Computer Security

07r. Assignment 5 review

Paul Krzyzanowski
Rutgers University
Spring 2017

Question 1

Why did the Storm worm/Trojan propagate even though Windows warns users that the app isn't signed and asks them if they really want to install it?

- This is the social engineering aspect of Trojans: people felt that they were installing a useful app that they wanted (football game tracking), so they would go through the steps necessary to install it
- Windows has a history of popping up annoying dialog boxes that people are trained to click them without reading to make them go away

Question 2

What is a polymorphic virus?

- It is a virus that re-encrypts itself with a different key each time it replicates
- It also modifies the decryption code by substituting instructions with equivalent sequences of instructions
  - E.g., add/remove NOP instructions, change adds to subtracts, invert comparisons and jumps
- Goal: bypass virus checkers that search for known patterns (virus signatures)

Question 3

What is meant by an epidemic threshold of a virus?

- When the rate of virus replication exceeds the rate at which the virus is removed
  - The virus is spreading faster than it is being removed

Question 4

In the example cited in the Panda security report, how did hackers convince people to enable macros in a downloaded word document?

- At the top of the document in bold capital letters there was a message that indicated that the image was blurred for security reasons.
- If the user wanted access to the information then they had to enable the macros, with an arrow pointing to the button to be pressed.
- Once enabled, it showed you the clear image while simultaneously infecting you with a form of Cryptolocker malware.

Question 5

Explain what *spear phishing* is.

- Phishing is an attempt to get personal information from users
- Spear phishing is a targeted form of phishing
  - Messages are designed to appear to come from someone the recipient knows and trusts
  - Subject lines & content may be specifically tailored to that user

Question 6

How did the use of Bitly links help in the email hacking of the Democratic National Committee (DNC)?

- Bit.ly made the links look harmless
- The short links hid the presence of long URLs containing the actual malicious domain and long list of parameters.
- Users rarely check the full URL associated with short links so URL-shortening services can be used to hide malicious URLs

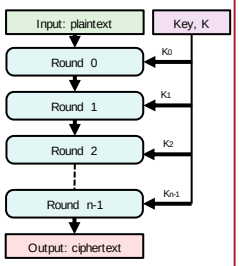
Computer Security

07r. Cryptography (continued)

Paul Krzyzanowski
Rutgers University
Spring 2017

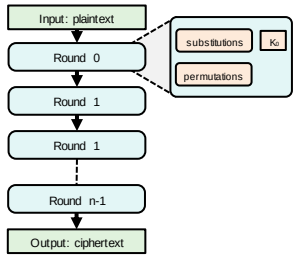
Block ciphers

- Block ciphers encrypt a *block* of plaintext at a time and produce ciphertext
- DES & AES are two popular block ciphers
  - DES: 64 bit blocks
  - AES: 128 bit blocks
- Block ciphers are usually iterative ciphers
  - The encryption process is an iteration through several *round* operations



Block cipher rounds

Each round consists of substitutions & permutations



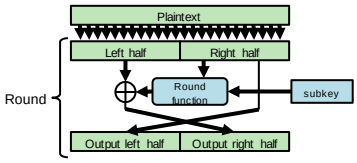
Substitution = **S-box**
- Table lookup
- Converts a small block of input to a block of output
- Changing one bit of input should change approximately ½ of output bits

Permutation
- Scrambles the bits in a prescribed order

Key application per round
- Subkey per round derived from the key
- Can drive behavior of s-boxes
- May be XORed with the output of each round

Feistel cipher

- DES is a type of Feistel cipher, which is a form of a block cipher
- Plaintext block is split in two
  - Round function applied to one half of the block
  - Output of the round function is XORed with other half of the block
  - Halves are swapped
- AES is *not* a Feistel cipher



AES (Advanced Encryption Standard)

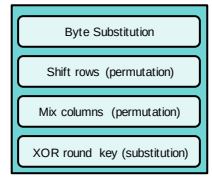
- Block cipher: 128-bit blocks
  - DES used 64-bit blocks
- Successor to DES as a standard encryption algorithm
  - DES: 56-bit key
  - AES: 128, 192, or 256 bit keys

AES (Advanced Encryption Standard)

- Iterative cipher, just like most other block ciphers
  - Each round is a set of substitutions & permutations
- Variable number of rounds
  - DES always used 16 rounds
  - AES:
    - 10 rounds: 128-bit key
    - 12 rounds: 192-bit key
    - 14 rounds: 256-bit key
  - A subkey ("round key") derived from the key is computed for each round
    - DES did this too

