Encryption 2

Tom Chothia
Computer Security: Lecture 3
This Lecture

• Counter Mode (CTR) encryption

• PKCS7 Padding

• Diffie Helleman key exchange

• Public Key Encryption
  – Elgaml
What I want you to know so far:

• Symmetric Key Encryption Ciphers
  – Frequency Analysis
  – One time pads
  – AES, DES and 3-DES

• Block cipher modes

• Truecrypt
Reminder: Cipher Block Chaining mode (CBC)

• Plain text = \(B_1, B_2, B_3, \ldots\)
• IV = random number (sent in clear)
• Cipher text = \(C_1, C_2, C_3, \ldots\) where
  \[C_1 = \text{encrypt}_K(B_1 \text{ xor IV}),\]
  \[C_2 = \text{encrypt}_K(B_2 \text{ xor } C_1),\]
  \[\ldots\]
  \[C_i = \text{encrypt}_K(B_i \text{ xor } C_{i-1}).\]
CBC encrypt

Cipher Block Chaining (CBC) mode encryption

CBC decrypt

• Receive IV
• Receive cipher text = $C_1, C_2, C_3, \ldots$
• Plain text = $B_1, B_2, B_3, \ldots$ where:

  $B_1 = \text{decrypt}_K(C_1) \text{ xor IV}$,
  $B_2 = \text{decrypt}_K(C_2) \text{ xor } C_1$, 
  
  \[ \ldots \]
  $B_i = \text{decrypt}_K(B_i) \text{ xor } C_{i-1}$. 
CBC decrypt

Cipher Block Chaining (CBC) mode decryption

Block Cipher Modes

Original  ECB  CBC

Probabilistic Encryption

- Probabilistic encryption schemes use random elements to make every encryption different.

- E.g. adding a little bit of random plain text data to make the cipher text random.

- This extra data is then discarded when the cipher text is decrypted.

- A random IV is a good way to make encryption probabilistic.
Sony PlayStation

Sony needs to stop games being copied.

CD & full disk encryption

User can read and write particular areas of the hard disk.
• With CBC, you need to encrypt, or decrypt, the whole file to get to the end.

• The Sony PlayStation uses ECB full disk encryption, to stop people copying games.

• User can access files they made themselves (notes, music, video …)

• Hardware controls user access to data.
Sony PlayStation Disk Encryption Attack

1. Remove disk and make a copy.
2. Write a user accessible file to the disk.
3. Find the bit of disk that changed (that’s your encrypted file).
4. Over write this with what you want to decrypt.
5. Restart the PlayStation and ask for your file back.
6. PlayStation decrypts the file and gives you the plain text.
Counter Mode (CTR)

• Plain text = $B_1, B_2, B_3, \ldots$
• IV = random number (sent in clear)
• Cipher text = $C_1, C_2, C_3, \ldots$ where
  
  $C_1 = B_1 \text{xor} \ \text{encrypt}_K(IV)$,
  
  $C_2 = B_2 \text{xor} \ \text{encrypt}_K(IV+1)$,
  
  $C_3 = B_3 \text{xor} \ \text{encrypt}_K(IV+2)$,
  
  $\ldots$

  $C_i = B_i \text{xor} \ \text{encrypt}_K(IV + i-1)$,
Counter (CTR) mode encryption

Counter (CTR) mode decryption

Padding

- Block ciphers only work on fixed size blocks.

- If the message isn’t of the right block size we need to pad the message.

- But receiver need to tell the difference between the padding and message.
Padding

• Add random bytes to the end of the block?

• Add zeros to the end of the block?

• Write “this is padding”? 
Padding: PKCS7

• If there is 1 byte of space write 01
• If there are 2 byte of space write 0202
• If there are 3 byte of space write 030303
• ...

• If the message goes to the end of the block add a new block of 16161616..
The Key Problem

- These encryption schemes work well. AES is effectively unbreakable with a “long enough key”.

- The problem is how do you get the key in the first place?
Some History

Before cheap powerful computers, unbreakable encryption was almost impossible.

Governments wanted to read the codes of others.

They could control the export of these machines.

When IBM designed DES they could get it weakened.

• Cipher machines looked like this:
Some History

During 1970-1990 all that changed.

Personal computers could do anything a cipher machine could do.

• Cipher machines looked like this:
Some History

During 1970-1990 all that changed.

Personal computers could do anything a cipher machine could do.

University academics worked on encryption with the aim of making it available to everyone.
Public Key Encryption

• Public key encryption helps (but doesn’t solve) this problem.

• The idea of public key encryption is that you have two keys:
  – one for encryption
  – and another for decryption.

• The encryption key is made public, the decryption key is always secret.
Diffie-Hellman

- Diffie-Hellman is a widely used key agreement protocol.

- It relies on some number theory:
  - \( a \mod b = n \) where for some “\( m \)”: \( a = m \cdot b + n \)

- The protocol uses two public parameters
  - generator “\( g \)” (often 160 bits long)
  - prime “\( p \)” (often 1024 bits long)
Diffie-Hellman

- Alice and Bob pick random numbers $r_A$ and $r_B$ and find
  \[ t_A = g^{r_A} \mod p \]  and \[ t_B = g^{r_B} \mod p \]

- The protocol just exchanges these numbers:
  1. $A \rightarrow B : t_A$
  2. $B \rightarrow A : t_B$

- “Alice” calculates \[ t_A^{r_A} \mod p \] and “Bob” \[ t_A^{r_B} \mod p \]
  this is the key:
  - $K = g^{r_A r_B} \mod p$
Diffie-Hellman

- An observer cannot work out $r_A$ and $r_B$ from $t_A$ and $t_B$ therefore the attacker cannot calculate the key.

- So we have a “Good Key” but know nothing about the participants.

- We did not need to share any keys at the start, therefore this is a very powerful protocol.

- In practice: use DH to set up a secure channel, then use something else to authenticate the person at the other end.
Elgamal

• Elgamal, is Diffie-Hellmen turned into a public key scheme. It uses a fix g & p

• “Alice” picks \( r_A \) as her private key
  & “\( t_A = g^{r_A} \mod p \)” is the public key.

• To encrypt message “M”, Bob picks \( r_A \) finds and sends \((g^{r_B} \mod p, M, (t_A)^{r_B} \mod p)\)
Exercise VM

• The exercises for this module will be based on a Linux VM.
  – Download it from the exercise webpage
  – Run it using VirtualBox.

• To get you started:
  – username:alice password:alice
  – Username:bob password:bob
Tutorial Groups

• Weekly small group meetings.

• Your tutor will contact you to arrange a time and place to meet
  – If you don’t get an e-mail by Wednesday then let me know.

• Don’t understand something? E-mail your tutor, he will discuss it in tutorials.
WorkSheet 1

• Worksheets 1: Get the VM working & do some simple crypto
  – You get a few token marks.

• This week only: e-mail you solutions to your tutor before your meeting.
  – Your e-mail must be signed and encrypted.
Lab Session

• Lab session Wednesday 10-12 in UG04
  – Bring your laptop, or use a school machine.

• Myself and/or tutors will be present to give you individual help with your programs, tools, exercises.

• Come along on Wednesday, with your laptop, and we will help get a VM installed and PGP set up.