Catch-up
&

Tasks on DH, RSA and Padding

Lec 6a

Overview

- By the end of the session, you should :
  - know what we have done in the module so far? [like a summary of key points]
  - Be able to solve DH, RSA and padding related problems.

What we have done so far?

Week 1

a) CIA & Authentication and Accountability

b) Types of threats: Active/Passive, Insider/Outsider

c) Attack surface and attack trees

d) Other key terms like vulnerability etc

Week 2

a) General intro to malware

b) Basic operation of viruses, worms and trojans

c) Payload types

d) bots, spyware and rootkits

e) Countermeasures

What we have done so far?

- Weeks 3-6 Cryptography
  a) General introduction
  b) Cryptographic system categories
    - Substitution/Transposition
      Single letter, multi-letter substitutions
      Rail-fence and route transposition techniques
    - Block/Stream
      Padding techniques: Random,Zelolen, Null etc
      Salt: playback issue, CBC, CFB, OFB, CTR
      GCM
    - Symmetric/Asymmetric
      Explained role of keys
      Symmetric: DES, 3DES and AES
      Asymmetric: RSA, DH, Elliptic Curve
      Digital signature, certificates
  c) Cryptographic hashing (Friday's lecture)

What we have covered in labs?

1) CIA concepts based on scenario
2) Behaviours of malware
3) PHP application
4) Used different ciphers to secure an asset
5) Cryptographic standard OpenSSL library

Some Mac users have issue installing OpenSSL – only two students reported it to me. It is solved, check discussion pages on Canvas.

Module learning aims

- Systematically discuss key dimensions of computer security (e.g. secrecy, authentication, integrity, anonymity), and their relationship to the main threats and attack techniques relevant to computer security.

- Systematically describe the main building blocks of cryptography (e.g. public and private key encryption, cryptographic hashing), and their relationship with the key dimensions of computer security from LO1.

- Deploy up-to-date tools and techniques for finding vulnerabilities in computer systems. Draft security policies and implement policy enforcement processes and mechanisms.

- Design secure computer systems by using established computer security principles.

Quiz

How are you attempting lab work? Are you using your own PC?

If NOT, I need to know by end of this week to avoid any problem in setting up SQLi and XSS labs.

https://canvas.sussex.ac.uk/courses/13026/quizzes/17639

Catch-up

- You have interim report for FYP but make sure that you progress in computer security module

- Use this week to catch up with this module both in theory and labs work

Next week – Web security [SQLi and XSS Attacks], HTTP

Revise it for Exam

Study examples of DH, RSA and Padding then attempt tasks provided on each of the three topics

Task - DH

Examples of DH are on the next two slides.

Problem-1

Suppose that two parties A and B wish to set up a common secret key (D-H key) between themselves using the Diffie-Hellman key exchange technique. They agree on 7 as the modulus and 3 as the primitive root. Party A chooses 2 and party B chooses 5 as their respective secrets. Find the DH key.

