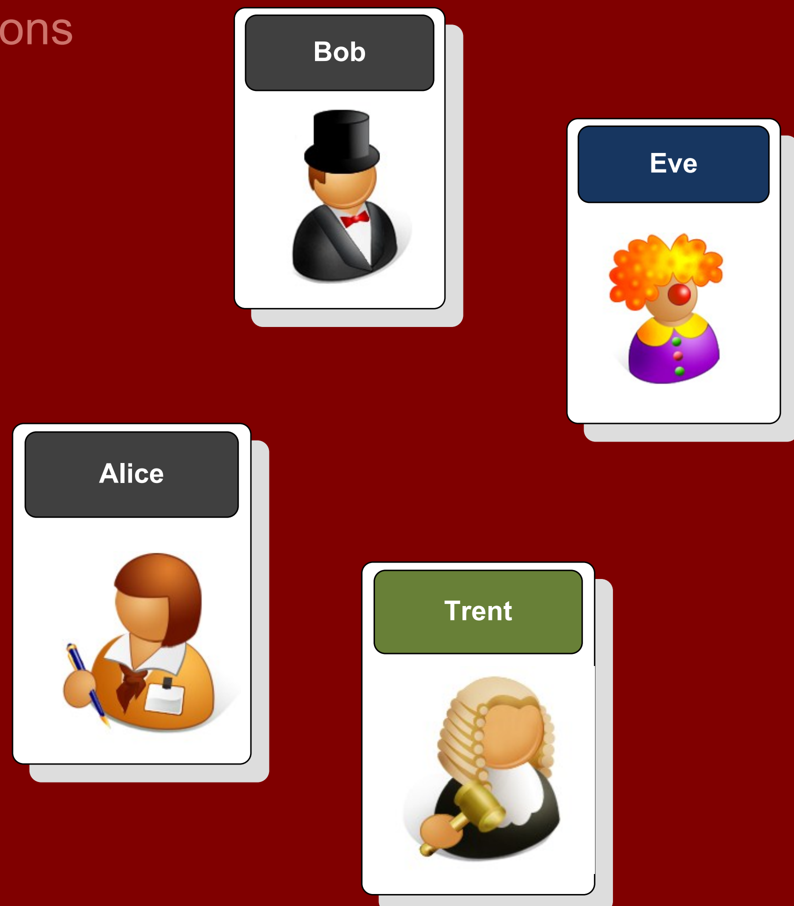
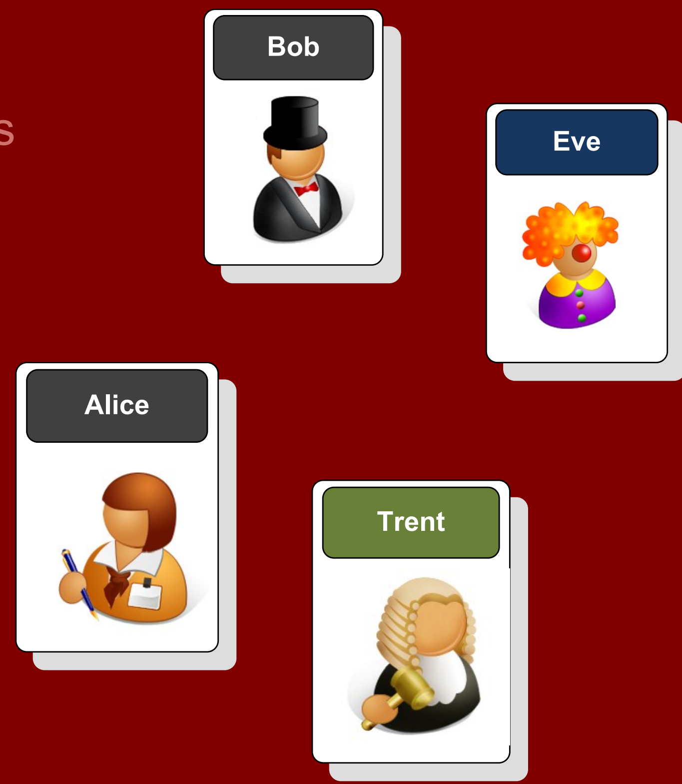
Encryption

Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption



Encryption



Introduction

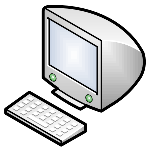
Meet the cast

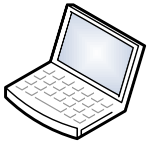
Bob and Alice – good
Eve – bad
Trust – trusted?

