Sy

Basic:
Block
Secre
Saltin
AES
Key E

Pro

https

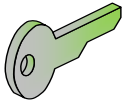
No	Date	Subject	Lab
2	15 Sept 2022	1. Introduction [Link] 2. Intrusion Detection Systems [Link]	Introduction to Vyatta Lab
3	22 Sept 2022	3. Network Security [Link]	Vyatta and Snort. [Link]
4	29 Sept 2022	4. Ciphers and Fundamentals [Link]	pfSense.
5	6 Oct 2022	5. Secret Key 6. Hashing [Link]	AWS Security and Server Infrastructures
6	13 Oct 2022	7. Public Key [Link] 8. Key Exchange [Link]	Public/Private Key and Hashing
7	20 Oct 2022	9. Digital Certificates	Certificates here
8	27 Oct 2022	10 Network Forensics here	Network Forensics lab
9	3 Nov 2022	Test 1 here	
10	10 Nov 2022	11. Splunk here	Splunk Lab
11	17 Nov 2022	12. Splunk and Machine Learning	Splunk and ML Lab
12	24 Nov 2022	13. Tunnelling here	Tunnelling
13	1 Dec 2022	14. Blockchain and Cryptocurrencies here	Blockchain Lab.
14	8 Dec 2022		
15	15 Dec 2022	Hand-in: TBC [Here]	

Typical application:
Diffie-Hellman used to generate private-key.
Public-key used for authentication.
Private-key used for encryption.

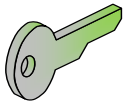
Encryption/ Decryption

Communications Channel

Encryption/ Decryption

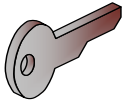


Key exchange (Diffie-Hellman)

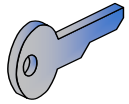


Secret key used to encrypt/decrypt
(DES/3DES/AES)

Public key



Used to authenticate (RSA)



Private key

Private key

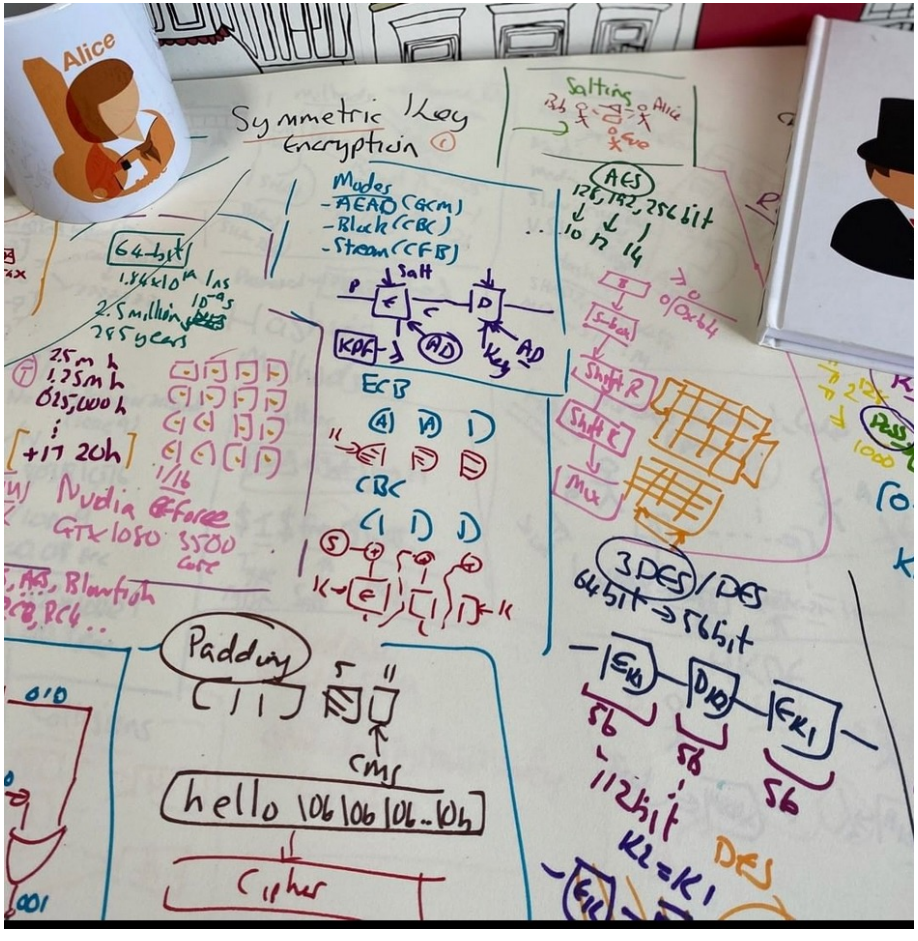
RSA 2048 bits
Replace by:
ElGamal 160bits

Public key

Author: Prof Bill Buchanan

Secret Key

Basics
Block or Stream?
Secret Key Methods
Salting
AES
Key Entropy



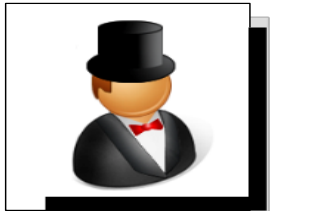
Prof Bill Buchanan OBE

https://asecuritysite.com/symmetric

Intruder

Eve

Privacy (Private Key)
Identity (Public Key)
Integrity (Public/Private Key)