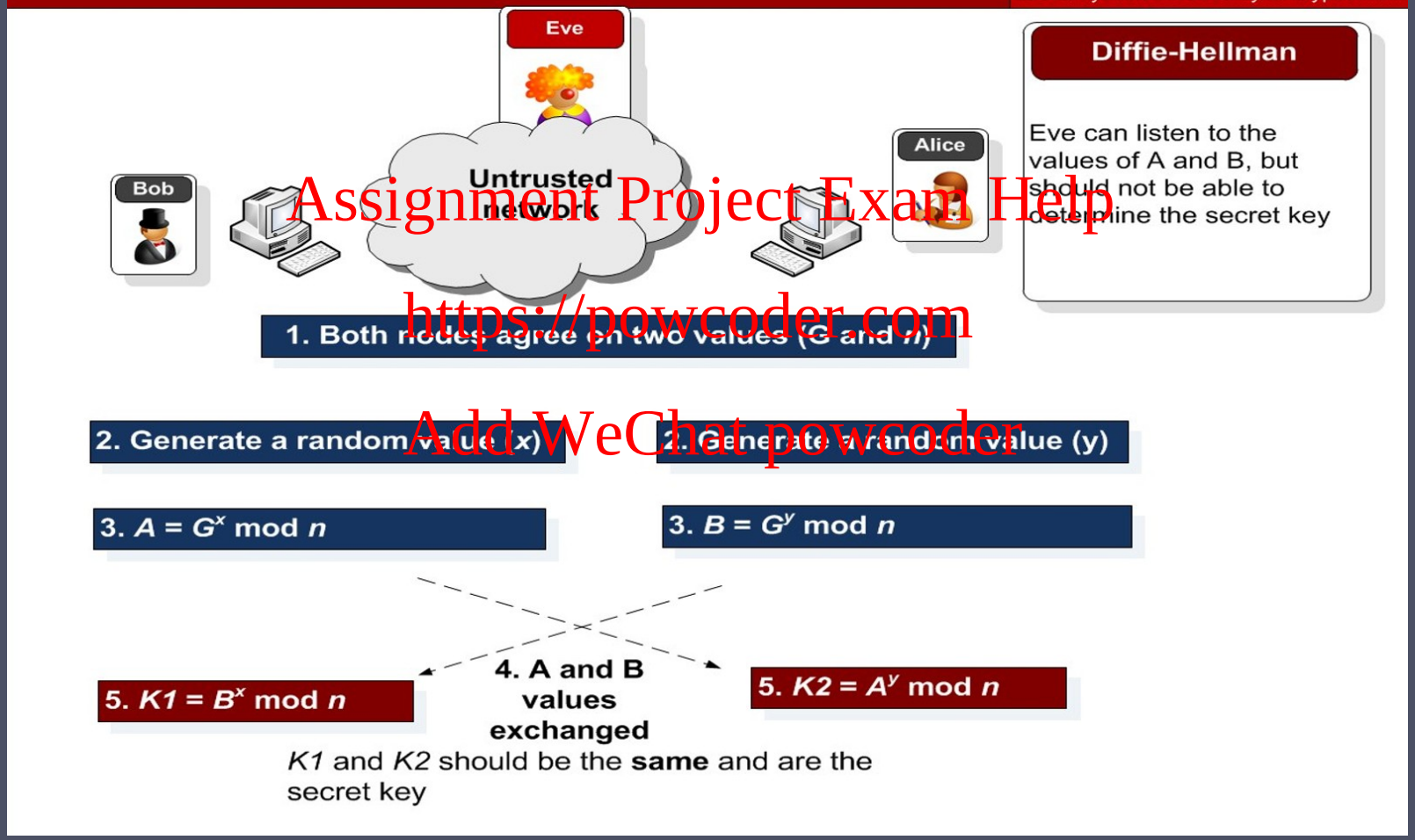
Problem-2

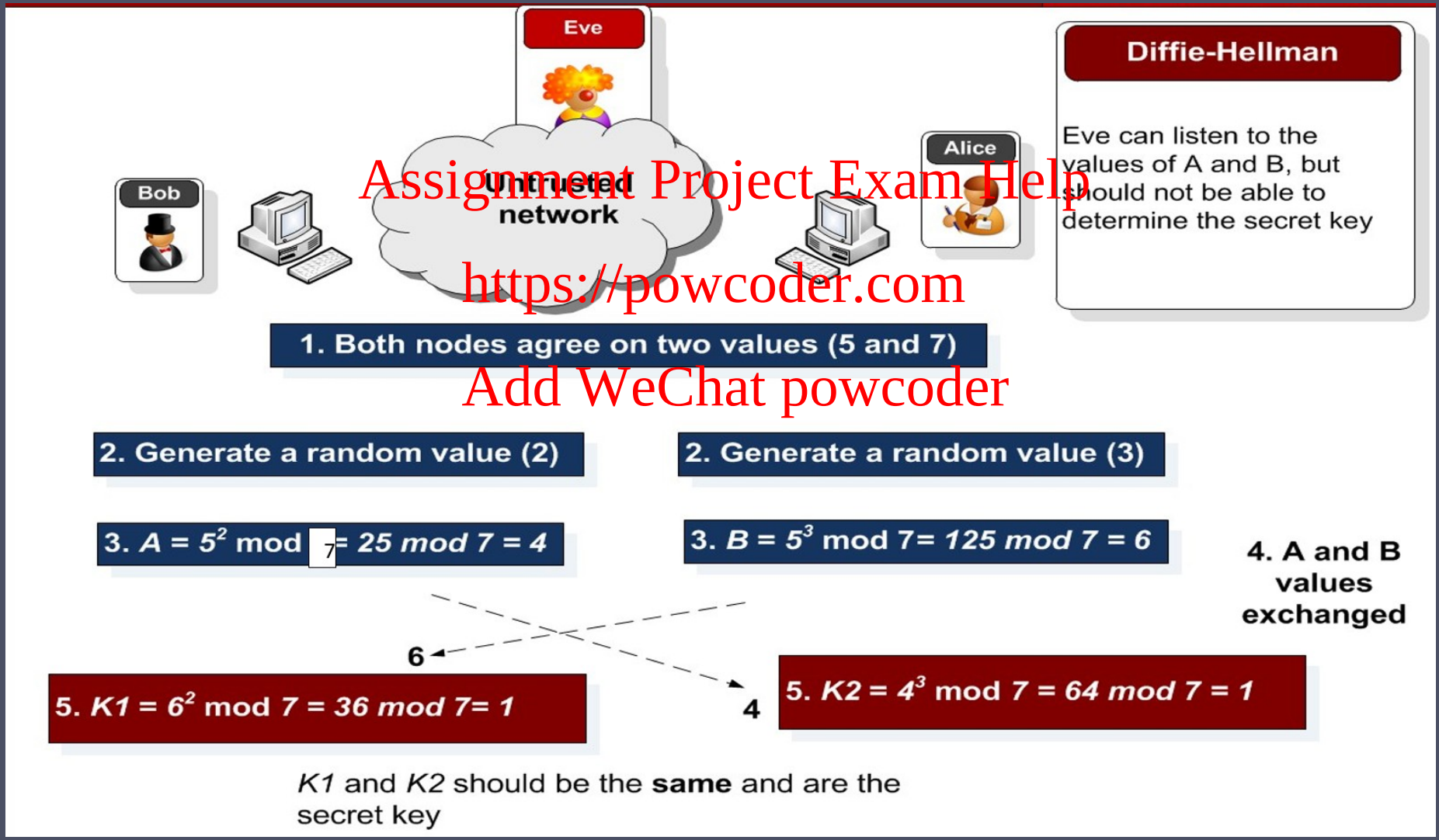
In a Diffie-Hellman Key Exchange, Alice and Bob have chosen prime value q = 17 and primitive root = 5. If Alice's secret key is 4 and Bob's secret key is 6, what is the secret key they exchanged?