Each AES Round

- Step 1: Byte Substitution (s-boxes)
  - Substitute 16 input bytes by looking each one up in a table (S-box)
  - Result is a 4x4 matrix
- Step 2: Shift rows
  - Each row is shifted to the left (wrapping around to the right)
  - 1st row not shifted; 2nd row shifted 1 position to the left; 3rd row shifted 2 positions; 4th row shifted three positions
- Step 3: Mix columns
  - 4 bytes in each column are transformed
  - This creates a new 4x4 matrix
- Step 4: XOR round key
  - XOR the 128 bits of the round key with the 16 bytes of the matrix in step 3



AES

- Decryption process does the same rounds … but in reverse order

DES Disadvantages

- DES has been shown to have some weaknesses against differential and linear cryptanalysis
  - Key can be recovered using 2^{47} chosen plaintexts or 24^3 known plaintexts
  - *Note that this is not a practical amount of data to get for a realattack*
- Short block size (8 bytes = 64 bits)
- The real weakness of DES is its 56-bit key
  - Exhaustive search requires 2^{55} iterations on average
- 3DES solves the key size problem: we can have keys up to 168 bits
  - Differential & linear cryptanalysis is not effective here: the three layers of encryption use 48 rounds instead of 16 making it infeasible to reconstruct s-box activity.
- DES is relatively slow
  - It was designed with hardware encryption in mind
    - 3DES is 3x slower than DES
  - *Still much faster than RSA public key cryptosystems!*

AES Advantages

- Larger block size: 128 bits vs 64 bits
- Larger & varying key sizes: 128, 192, and 256 bits
  - 128 bits is complex enough to prevent brute-force searches
- No significant academic attacks beyond brute force search
  - Resistant against linear cryptanalysis thanks to bigger S-boxes
    - S-box = lookup table that adds non-linearity to a set of bits via transposition & flipping
  - DES: 6-bit inputs & 4-bit outputs
  - AES: 8-bit inputs & 8-bit outputs
- Typically 5-10x faster in software than 3DES

Attacks against AES

- Attacks have been found
  - This does ***not*** mean that AES is insecure!
- Because of the attacks:
  - AES-128 has computational complexity of $2^{126.1}$ (~126 bits)
  - AES-192 has computational complexity of $2^{189.7}$ (~189 bits)
  - AES-256 has computational complexity of $2^{254.9}$ (~254 bits)
- The security of AES can be increased by increasing the number of rounds in the algorithm
- However, AES-128 still has a sufficient safety margin to make exhaustive search attacks impractical

Cryptographic attacks

- Chosen plaintext
  - Attacker can create plaintext and see the corresponding ciphertext
- Known plaintext
  - Attacker has access to both plaintext & ciphertext but doesn't get to choose the text
- Ciphertext-only
  - The attacker only sees ciphertext
  - Popular in movies but rarely practical in real life

Differential Cryptanalysis

Examine how changes in input affect changes in output

- Discover where a cipher exhibits non-random behavior
  - These properties can be used to extract the secret key
  - Applied to block ciphers, stream ciphers, and hash functions (functions that flip & move bits vs. mathematical operations)
- Chosen plaintext attack is normally used
  - Attacker must be able to choose the plaintext and see the corresponding cipher text

Differential Cryptanalysis

- Provide plaintext with known differences and see how those differences appear in the ciphertext
- The properties depend on the key and the s-boxes in the algorithm
- Do this with lots and lots of known plaintext-ciphertext sets
- Statistical differences, if found, may allow a key to be recovered faster than with a brute-force search
  - You can deduce that certain keys are not worth trying

Linear Cryptanalysis

Create a predictive approximation of inputs to outputs

- Instead of looking for differences, linear cryptanalysis attempts to come up with a linear formula (e.g., a bunch of xor operations) that connects certain input bits, output bits, and key bits with a probability higher than random
  - Goal is to approximate the behavior of s-boxes
- It will *not* recreate the working of the cipher
  - You just hope to find non-random behavior that gives you insight on what bits of the key might mattes
- Works better than differential cryptanalysis for known plaintext. Differential cryptanalysis works best with chosen plaintext
- Linear & differential cryptanalysis will rarely recover a key but may be able to reduce the number of keys that need to be searched.

The end