Bob

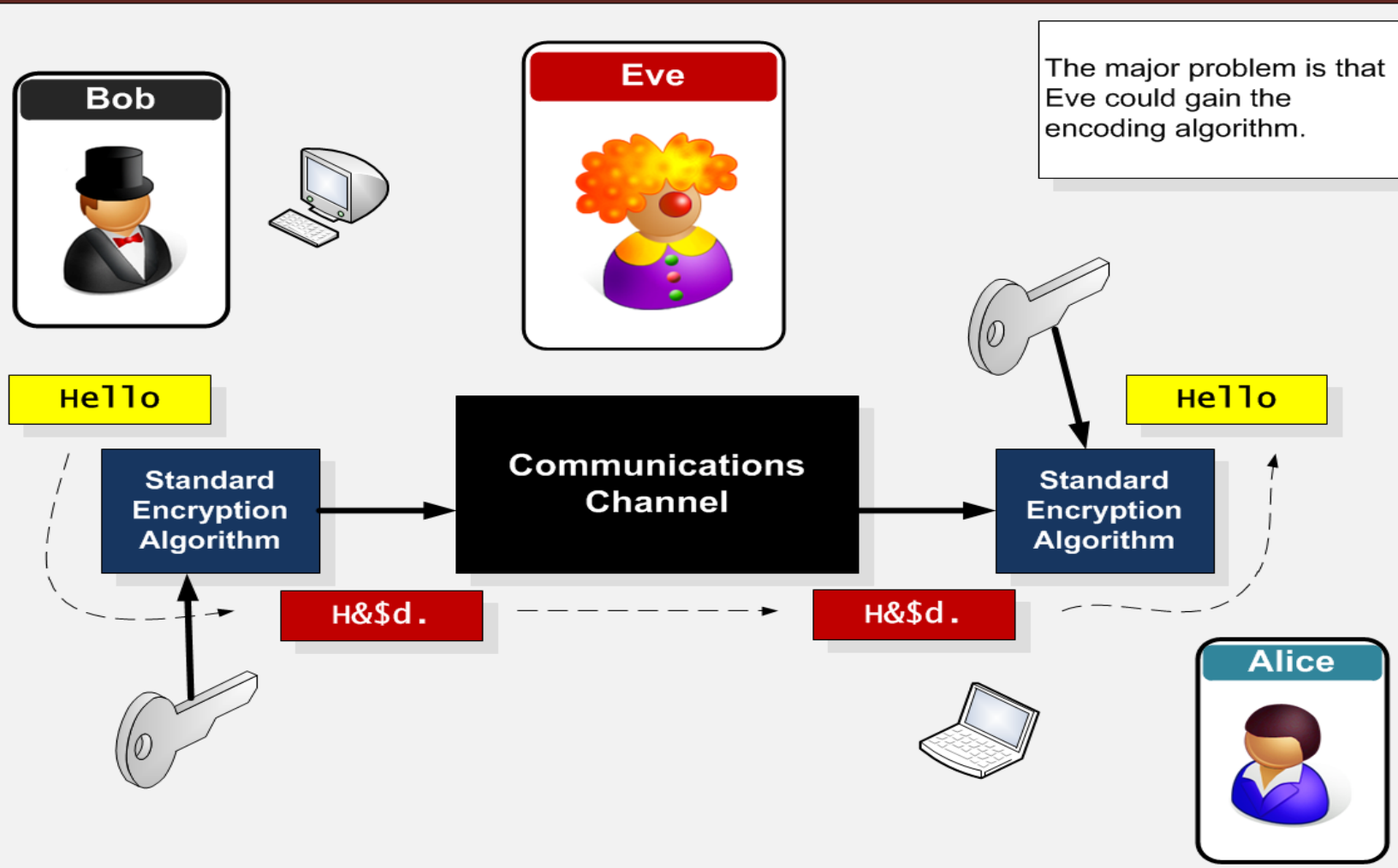
Alice

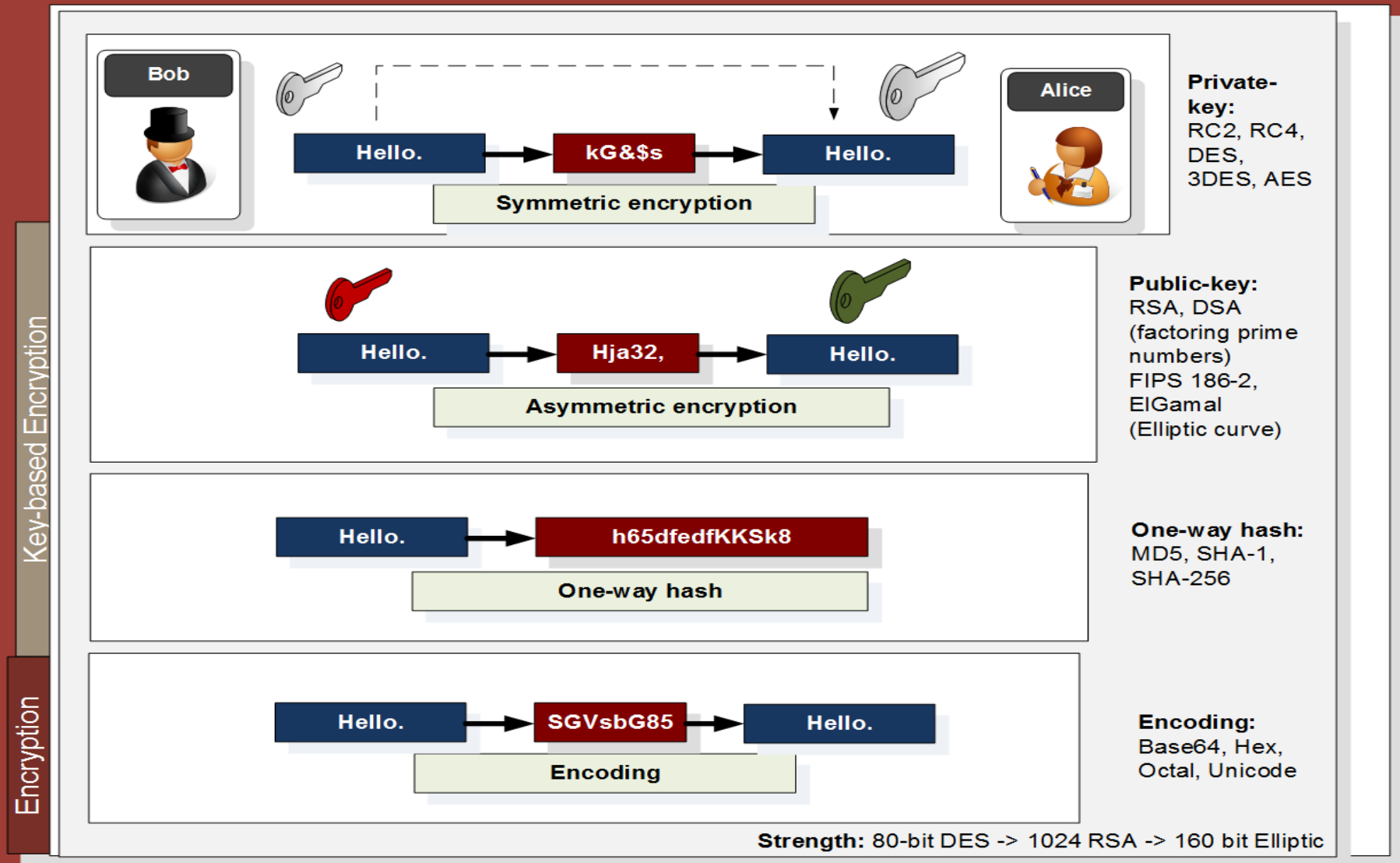
John

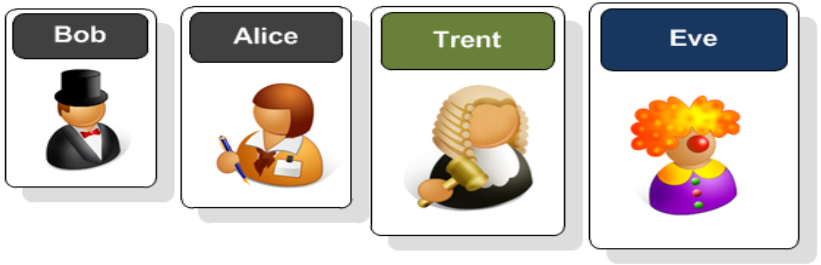
Trusted third party

Trent

Author: Prof Bill Buchanan


Bob
Eve
The major problem is that Eve could gain the encoding algorithm.
Hello
Standard Encryption Algorithm
Communications Channel
Standard Encryption Algorithm
Hello
H&$d.
H&$d.
Alice


Encryption
Key-based Encryption
Bob
Alice
Hello.
kG&$s
Hello.
Symmetric encryption
Private-key:
RC2, RC4, DES, 3DES, AES
Hello.
Hja32,
Hello.
Asymmetric encryption
Public-key:
RSA, DSA (factoring prime numbers)
FIPS 186-2, ElGamal (Elliptic curve)
Hello.
h65dfedfKKSk8
One-way hash
One-way hash:
MD5, SHA-1, SHA-256
Hello.
SGVsbG85
Hello.
Encoding
Encoding:
Base64, Hex, Octal, Unicode
Strength: 80-bit DES -> 1024 RSA -> 160 bit Elliptic



Number of keys

The larger the key, the greater the key space.

security level	volume of water to bring to a boil	bit-lengths: symmetric key	bit-lengths: cryptographic hash	bit-lengths: RSA modulus
teaspoon security	0.0025 liter	35	70	242
shower security	80 liter	50	100	453
pool security	2 500 000 liter	65	130	745
rain security	0.082 km^3	80	160	1130
lake security	89 km^3	90	180	1440
sea security	3 750 000 km^3	105	210	1990
global security	1 400 000 000 km^3	114	228	2380
solar security	-	140	280	3730

9	512	44	1.76×10^{13}	84	1.93×10^{25}
10	1024	48	2.81×10^{14}	88	3.09×10^{26}

Time to crack

- It is important to understand the length of time that a message takes to crack as it may need to be secret for a certain time period.

Okay… we select a **64-bit key** …
which has 1.84×10^{19} combinations

18.4 million million million different keys
000000000000….0000000000000000000
To
111111111111….1111111111111111111

How long will it take to crack it by brute-force (on average)?

Author: Prof Bill Buchanan

Time to crack

- It is important to understand the length of time that a message takes to crack as it may need to be secret for a certain time period.

A 64-bit key has 1.84×10^{19} combinations and it could be cracked by brute-force in 0.9×10^{19} goes.

If we use a fast computer such as 1GHz clock (1ns), and say it takes one clock cycle to test a code, the time to crack the code will be:

9,000,000,000 seconds (150 million minutes) … **2.5 million hours** (285 years)

Author: Prof Bill Buchanan

Time to crack

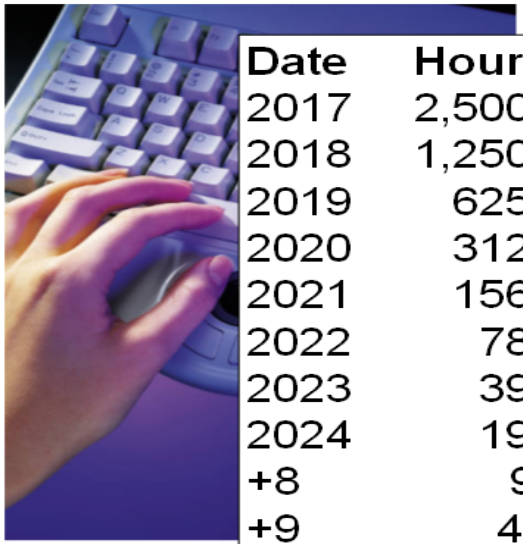
- It is important to understand the length of time that a message takes to crack as it may need to be secret for a certain time period.

If it takes 2.5 million hours (285 years) to crack a code. How many years will it take to crack it within a day?

Computers typically improve their performance every year … so assume a **doubling** of performance each year.

