

# Chapter 5: Key Exchange

Diffie-Hellman

Diffie-Hellman Weaknesses

Passing Key Using Public Key

**Prof Bill Buchanan OBE**

<http://asecuritysite.com/crypto05>

<http://asecuritysite.com/encryption>



Eve



## Private key

Private key uses the same key for encryption and decryption ... how does Bob send the key to Alice?

How do Bob and Alice send their private (secret) key without Eve getting it?

Hello

Encryption

Communications Channel

Decryption

Hello

H&\$d.

H&\$d.

Bob



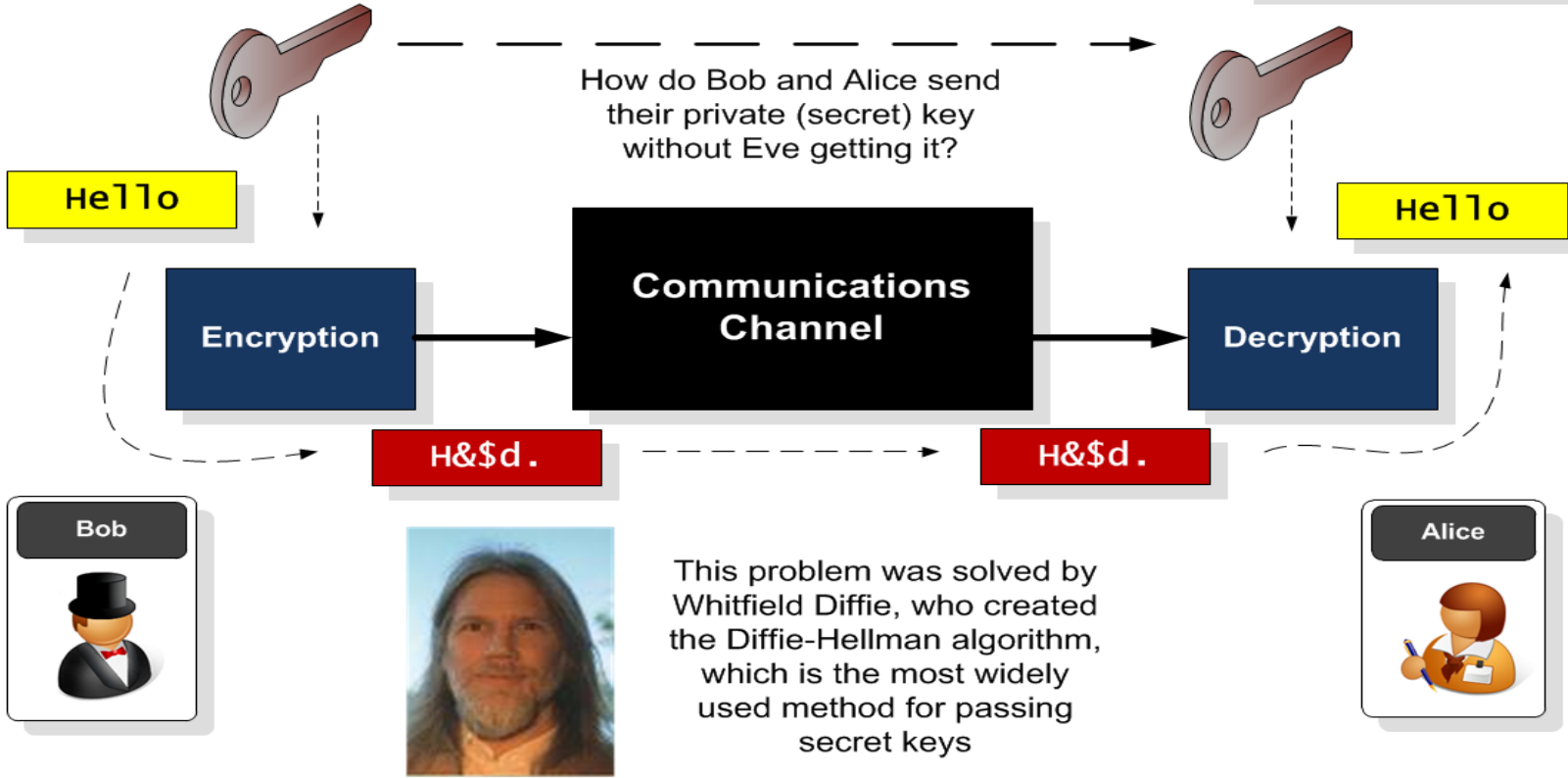
Alice





Diffie-Hellman

One of the most widely method for creating a secret key which is the same for Bob and Alice