Eve

Intruder

Bob





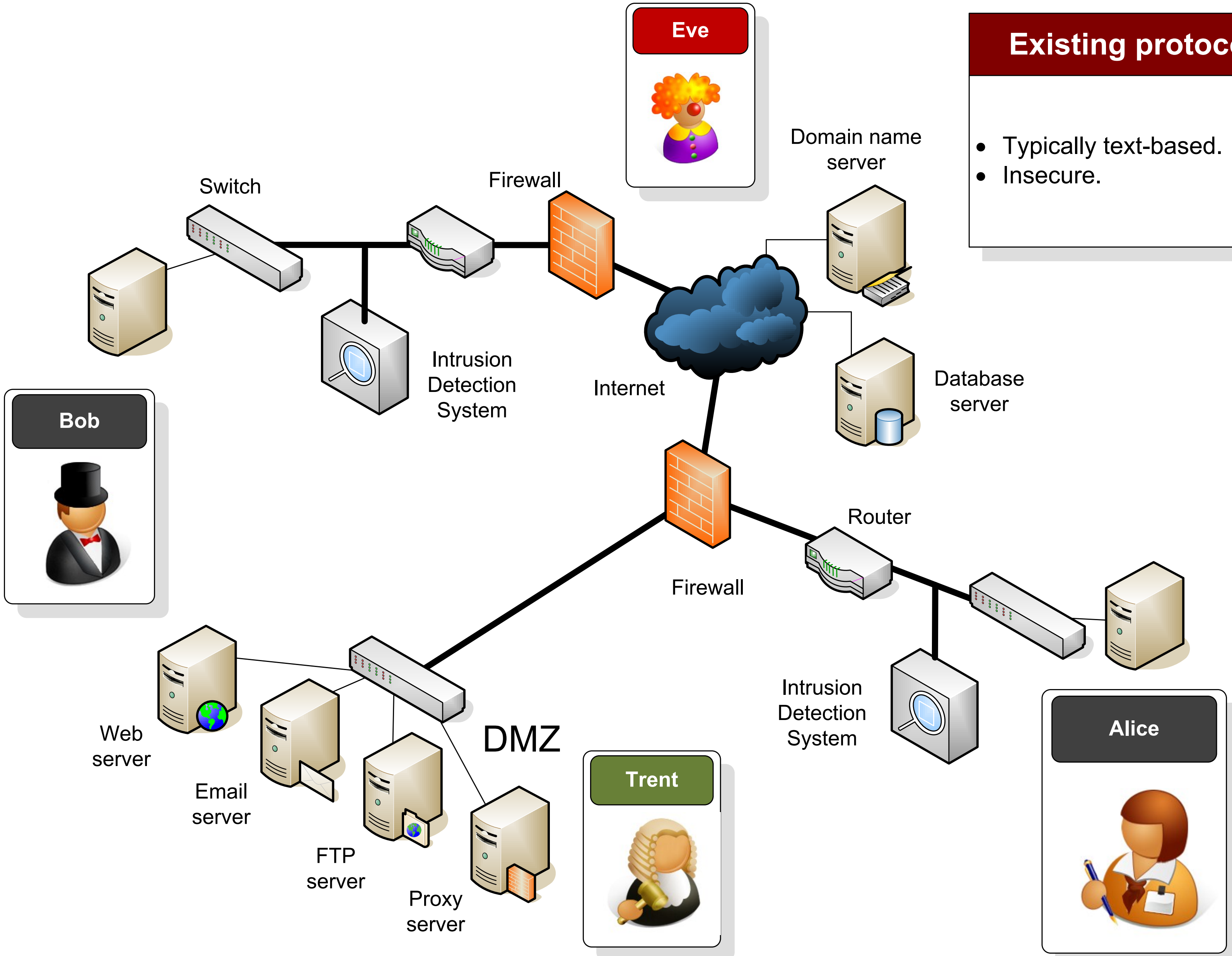
Alice

Trent

Trusted third party

Existing protocols

- Typically text-based.
- Insecure.