Date	Hours	Days	Years
2017	2,500,000	104,167	285
2018	1,250,000	52,083	143

Author: Prof Bill Buchanan



Date	Hours	Days	Years
2017	2,500,000	104,167	285
2018	1,250,000	52,083	143
2019	625,000	26,042	71
2020	312,500	13,021	36
2021	156,250	6,510	18
2022	78,125	3,255	9
2023	39,063	1,628	4
2024	19,532	814	2
+8	9,766	407	1
+9	4,883	203	1
+10	2,442	102	0.3
+11	1,221	51	0.1
+12	611	25	0.1
+13	306	13	0
+14	153	6	0
+15	77	3	0
+16	39	2	0
+17	**20**	**1**	**0**

Time to crack

- From 285 years to 1 day, just by computers increasing their computing power.

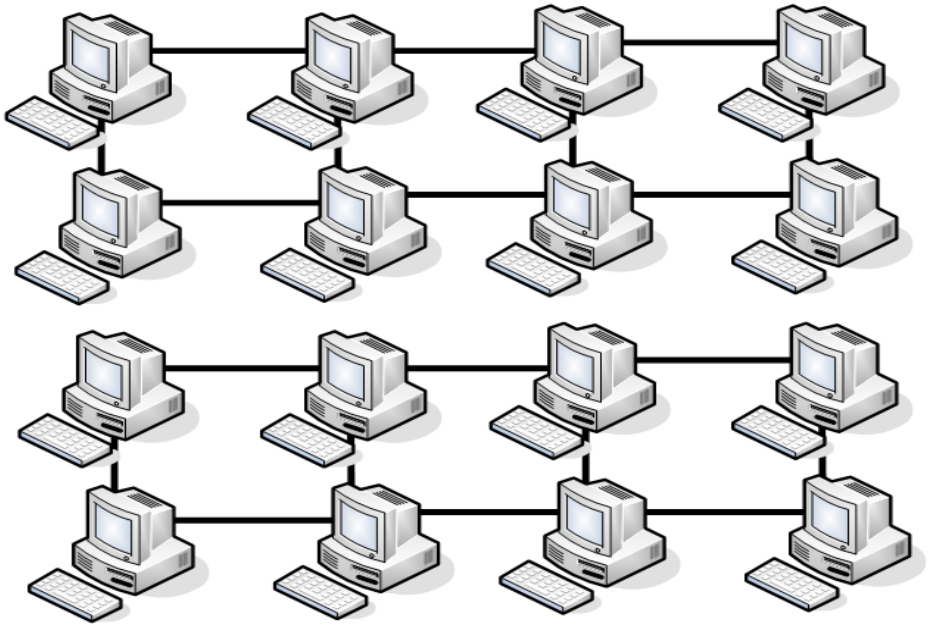
56-bit DES:
Developed
1975
30 years ago!
… now easily
crackable

Author: Prof Bill Buchanan

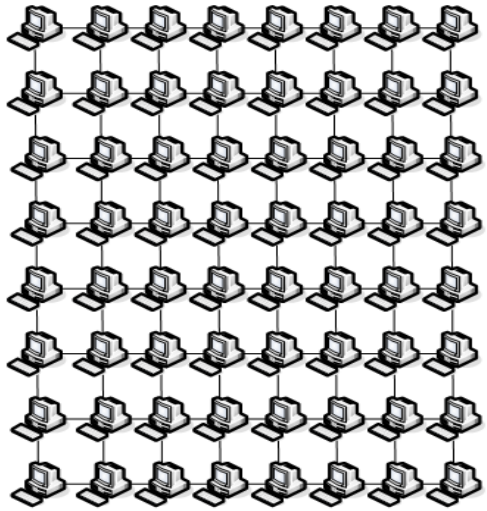
Parallel processing

- 2x2 array = 4 computers.
- 4x4 array = 16 computers.
- 8x8 array = 64 computers.

2x1 =2 element array



4x4 = 16 element array

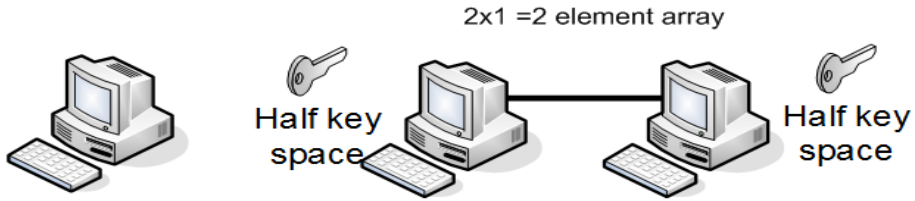


16x16 = 256 element array

Author: Prof Bill Buchanan

Parallel processing

- 64-bit key --- from **104,000 days** (284 years) to one hour or less.



Processors	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
1	104000 days	52000	26000	13000	6500	3250
4	26000	13000	6500	3250	1625	813
16	6500	3250	1625	813	407	204
64	1625	813	407	204	102	51
256	406	203	102	51	26	13
1024	102	51	26	13	7	4
4096	25	13	7	4	2	1
16,384	152hr	76hr	38hr	19hr	10hr	5hr
65,536	38hr	19hr	10hr	5hr	3hr	2hr
262,144	10hr	5hr	3hr	2hr	1hr	
1,048,576	2hr	1hr				