## Key Exchange

- **Forward secrecy (FS)**, which means that a compromise of the long-term keys will not compromise any previous session keys. A leakage of the public key of the server would cause all the sessions which used this specific public key to be compromised. FS thus aims to overcome this by making sure that all the session keys could not be compromised, even though the long-term key was compromised.
- **Ephemeral**. With some key exchange methods the same key will be generated if the same parameters are used on either side. This can cause problems as an intruder could guess the key, or even where the key was static and never changed. With ephemeral methods, a different key is used for each connection, and, again, the leakage of any long-term would not cause all the associated session keys to be breached.



$$A^x A^y$$



$$A^{(x+y)}$$

$$(A^x)^y$$



$$A^{xy}$$





Random value

**X**

**A**

Agreed number

Random value

**y**



**A<sup>x</sup>**

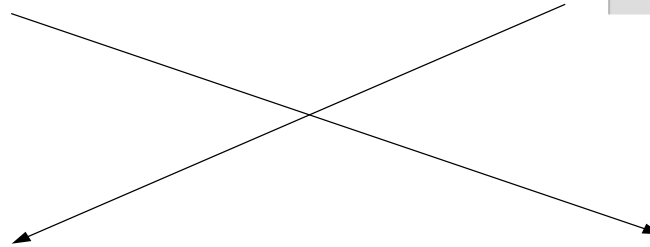
**A<sup>y</sup>**



**A<sup>y</sup>**

**A<sup>x</sup>**

Private key



Logs

Encryption



Random value

**X**

**A**

Agreed number

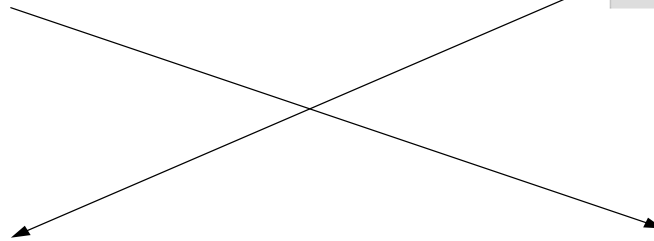
Random value

**y**



$A^x$

$A^y$



$A^y$

$A^x$

$(A^y)^x$

$(A^x)^y$

Logs

Encryption



## Diffie-Hellman

Eve can listen to the values of A and B, but should not be able to determine the secret key

1. Both nodes agree on two values (G and  $n$ )

2. Generate a random value ( $x$ )

2. Generate a random value ( $y$ )

3.  $A = G^x \bmod n$

3.  $B = G^y \bmod n$

5.  $K1 = B^x \bmod n$

4. A and B  
values  
exchanged

5.  $K2 = A^y \bmod n$

$K1$  and  $K2$  should be the **same** and are the secret key





## Diffie-Hellman

Eve can listen to the values of A and B, but should not be able to determine the secret key

1. Both nodes agree on two values (5 and 7)

2. Generate a random value (2)

2. Generate a random value (3)

$$3. A = 5^2 \bmod 8 = 25 \bmod 7 = 4$$

$$3. B = 5^3 \bmod 7 = 125 \bmod 7 = 6$$

4. A and B values exchanged

$$5. K1 = 6^2 \bmod 7 = 36 \bmod 7 = 1$$

$$5. K2 = 4^3 \bmod 7 = 64 \bmod 7 = 1$$

K1 and K2 should be the **same** and are the secret key

Example



# Diffie-Hellman Generator

$$Y = G^x \bmod p$$

p	11							
Generator	2	3	4	5	6	7	8	9
x	$g^x \bmod p$	$g^x \bmod p$	$g^x \bmod p$	$g^x \bmod p$	$g^x \bmod p$	$g^x \bmod p$	$g^x \bmod p$	$g^x \bmod p$
2	4	9	5	3	3	5	9	4
3	8	5	9	4	7	2	6	3
4	5	4	3	9	9	3	4	5
5	10	1	1	1	10	10	10	1
6	9	3	4	5	5	4	3	9
7	7	9	5	3	8	6	2	4
8	3	5	9	4	4	9	5	3
9	6	4	3	9	2	8	7	5
10	1	1	1	1	1	1	1	1

Picking G

## Diffie-Hellman Generation

```
C:\> openssl dhparam -out dhparams.pem 768 -text
```

```
C:\> type dhparams.pem
```

```
Diffie-Hellman-Parameters: (768 bit)
```

```
prime:
```

```
00:d0:37:c2:95:64:02:ea:12:2b:51:50:a2:84:6c:
```

```
71:6a:3e:2c:a9:80:e2:65:b2:a5:ee:77:26:22:31:
```

```
66:9e:fc:c8:09:94:e8:9d:f4:cd:bf:d2:37:b2:fb:
```

```
b8:38:2c:87:28:38:dc:95:24:73:06:d3:d9:1f:af:
```

```
78:01:10:6a:7e:56:4e:7b:ee:b4:8d:6b:4d:b5:9b:
```

```
93:c6:f1:74:60:01:0d:96:7e:85:ca:b8:1f:f7:bc:
```

```
43:b7:40:4d:4e:87:e3
```

```
generator: 2 (0x2)
```

```
-----BEGIN DH PARAMETERS-----
```

```
MGYCYQDQN8KVZALqEitRUKKEbHFqPiypgOJlsqXudyYiMWae/MgJlOid9
```

```
M2/0jey
```

```
+7g4LIcoONyVJHMG09kfr3gBEGp+Vk577rSNa021m5PG8XRgAQ2WfoXKu
```

```
B/3vEO3
```

```
QE1Oh+MCAQI=
```

```
-----END DH PARAMETERS-----
```