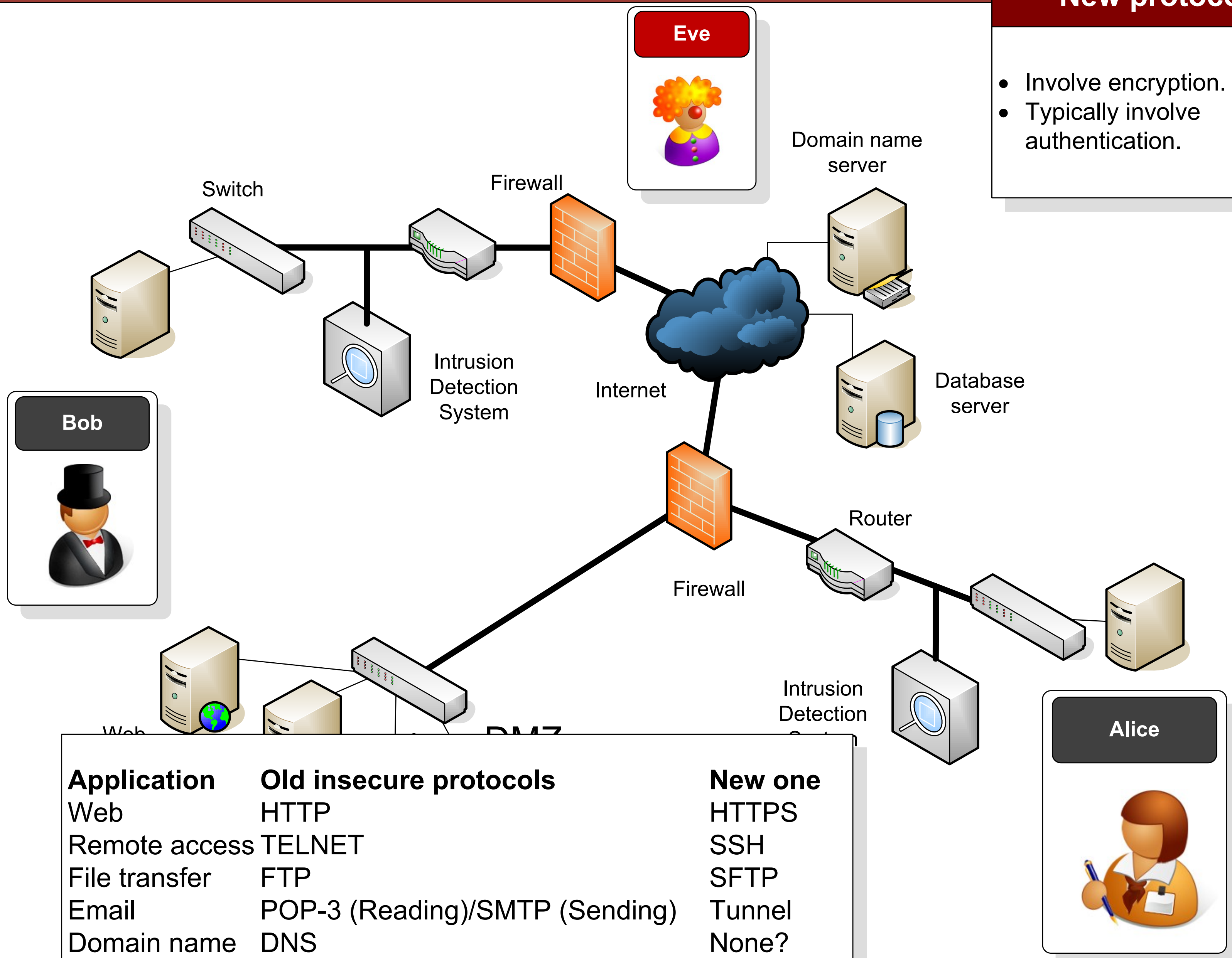


Introduction

Encryption

New protocols

- Involve encryption.
- Typically involve authentication.