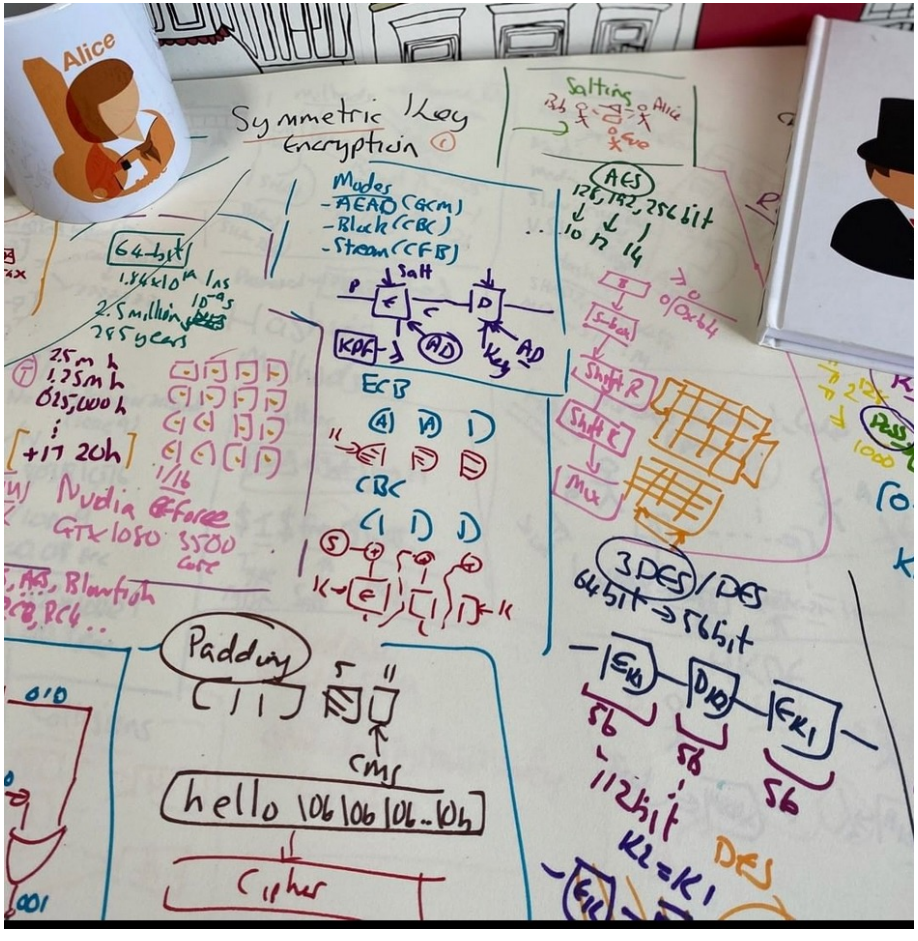
16x16 = 256 element array

4x4 = 16 element array

Author: Prof Bill Buchanan

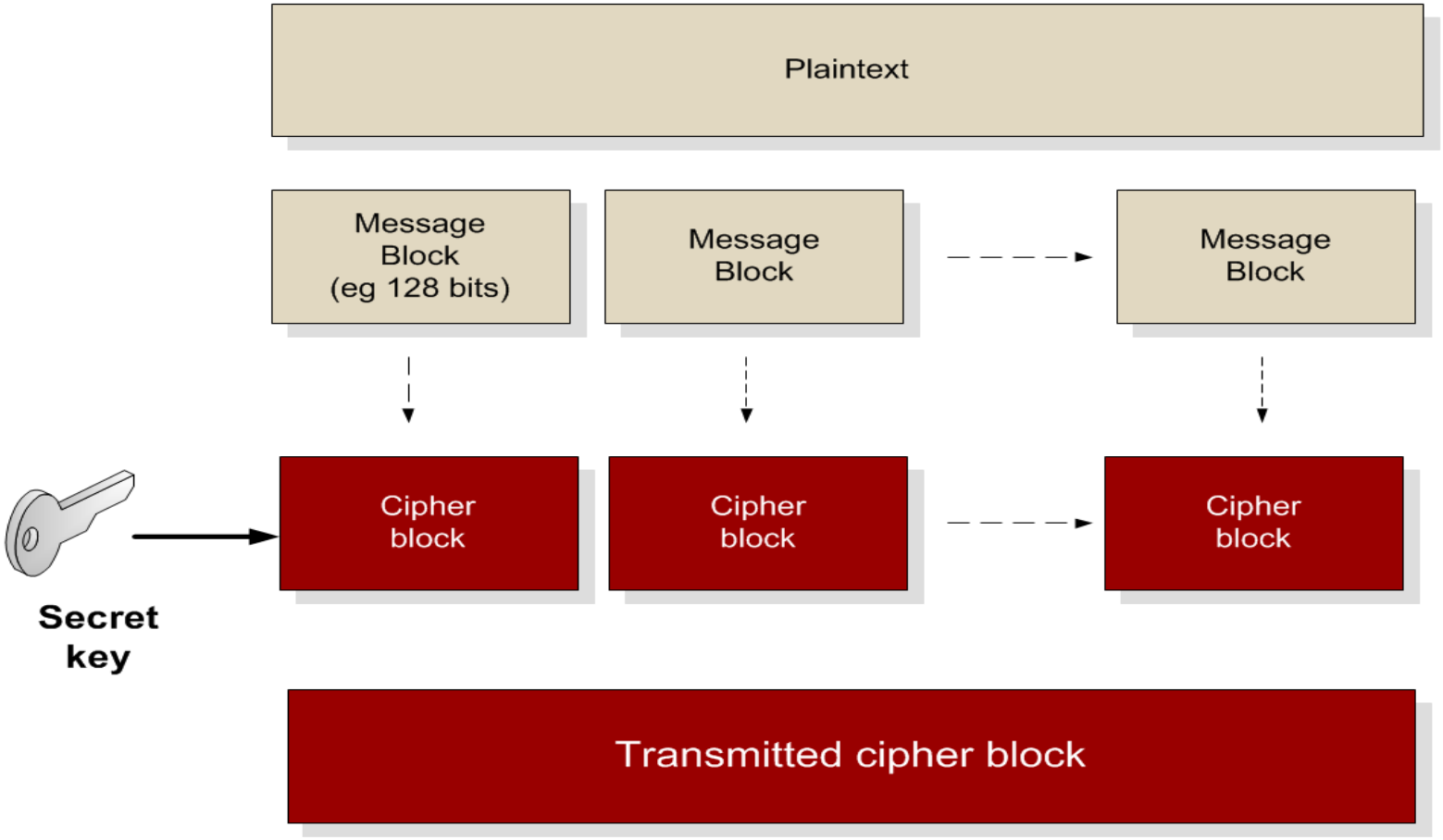
Symmetric Key

Basics
Block or Stream?
Secret Key Methods
Salting
AES
Key Entropy



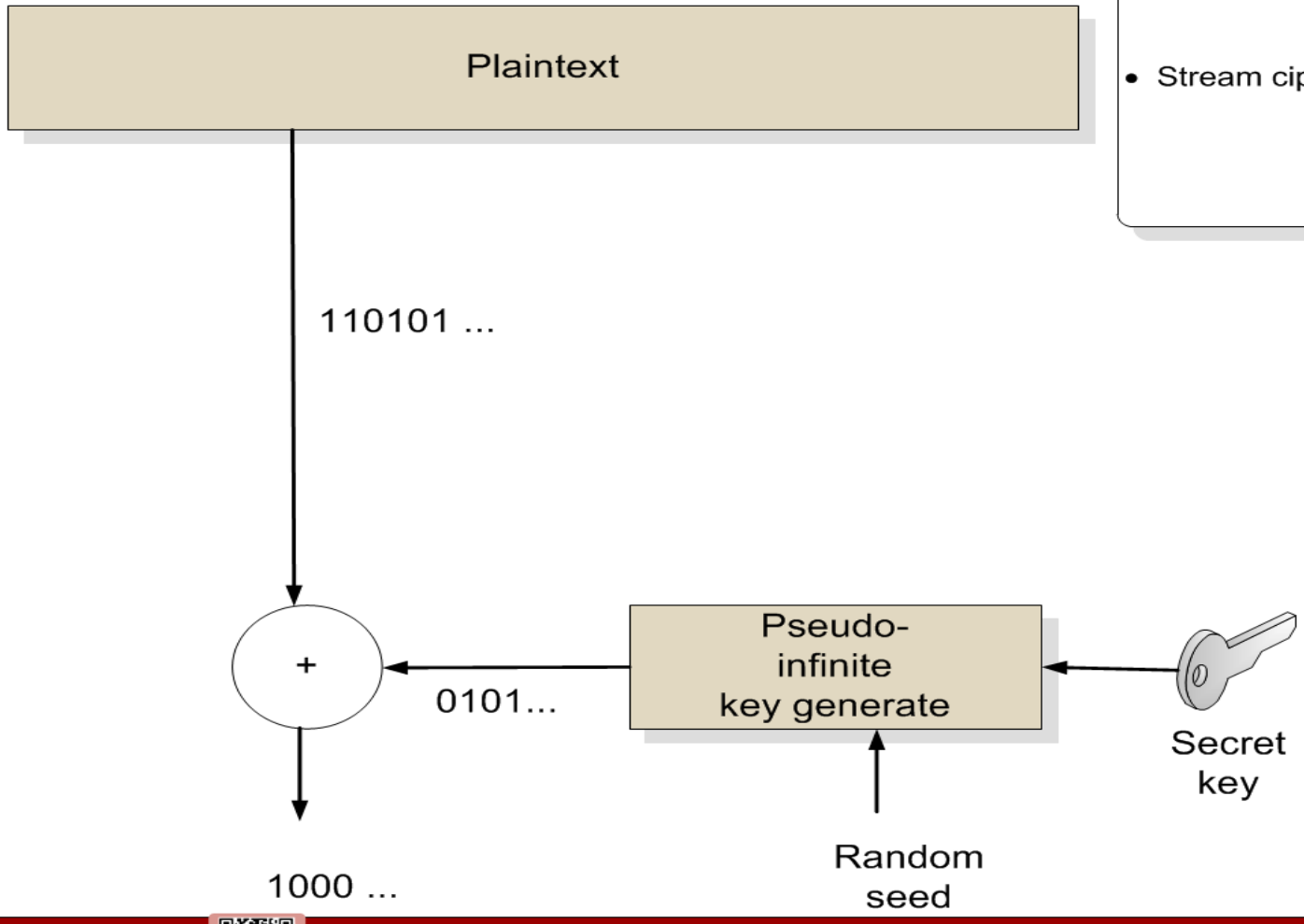
Prof Bill Buchanan OBE

https://asecuritysite.com/symmetric

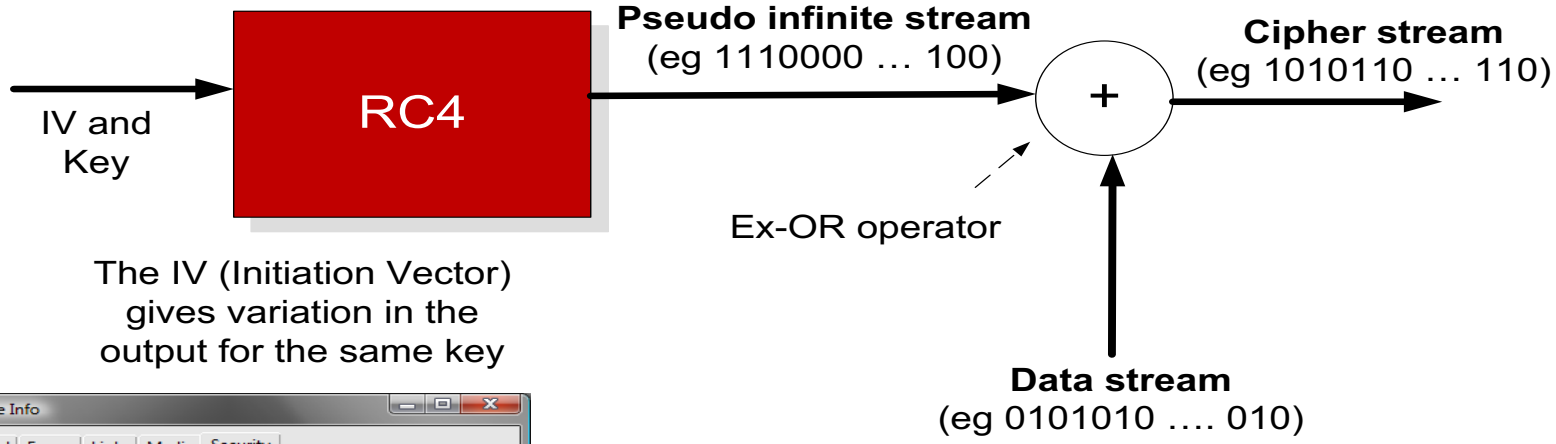

Plaintext
Message Block (eg 128 bits)
Message Block
Message Block
Cipher block
Cipher block
Cipher block
Secret key
Transmitted cipher block