Once you attempt, check your solution against:

https://www.gatevidyalay.com/tag/diffie-hellman-key-exchange-tutorial/

Example of DH protocol





Task - RSA

- In an RSA cryptosystem, a particular A uses two prime numbers p = 13 and q =17 to generate her public and private keys. If the public key of A is 35. Then the private key of A is?

RSA – example01

Encryption	Decryption
Public key: (5,14)	Private key (11, 14)
Plaintext: B → 2 index	Note: 14 is the same
C= mod N	Ciphertext: D→ 4
(mod) 14 = 32 (mod 14) = 4 (mod) 14 = D = 4 index	M= mod N (mod)14 = 4194304 (mod 14) = 2 (mod 14) = B = 2 index

How does it work?

1st step: two primes number p and q

p=2 and q=7

2nd step: product of p and q = p x q = 14 = N

which is mod in public and private key, it is publicise

3rd step: (pronounced as PHI(N) = (p-1)(q-1)

=(2-1)(7-1)

= 6 = total number of co-prime

Coprime
1=1x1 3=3x1 5=5x1 9=3x3 11=11x1 13=13x1

How many coprime below 14?	
14=2x7	2=2x1 4=2x2 6=3x2 8=2x2x2 12=2x2x3
14=2x7	1=1x1 3=3x1 5=5x1 7=7x1 9=3x3 11=11x1 13=13x1

4th step: Choose e 1< e < (N) = 2,3,4,5

{ co-prime with N, (N) = 2,3,4,5

N=14, (N)=6;

public key = 5, 14

5th step: choose d: de (mod (N)) = 1

5d (mod 6) = 1

d should be such a number that when it multiplies with 5 and find mod by 6, it should give you 1

d	1	2	3	4	5	…..
5d	5	10	15	20	25	……
mod 6	5	4	3	2	1	0

This pattern repeat, pick any number that give you mod 1

1 2 3 4 5 6 7 8 9 10 11 12 13 14

RSA - example02

Encryption	Decryption
two primes p x q ; p=3, p=11 N = p x q = 3 x 11 = 33 (N) = (p-1)(q-1) = (3-1) (11-1) = 2 x 10 = 20 [this will be our mod] = Both parties will have this value	
Selecting e 1< e < (N) = 1<e<20 { co-prime with N, (N) e=3 public key = [3, 33]	(d x e) mod (N) = 1 (d x 3) mod 20 = 1

d	e	PHI	= 1
	e [must not have a common factor with 20]	Mod 20	
1	3	Mod 20	≠
2	3	Mod 20	-
3	3	Mod 20	-
4	3	Mod 20	-
5	3	Mod 20	-
6	3	Mod 20	-
7	3	Mod 20	1

d = 7

Task - Padding

Show the working of different padding techniques for the plaintext hell when the cipher is AES

You will need to ASCI table

Padding examples

Plaintext: hello where h=68, e=65 and so on …

68=h,e=65 [0b in hexadecimal = 11]

After padding (CMS): 68656c6c6f0b0b0b0b0b0b0b0b0b0b0b

Cipher (ECB): 0a7ec77951291795bac6690c9e7f4c0d

Message hex [80=128 by Bruce] zeros bytes

After padding (Bit): 68656c6c6f8000000000000000000000

Cipher (ECB): 731abffc2e3b2c2b5caa9ca2339344f9

ASCII Check values here http://www.asciitable.com/

Afterpadding(ZeroLen):

[Number of padding bytes ten, excluding 0a (hex=10)]

68656c6c6f000000000000000000000a

Cipher (ECB): d28e2f7e8e44e068732b292bde444245

After padding(Null): 68656c6c6f0000000000000000000000

Cipher (ECB): 4447974224604 53d95856eb2a1520ece

After padding (Space): 68656c6c6f0000000000000000000000

Cipher (ECB): 4447974224604 53d95856eb2a1520ece

Error this is actually 20……

[Number of random bytes]

After padding (Random): 68656c6c6*ffc6ecfd884a38798d62a***0a**

Cipher (ECB): c2c88b4364d2c2dc6f2cac9ab73c995d

Another example of padding:

Plaintext: hello123

For CMS with AES,

AES use 16 bytes

The plaintext will use 8 bytes (count letters in plaintext)

Padding bytes = 16 – 8 (plaintext bytes) = 8 bytes