confidentiality is ensured*

Figure 8-6 illustrates how the authenticity of the message is ensured through asymmetric key cryptography (i.e., public key cryptography).

Public Key crypto-ensure authentication

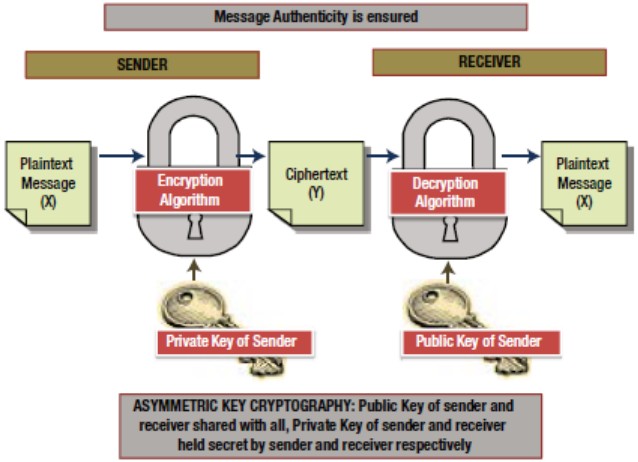

Figure 8-6. Public Key Cryptography – How Authenticity is ensured

Hash Functions

- Basically a fast one-way function to create a fixed length message digest
  - Famous example: SHA-256, SHA-512, KECCAK

- Used in data integrity, digital signatures

- Frequently when you want to download software on website, website will contain SHA(software) for integrity check
  - Apply SHA hash to software downloaded & see if both values agree

Random Number Generators (RNGs)

- Crypto algorithms need keys, and they are n-bit numbers generated by RNGs
- Good keys generated are unbiased and equally likely
- Need crypto-secure RNG to generate crypto keys

Simple Overview SSL/TLS Protocol

- Client & Server say hello to each other
- Client tell server what algos it has
- Server (normally) picks strongest algo & hash offered by Client
- They exchange info leading to establish common key
- They can start talking (securely!)

2 Types Crypto Algorithms

- Private Key (symmetric)
  - Famous eg AES (128,192, 256)
    - index number refers to key length
- Public Key (asymmetric)
  - Famous eg RSA(1024, 2048)

Strength of Private key Algo

The strength of Strong Algorithms like AES depends on

- Algo Design
- Keys generated (quality of RNG)
- Key length
- Secure implementation!

RSA

- RSA public parameter N is the product of Two k-bit prime numbers (independently generated)
  - Need good RNG to generate LARGE random odd number first
  - Need robust prime generation routines.
    - Basically find nearest prime to it, call it p
    - Generate another large random odd q. Then find nearest prime to it.
    - Form N=p*q

- Security of RSA is based on difficulty of factoring large numbers

- However there are certain RSA parameters that are weak. Such weakened RSA system can be broken easily without the need to factor the Large number N.

RSA in 1 Page

1. Choose 2 random k-bit indep primes p & q and form n =p*q.
2. Compute $\Phi(n) = \Phi(pq) = (p-1)(q-1)$.
3. Choose e, encryption exponent s.t. gcd (e, $\Phi(n)$) =1.
4. Public parameters is {n,e}.
5. Decryption exponent $d = e^{-1} \pmod{\Phi(n)}$
6. Encrypt msg M (<n) by $M^e \pmod{n} = C$
7. Decrypt cipher C by $C^d \pmod{n} = M$!

Some Weak RSA Parameters & Implementations

- Size Primes p & q ≤ 512 bits.
- p & q are not independently generated
- p & q are not randomly generated by crypto-secure RNG
- Decryption key d ≤ [N^(0.25)]/3 (recall N=p*q) (“short d”)
- p-1 is a product of only “small” prime factors
- Common use of same N for users in same company, although they use different encryption and decryption exponents.

Crypto Advice

- DO NOT TRUST DO NOT TRUST VENDORS!
- Always verify their algorithms used if you have to purchase their products
- Verify their RNG used produces truly (close to) random bits
  - NIST Suite tests randomness
  - If in doubt, use your own RNG, FORTUNA and ISAAC are good so far.