Stream cipher

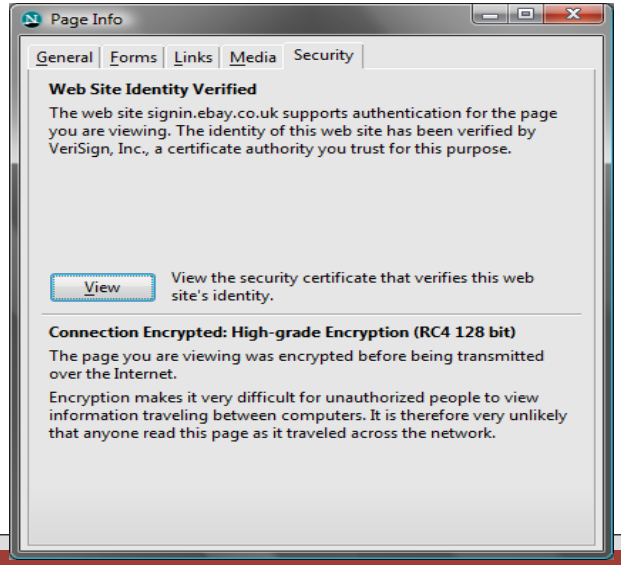
- Stream cipher (RC4)



RC4. This is a **stream** encryption algorithm, and is used in wireless communications (such as in WEP) and SSL (Secure Sockets).



The IV (Initiation Vector) gives variation in the output for the same key



Data stream	0101010 … 010
Pseudo infinite stream	1110000 … 100 ⊕
Cipher stream	1010110 … 110

Author: Prof Bill Buchanan

Padding

- **CMS** (Cryptographic Message Syntax). This pads with the same value as the number of padding bytes. Defined in RFC 5652, PKCS#5, PKCS#7 and RFC 1423 PEM.
- **Bits**. This pads with 0x80 (10000000) followed by zero (null) bytes. Defined in ANSI X.923 and ISO/IEC 9797-1.
- **ZeroLength**. This pads with zeros except for the last byte which is equal to the number (length) of padding bytes.
- **Null**. This pads will NULL bytes. This is only used with ASCII text.
- **Space**. This pads with spaces. This is only used with ASCII text.
- **Random**. This pads with random bytes with the last byte defined by the number of padding bytes.

Padding

- After padding (CMS): 68656c6c6f0b0b0b0b0b0b0b0b0b0b0b
  Cipher (ECB): 0a7ec77951291795bac6690c9e7f4c0d
- After padding (Bit): 68656c6c6f80000000000000000000
  Cipher (ECB): 731abffc2e3b2c2b5caa9ca2339344f9
- After padding (ZeroLen): 68656c6c6f0000000000000000000a
  Cipher (ECB): d28e2f7e8e44e068732b292bde444245
- After padding (Null): 68656c6c6f00000000000000000000
  Cipher (ECB): 44479742246045 3d95856eb2a1520ece
- After padding (Space): 68656c6c6f20202020202020202020
  Cipher (ECB): 44479742246045 3d95856eb2a1520ece
- After padding (Random): 68656c6c6*ffc6ecfd884a38798d62*a**0a**
  Cipher (ECB): c2c88b4364d2c2dc6f2cac9ab73c995d

Symmetric Key

Basics

Block or Stream?

Secret Key Methods

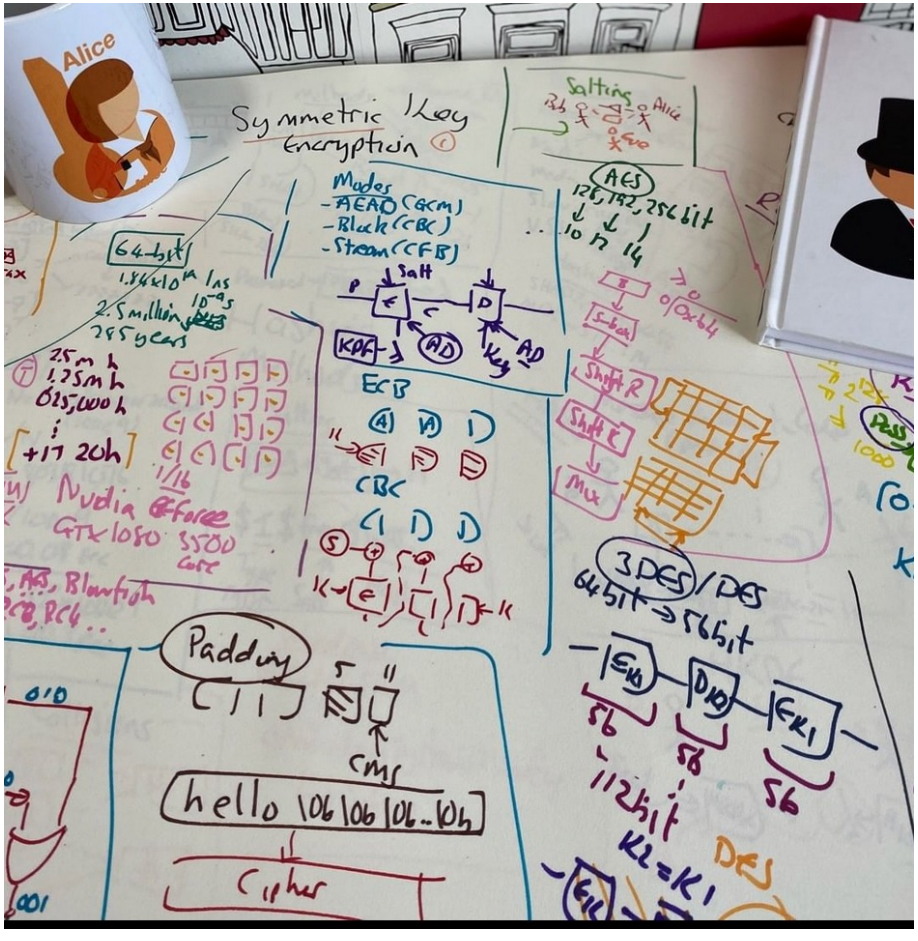
Salting

AES

Key Entropy

Prof Bill Buchanan OBE

https://asecuritysite.com/symmetric



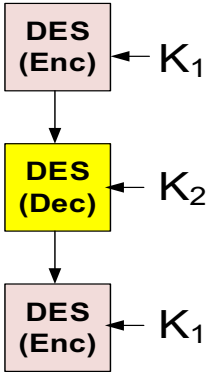
Encryption

Private key

DES

DES. DES encryption algorithm is block cipher and uses a 64-bit block and a 64-bit encryption key.

3DES. DES encryption algorithm is block cipher and uses a 64-bit block and a 64-bit encryption key (of which only 56 bits are actively used in the encryption process). Unfortunately DES has been around for a long time, and the 56-bit version is now easily crackable (in less than a day, on fairly modest equipment). An enhancement, and one which is still fairly compatible with DES, is the 3-DES algorithm. It has three phases, and splits the key into two. Overall the key size is typically 112 bits (2x54 bits - with a combination of the three keys - of which two of the keys are typically the same). The algorithm is EncryptK3(DecryptK2(EncryptK1(message), where K1 and K3 are typically the same (to keep compatibility).



Twofish

Bruce Schneier created Twofish with a general-purpose private key block cipher encryption algorithm.

AES

AES. AES (or Rijndael) is a new block cipher, and is the new replacement for DES, and uses 128-bit blocks with 128, 192 and 256 bit encryption keys. It was selected by NIST in 2001 (after a five year standardisation process). The name Rijndael comes from its Belgium creators: Joan Daemen and Vincent Rijmen.

RC2

RC2. RC2. RC2 ("Rivest Cipher") is a block cipher, and is seen as a replacement for DES. It was created by Ron Rivest in 1987, and is a 64-bit block code and can have a key size from 40 bits to 128-bits (in increments of 8 bits). The 40-bit key version is seen as weak, as the encryption key is so small, but is favoured by governments for export purposes, as it can be easily cracked. In this case the key is created from a Key and an IV (Initialisation Vector). The key has 12 characters (96 bits), and the IV has 8 characters (64 bits), which go to make the overall key.

Blowfish

Blowfish. Bruce Schneier created Blowfish with a general-purpose private key block cipher encryption algorithm.
Blowfish (with CBC). Blowfishcbc. With CBC we split the message into blocks and encrypt each block. The input from the first stage is the IV (Initialisation Vector), and the input to the following stages is the output from the previous stage. In this example we will use Blowfish to encrypt, using CBC.