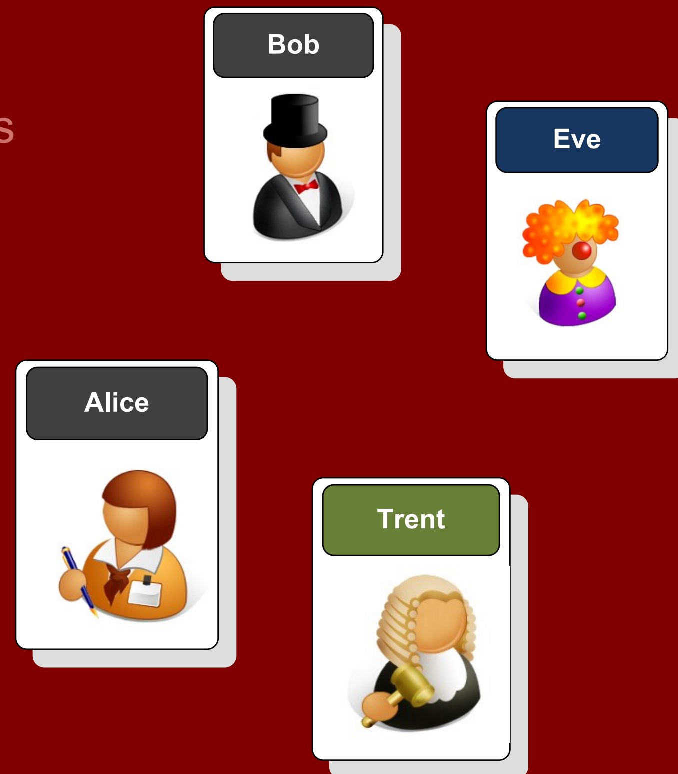


Application	Old insecure protocols	New one
Web	HTTP	HTTPS
Remote access	TELNET	SSH
File transfer	FTP	SFTP
Email	POP-3 (Reading)/SMTP (Sending)	Tunnel
Domain name	DNS	None?


Bruce Schneier
Twofish,
Blowfish
Vincent Rijmen and Joan Daemen
AES
Modern private key encryption
Rivest, Shamir & Aldeman
Public-k
encryp
Ron Rivest
Hashing
Whitfield Diffie
Key interchange
Phil Zimmerman
PGP Encryption

Encryption

Encryption



Before electronic communications

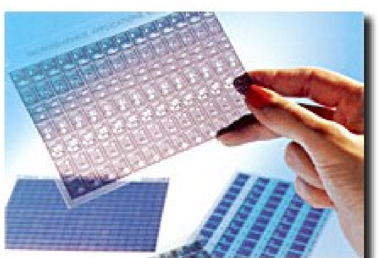
Secret Communications

- Quilts
- Carrier pigeon
- Smoke signals
- Etc...

Quilt patterns (used by slaves to escape)