- **DH Group 5:** 1,536 bit prime.
- **DH Group 2:** 1,024 bit prime.
- **DH Group 1:** 768-bit prime.

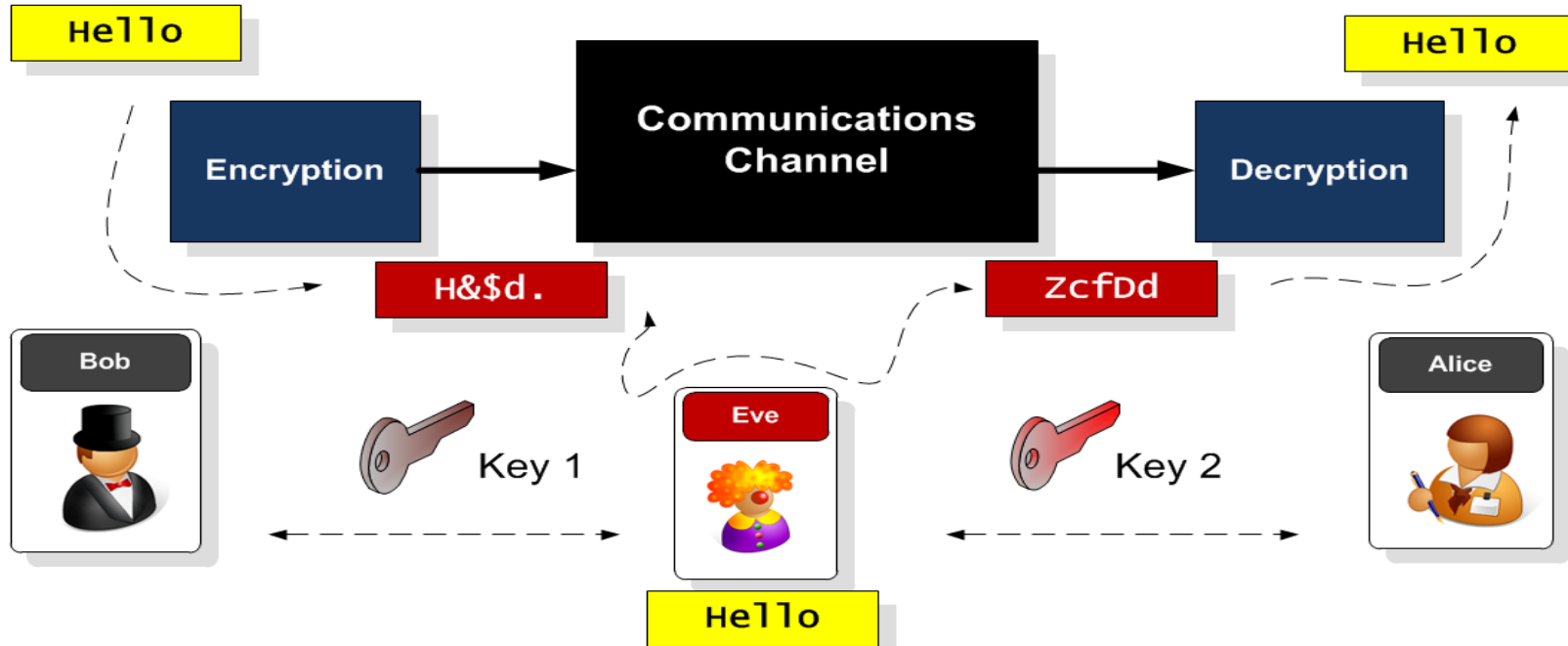
## Diffie-Hellman Weaknesses

- In 2015, a paper entitled *Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice* – showed that it was fairly easy to precompute on values for two popular Diffie-Hellman parameters (and which use the DHE\_EXPORT cipher set).
- The research team found that one was used as a default in the around 7% of the Top 1 million web sites and was hard coded into the Apache httpd service. Overall, at the time, it was found that over 3% of Web sites were still using the default.
- Diffie-Hellman-Parameters: (512 bit)
- prime:
  - 00:9f:db:8b:8a:00:45:44:f0:04:5f:17:37:d0:ba:
  - 2e:0b:27:4c:df:1a:9f:58:82:18:fb:43:53:16:a1:
  - 6e:37:41:71:fd:19:d8:d8:f3:7c:39:bf:86:3f:d6:
  - 0e:3e:30:06:80:a3:03:0c:6e:4c:37:57:d0:8f:70:
  - e6:aa:87:10:33
- generator: 2 (0x2)