Others

- Skipjack. Skip jack. Skipjack is a block cipher, using private-key encryption algorithm, and designed by NSA.
- Camellia. Camillia is a block cipher created by Mitsubishi and NTT.
- RC4. RC4 is a stream cipher used in WEP (in wireless encryption).
- Affine. Affine is a stream cipher which uses an equation to encrypt.

Author: Prof Bill Buchanan

3-DES. The DES encryption algorithm uses a **64-bit block** and a 64-bit encryption key (of which only **56 bits** are actively used in the encryption process). Unfortunately DES has been around for a long time, and the 56-bit version is now easily crackable (in less than a day, on fairly modest equipment). An enhancement, and one which is still fairly compatible with DES, is the 3-DES algorithm. It has three phases, and splits the key into two. Overall the key size is typically **112 bits** (2x54 bits - with a combination of the three keys - of which two of the keys are typically the same). The algorithm is:

$Encrypt_{K3}(Decrypt_{K2}(Encrypt_{K1}(message)))$

where K1 and K3 are typically the same (to keep compatibility).

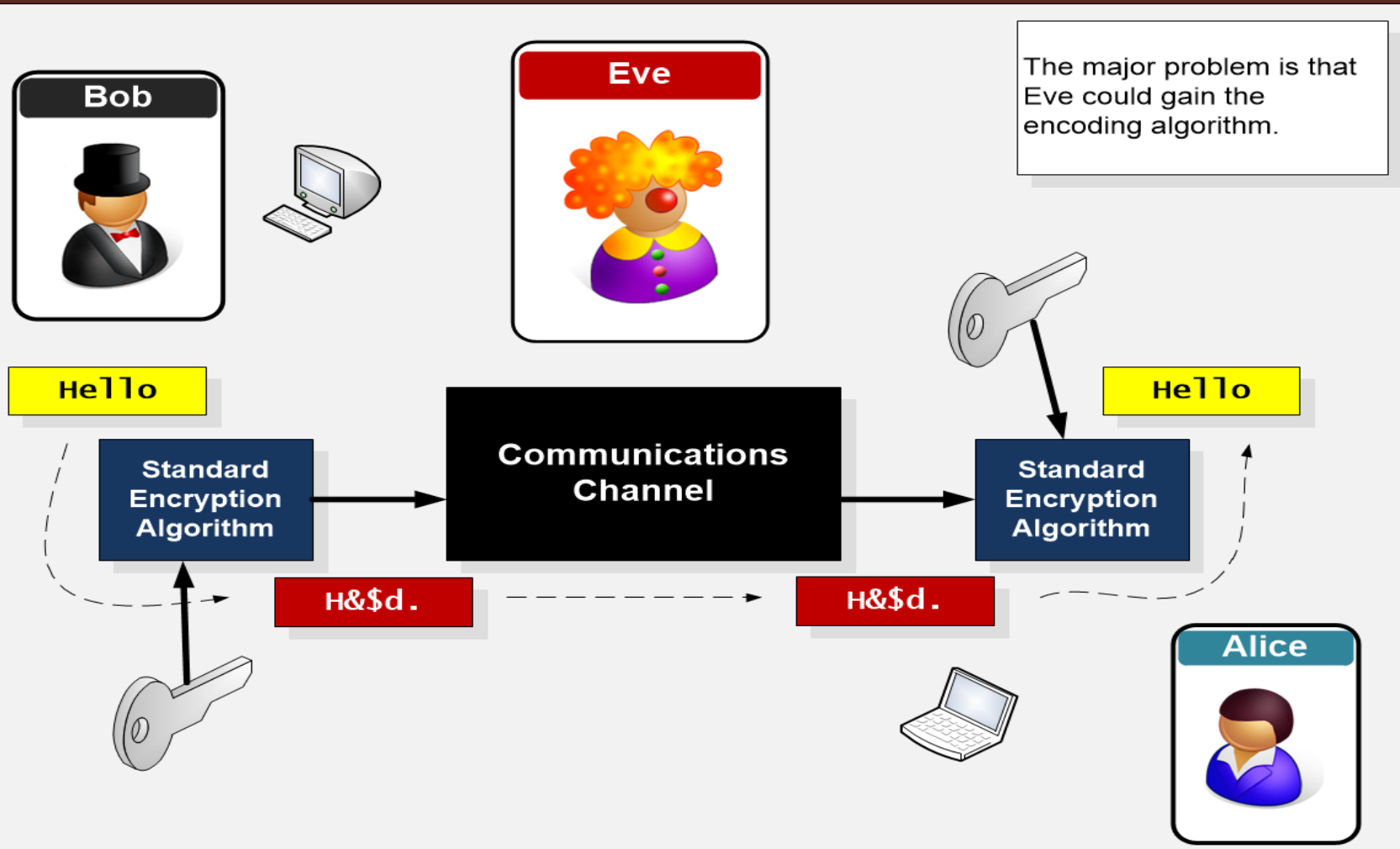
http://asecuritysite.com/encryption/threedes

RC-2. RC2 ("Rivest Cipher") is seen as a replacement for DES. It was created by Ron Rivest in 1987, and is a **64-bit block code** and can have a key size from 40 bits to 128-bits (in increments of 8 bits). The 40-bit key version is seen as weak, as the encryption key is so small, but is favoured by governments for export purposes, as it can be easily cracked. In this case the key is created from a Key and an IV (Initialisation Vector). The key has 12 characters (96 bits), and the IV has 8 characters (64 bits), which go to make the overall key.

http://asecuritysite.com/encryption/rc2

AES/Rijndael. AES (or Rijndael) is the new replacement for DES, and uses **128-bit blocks** with 128, 192 and 256 bit encryption keys. It was selected by NIST in 2001 (after a five year standardisation process). The name Rijndael comes from its Belgium creators: Joan Daemen and Vincent Rijmen.

http://asecuritysite.com/encryption/aes

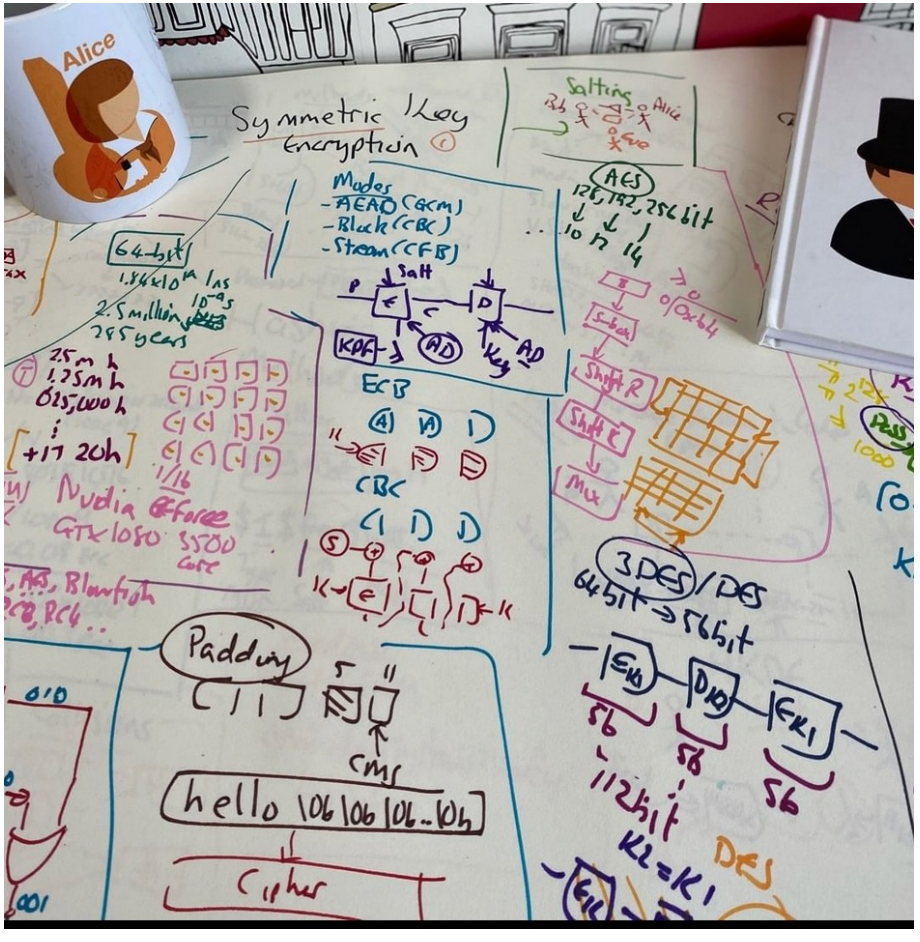

Bob
Eve
The major problem is that Eve could gain the encoding algorithm.
Hello
Standard Encryption Algorithm
Communications Channel
Standard Encryption Algorithm
Hello
H&$d.
H&$d.
Alice