Carrier pigeon

Smoke signals



Microfiche

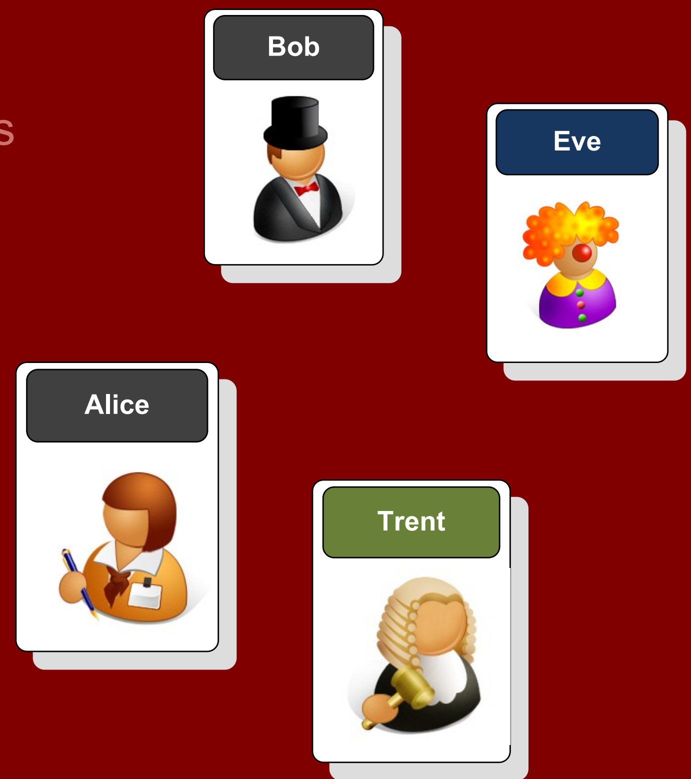
Code talkers: Navajo words

Before electronic communications

Encryption

Encryption

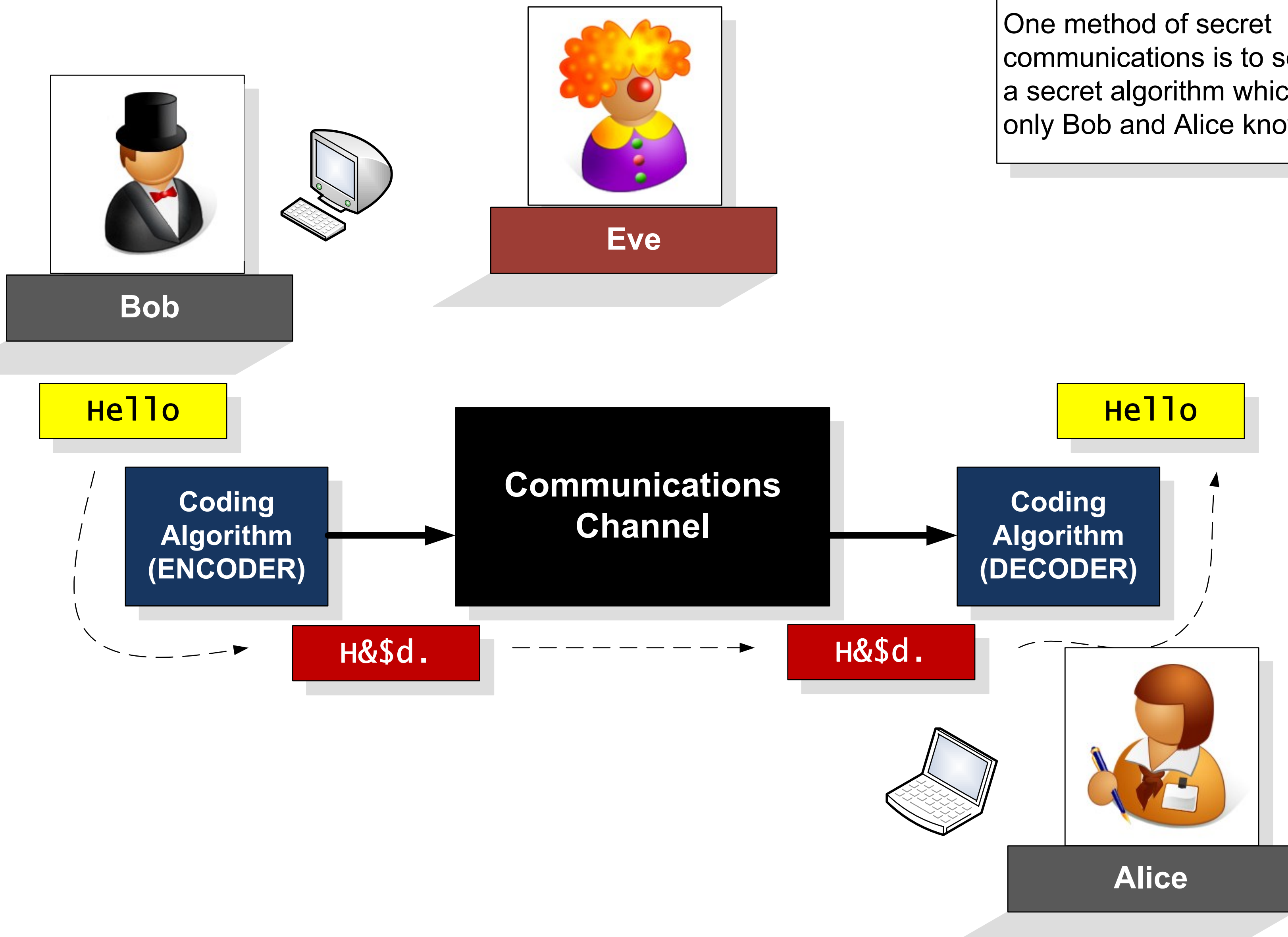
- Introduction
- Before electronic communications
- Codes
- A few fundamentals
- Key-based encryption
- Cracking the code
- Brute force
- Block or stream
- Private-key methods
- Encryption keys
- Passing keys
- Public-key encryption
- One-way hash
- Encrypting disks
- PGP encryption



