

1.264 Lecture 17

Cryptography

Premises for Internet security

- **Client-network-server are the 3 key components**
- **Client (browser or application) premises**
 - Remote server is operated by organization stated
 - Documents returned by server are free from viruses, etc.
 - Remote server will not distribute user's private info, such as Web use
- **Network premises (for both client and server)**
 - Network is free from 3rd party eavesdroppers
 - Network delivers information intact, not tampered with by 3rd parties
- **Server premises**
 - User will not attempt to break into or alter contents of Web site
 - User will not try to gain access to documents that he/she is not allowed
 - User will not try to crash the server or deny service to others
 - If user has identified him/herself, user is who he/she claims to be

Client risks

- **Active content: applets, scripts, ActiveX controls, plug-ins**
Browsers download and run software without advance notice
 - User sometimes cannot virus-check before using (zero-day attack)Formerly innocuous content such as Web pages can send viruses
Spyware, Trojan horses, etc. are prevalent (active content, cookies), often distributed by email but now via Web pages
Solutions are less use of active content, virus checkers (new technology needed), intrusion detection, XML-based apps
- **Privacy loss**
Web server site logs: IP address, document retrieved, date/time, previous URL, and more.
Cookies. Have been abused by marketing to track user habits
Email. Spam.
Solutions are spam filters, intrusion/extrusion detection, trust standards, email verification (IP addresses)
- **Server is being spoofed (phishing, other attacks)**
Confidential information sent to unauthorized party
Solution: education, IP improvements, certificates...

Server risks

- **Webjacking**
 - Break-in and modification of site**
 - Thousands have occurred, including most major corporations...
 - Database theft is most serious risk
 - Exploit operating system and email holes, poor configuration, poor passwords, ...**
 - Solutions are patch management, OS testing, good server mgt**
- **Denial of service**
 - Attacks that cause system to expend large resources in response**
 - Distributed denial of service attacks**
 - Solutions are distributed filters, identification of attacking servers, changes in Internet protocols to limit spoofing**
- **Major current risks:**
 - Takeover of PC or server to be used as spam server**
 - Revenue of about \$0.01 per spam email sent
 - Takeover of PC or server to be used for denial of service attacks**
 - Extortion by organized crime

Network risks

- **Packet sniffers: look for passwords, credit card numbers**
 - Kits available on Internet
 - Shared public networks (cable modem, WiFi) are major risk
 - Small programs, installed on compromised computer in network
 - Cryptography is technical solution to sniffing
- **IP address spoofing: pretend you're another machine**
 - Look-alike sites set up with stolen pages, etc.
 - Mimic Web merchant, bank, etc.
 - Phishing, social engineering attacks
 - Email with false URLs, from which credit cards are harvested
 - Sites with popups or pages with viruses
 - Authentication (digital certificates) is technical solution
 - Other solutions: education (limited success), no active content
- **Tampering is rare; denial of service is common**

Overall issue

- **Trust and identity**
 - **User identity is only to their email address in computer security**
 - **Actual identity is most often established using credit card number for users**
 - **Trading partner trust is not based on encryption, certificates, etc. but on knowledge of each other from face-to-face business dealings**
 - **Computer security is a minor, though crucial, part of trust and identity**
 - Global computer-based trust and identity seems impossible**
 - Trust and identity are context-specific**

Cryptographic primitives

- **Symmetric key encryption**
- **Asymmetric (public) key encryption**
- **Stream or block ciphers (apply key to message)**
- **Message digests (hashes)**
- **Digital signatures (certificates)**
 - Covered in next lecture

Managing network risks: Cryptography

- **Definitions**

Plaintext: original message

Ciphertext: encrypted message

Algorithm: function converting plaintext to ciphertext

Key: number used by algorithm to encrypt and/or decrypt

- Not the same as a database key (primary or foreign!)

- **Encryption process**



Symmetric: sender and receiver use same secret key

Asymmetric: sender and receiver use different, but related keys.

Receiver key public, used by all senders to that receiver

Symmetric encryption

- **Symmetric algorithms use same key to encrypt and decrypt**

DES (Data Encryption Standard): 56 bit key, in common use

- Splits data into pieces, XORs, reshuffles
- Cracked in two days in June, 1998

Triple DES: encrypt/decrypt/encrypt with 3 DES keys: 168 bit effective key length

- Backward compatible with DES in banking, etc.