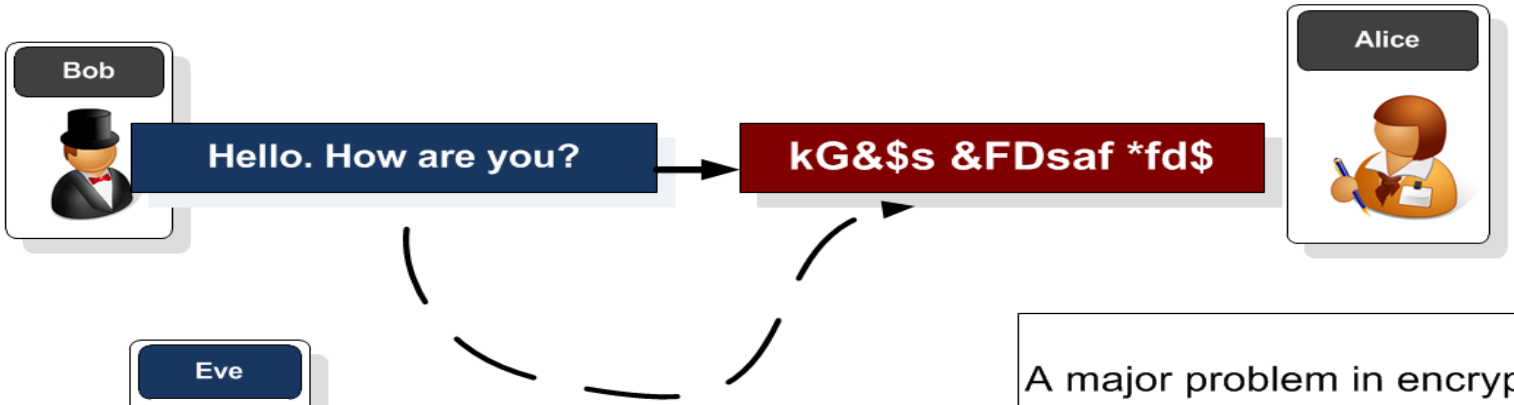
Symmetric Key

Basics
Block or Stream?
Secret Key Methods
Salting
AES
Key Entropy

Prof Bill Buchanan

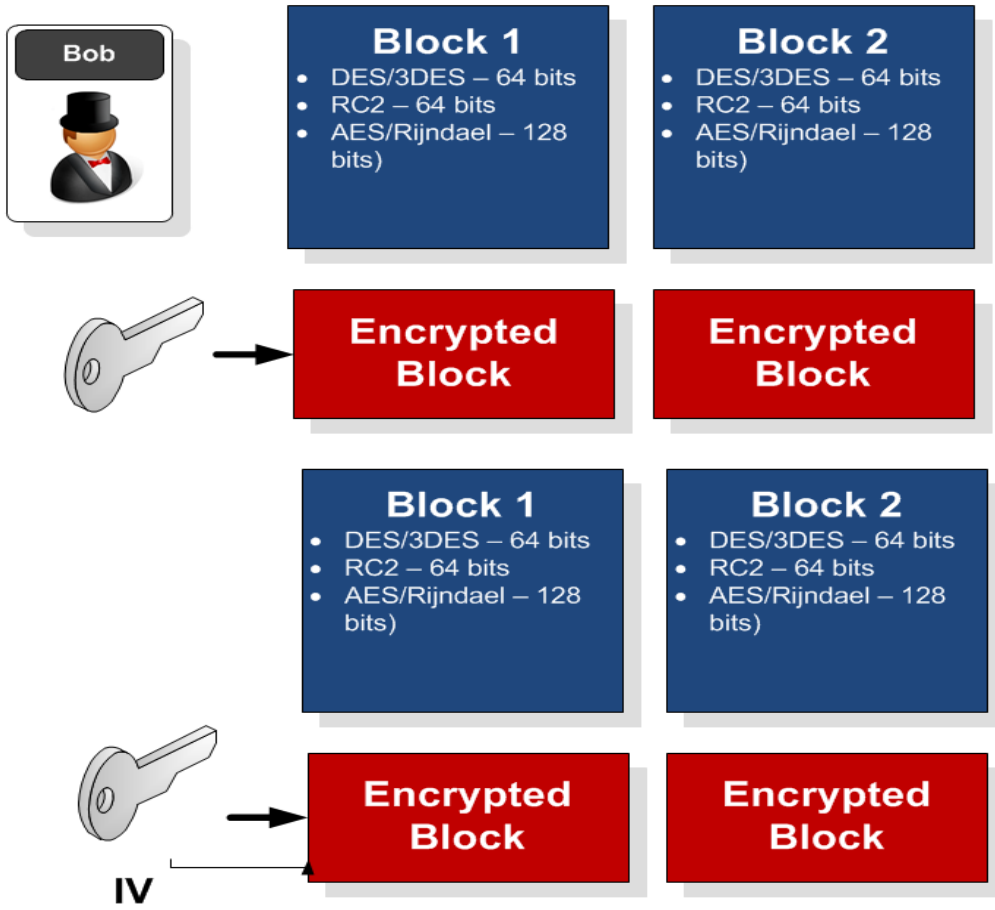
https://asecuritysite.com/symmetric





A major problem in encryption is playback where an intruder can copy an encrypted message and play it back, as the same plain text will always give the same cipher text.

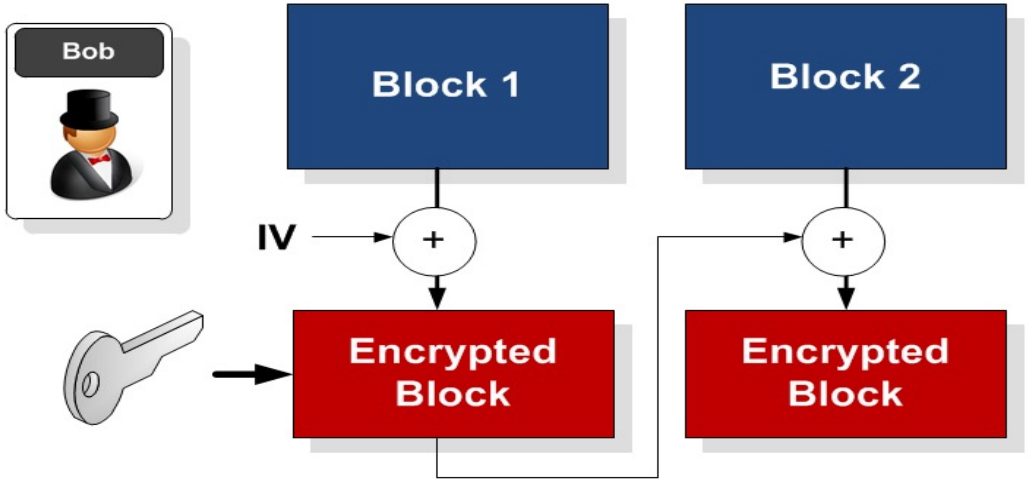
The solution is to add **salt** to the encryption key, as that it changes its operation from block-to-block (for block encryption) or data frame-to-data frame (for stream encryption)



Electronic Code Book (ECB) method. This is weak, as the same cipher text appears for the same blocks.

Hello → 5ghd%43f=
Hello → 5ghd%43f=

Adding salt. This is typically done with an IV (Initialisation Vector) which must be the same on both sides. In WEP, the IV is incremented for each data frame, so that the cipher text changes.



Cipher Block Chaining (CBC). This method uses the IV for the first block, and then the results from the previous block to encrypt the current block.