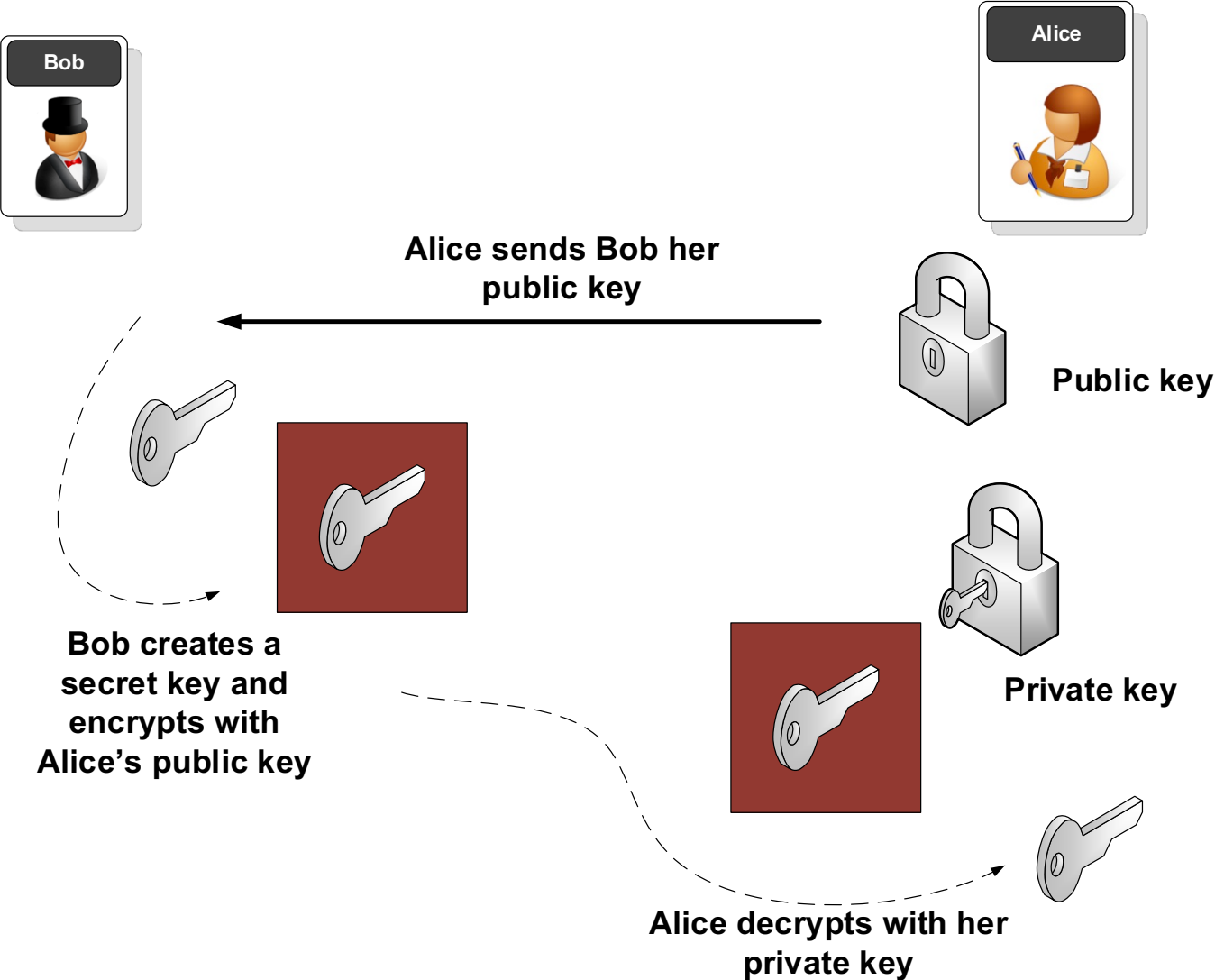
Diffie-Hellman suffers from a man-in-the-middle attack, where Eve negotiates for each side, and creates two encryption channels

### Man-in-the-middle

Diffie-Hellman suffers from Eve intercepting the key interchange, so that Bob thinks he's talking to Alice for the key exchange.



# Key Exchange with Public Key



# Chapter 5: Key Exchange

Diffie-Hellman

Diffie-Hellman Weaknesses

Passing Key Using Public Key

**Prof Bill Buchanan OBE**

<http://asecuritysite.com/crypto05>

<http://asecuritysite.com/encryption>