RC2, RC4, RC5: 40-2048 bit keys, in common use by encrypting Web servers and browsers

AES: Current US government standard , uses Rijndael algorithm

- **Problems with symmetric keys**

Must be exchanged in advance, via secure method

Multiway communication infeasible:

- If many users must communicate with server, compromising any one can compromise all

Exercise (very simplified from real thing!)

- Plaintext: 73628495
- Key: 31
 $k_1 = 3, k_2 = 1$
- Sender algorithm:
 Shift digits by k_1 to the left
 Subtract k_2 from each digit
- Ciphertext: _____
- Receiver algorithm:
 Shift digits by k_1 to the right
 Add k_2 to each digit
- Plaintext: _____

(Real symmetric algorithms chop, shift, add/subtract in complex patterns to remove statistical patterns in data—see text: S-boxes)

Solution

- Plaintext: 73628495
- Key: 31
 $k_1 = 3, k_2 = 1$
- Sender algorithm:
 Shift digits by k_1 to the left
 Subtract k_2 from each digit
- Ciphertext: 28495736 -> 17384625
- Receiver algorithm:
 Shift digits by k_1 to the right
 Add k_2 to each digit
- Plaintext: 62517384 -> 73628495

(Real symmetric algorithms chop, shift, add/subtract in complex patterns to remove statistical patterns in data)

Asymmetric or “public key” encryption



- **Key pairs: public key for encryption, private key for decryption**
 - RSA: 512-1024 bit, in common use for Web and email**
 - Patent expired in 2005**
- **Problem with public key algorithms**
 - Speed: RSA is 1000 times slower than symmetric algorithms**
 - **Problem avoided by using RSA to exchange a symmetric session key and then using symmetric encryption method for the rest of the session.**
 - **Use a different symmetric key each session to limit damage if key is broken**

Public key (RSA) concept

- Public key P is pair of integers (N, p)
- Secret or private key S is pair of integers (N, s)
- Generate 3 large random prime numbers (Fermat's Little Thm)
Largest is s . Call the other two x and y .
 $N = xy$
 $p =$ smallest integer such that $(ps) \bmod (x-1)(y-1) = 1$
- Break message into a series of chunks m_i
- Encrypt message chunk m_i to ciphertext chunk c_i by:
 $c = m^p \bmod N$
- Decrypt ciphertext chunk c_i to plaintext m_i by:
 $m_i = c^s \bmod N$
- s is hard to compute from N and p
Requires knowledge of x and y , which requires factoring N
Factoring is exponential time algorithm, so if the number to be factored is big enough, it takes a very long time...

Exercise (again, very simplified)

- Code ATTACK using: A=01, B=02, C=03, etc. (K=11,T=20):

- Generate 3 random primes: 47, 79, 97 (way too small in real life!)
- Use $s = _$, $x = _$, $y = _$. Verify that $p = 37$ using
 $(ps) \bmod (x-1)(y-1) = 1$
- Compute $N = xy$: _____
- Break the message into three 4-digit chunks:

- Create ciphertext: raise each chunk to the p power % N :

- Retrieve plaintext: raise each chunk to s power % N :

$$0311^{37} \% 3713 = 3536, 2001^{37} \% 3713 = 2932, 0120^{37} \% 3713 = 1404$$

$$1404^{97} \% 3713 = 0120, 2932^{97} \% 3713 = 2001, 3536^{97} \% 3713 = 0311$$

Solution

- Code ATTACK using: A=01, B=02, C=03, etc. (K=11,T=20):
 - 012020010311
- Generate 3 random primes: 47, 79, 97
- Use s= 97, x= 47, y= 79. Verify that p= 37 using
 $(ps) \bmod (x-1)(y-1) = 1$ $37*97= 3589$. $46*78= 3588$.
- Compute N= xy: 3713
- Break the message into three 4-digit chunks:
 - 0120 2001 0311
- Create ciphertext: raise each chunk to the p power % N:
 - 1404 2932 3536
 $0311^{37} \% 3713=3536$, $2001^{37} \% 3713= 2932$, $0120^{37} \% 3713= 1404$
- Retrieve plaintext: raise each chunk to s power % N:
 - 0120 2001 0311
 $1404^{97} \% 3713=0120$, $2932^{97} \% 3713= 2001$, $3536^{97} \% 3713= 0311$