Original image

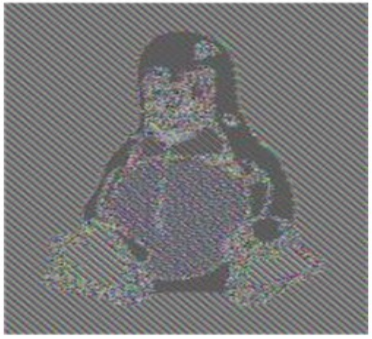


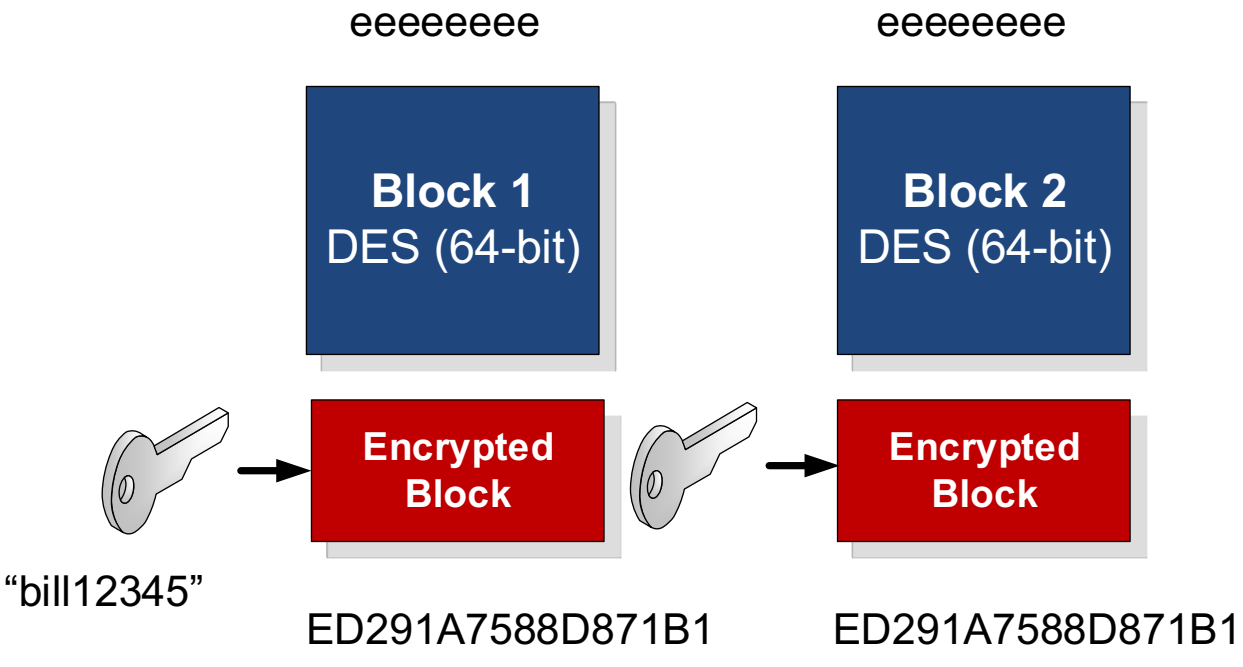
Image with AES using ECB

Image with AES using CBC

Image ref: http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee

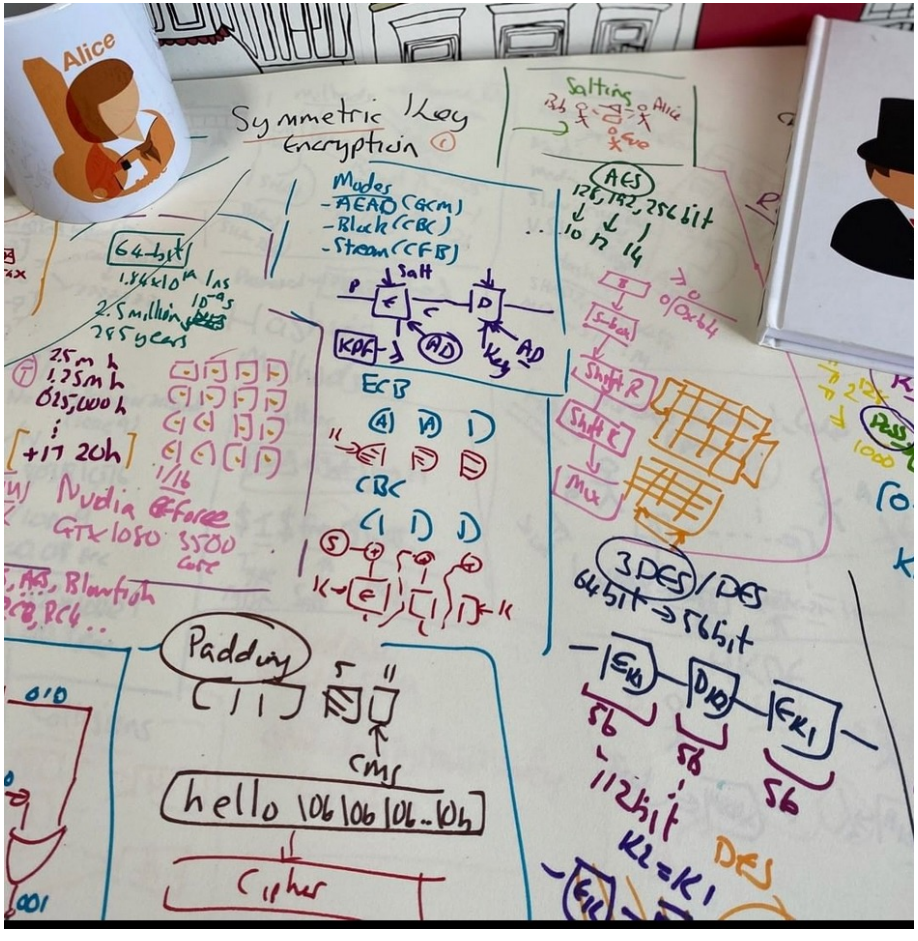
[eeeeeeee] [eeeeeeee] [eeeeeeeee][eeeeeeee] [eeeeeeee] [eeeeeeee][eeeeee <PADDING>]



ED291A7588D871B1ED291A7588D871B1ED291A7588D
871B1ED291A7588D871B1ED291A7588D871B1ED291A
7588D871B18D6DF6795DDEDACD

Symmetric Key

Basics
Block or Stream?
Secret Key Methods
Salting
AES
Key Entropy



Prof Bill Buchanan

https://asecuritysite.com/symmetric

Key Entropy

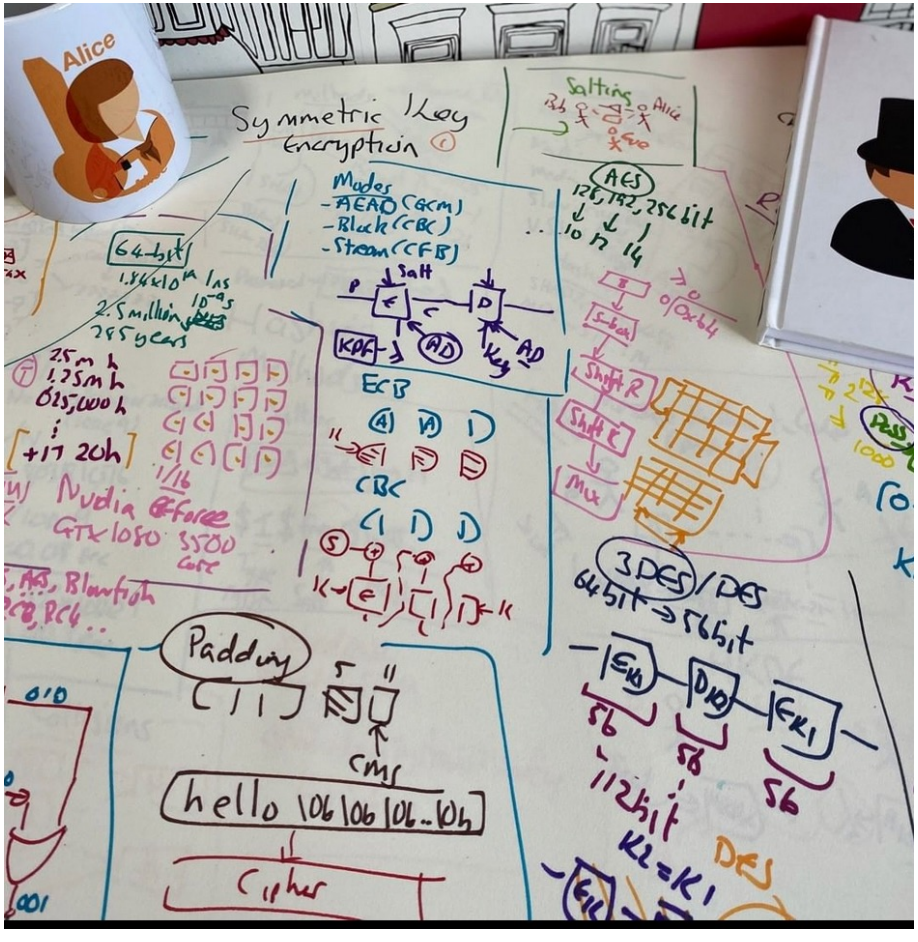
$$Key\ Entropy = log_2(Phrases) = \frac{log_{10}(Phrases)}{log_{10}(2)}$$

$$Key\ Entropy = log_2(26^8) = \frac{log_{10}(26^8)}{log_{10}(2)} = 37.6\ bits$$

Password definition	Number of possible characters	Total number of passwords	Entropy (bits)
[0-9]	10	100,000,000	26.6
[a-z]	26	2.08827×10^{11}	37.6
[a-zA-Z]	52	5.34597×10^{13}	45.6
[a-zA-Z0-9]	62	2.1834×10^{14}	47.6
[a-zA-Z0-9$%!@+=]	68	4.57163×10^{14}	48.7

Symmetric Key

Basics
Block or Stream?
Secret Key Methods
Salting
AES
Key Entropy



Prof Bill Buchanan OBE

https://asecuritysite.com/symmetric