Key length

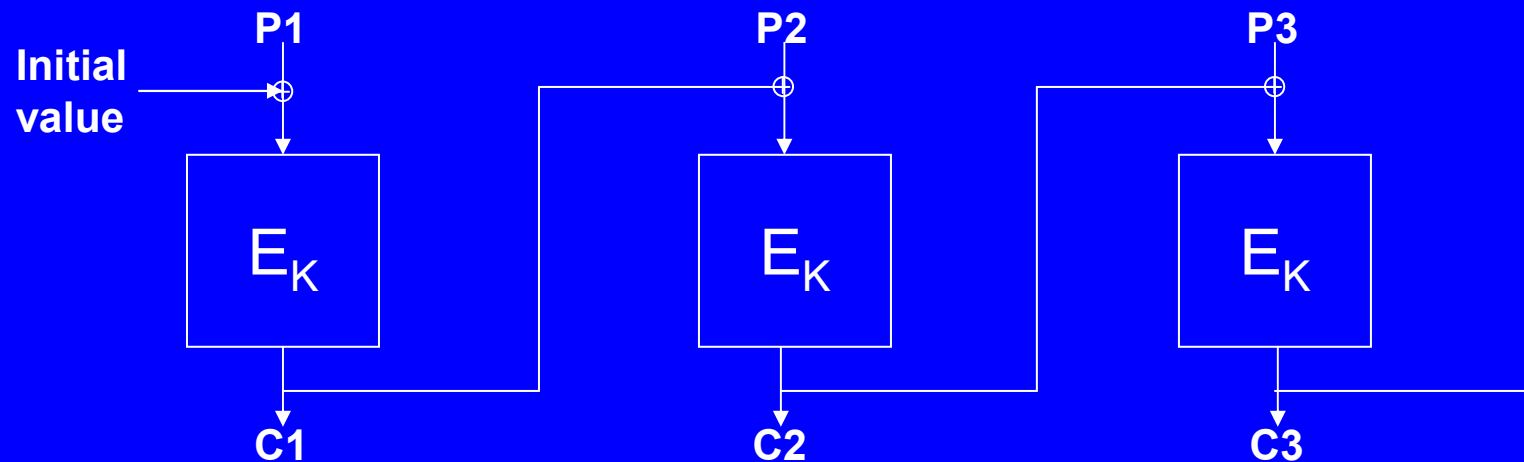
- **Assume:**
 - Algorithm is good
 - Algorithm coded correctly
 - Key management is correct and secure
- **Then only way to crack message is brute force**
 - Public key lengths must be longer than symmetric to provide same level of security
 - 384 bit public key offers same security as 40 bit symmetric key (not much)
 - Public keys should be at least 1024 bits.
 - Symmetric keys should be at least 128 bits, going to 256 bits soon

<u>Symmetric key length</u>	<u>Time to crack on PCs</u>	<u>Time to crack on servers</u>
40 bits	Seconds	Milliseconds
56 bits	Days	Hours
64 bits	Months	Days
80 bits	Millions of Years	Thousands of Years

Ciphers

- When a message is longer than the key (the usual case)
 - We exclusive-or (add bits without carrying) each block of plaintext with the previous block of ciphertext before encrypting it
 - This disguises any patterns in plaintext

Repeated plaintext is coded differently each time it appears



Message digests

- **Cryptographic hashes are a one-way function that creates a short number (128 to 160 bits, often) that is very unlikely to be generated by any other message**
 - Many hashes are the last (chained) block cipher of a message, so it depends on the entire message
 - It's used to verify that the message has not been altered
- **Common message digests:**
 - MD4: 3 rounds, 128 bit hash
 - MD5: 4 rounds, 128 bit hash
 - sha1: 5 rounds, 160 bit hash
 - sha256: 64 rounds, 256 bit hash
 - sha512: 80 rounds, 512 bit hash