Codes

Secret Communications

One method of secret communications is to setup a secret algorithm which only Bob and Alice know



Coding

Encryption

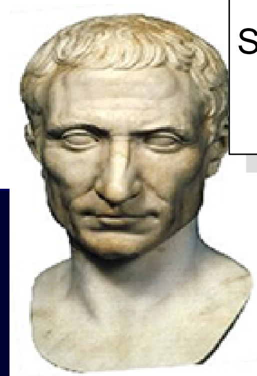
Caesar code

Simple alphabet shifting

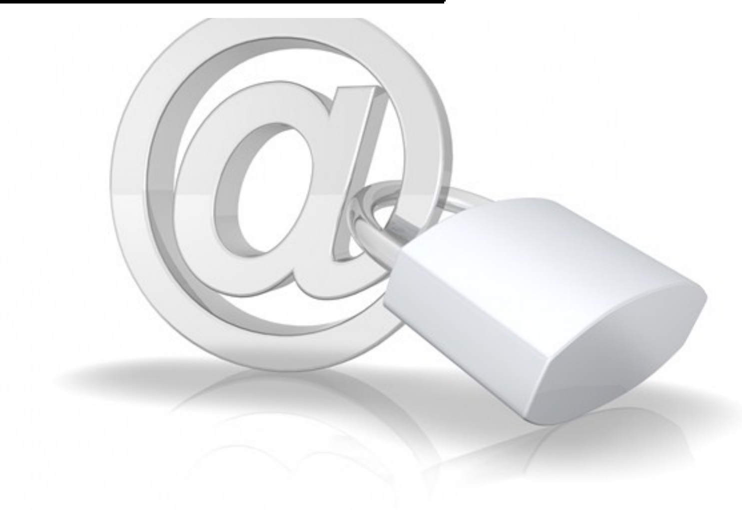
Caesar code

```
abcdefghijklmnopqrstuvwxyz
YZABCDEFGHIJKLMNOPQRSTUVWX

RFC ZMW QRMMB ML RFC ZSPLGLE BCAI
```



25 code mappings



Coding

Encryption

Code Mapping

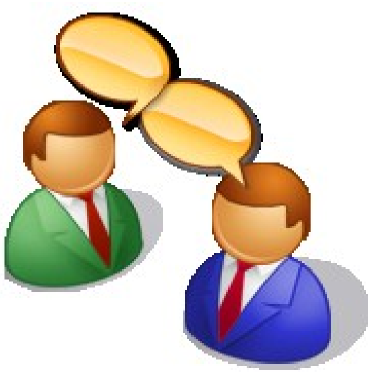
Code mapping scrambles the alphabet ..

403 million billion billion codes.

Code mapping

```
abcdefghijklmnopqrstuvwxyz
MGPOAFZBCDIEHXJKLNTQRWSUVY

QBCT CT MX AUMHKEA KCAPA JF QAUQ
```



4.03×10^{26} codes

Coding

Encryption

Letters (%)		Digrams (%)		Trigrams (%)		Words (%)	
E	13.05	TH	3.16	THE	4.72	THE	6.42
T	9.02	IN	1.54	ING	1.42	OF	4.02
O	8.21	ER	1.33	AND	1.13	AND	3.15
A	7.81	RE	1.30	ION	1.00	TO	2.36
N	7.28	AN	1.08	ENT	0.98	A	2.09
I	6.77	HE	1.08	FOR	0.76	IN	1.77
R	6.64	AR	1.02	TIO	0.75	THAT	1.25
S	6.46	EN	1.02	ERE	0.69	IS	1.03
H	5.85	TI	1.02	HER	0.68	I	0.94
D	4.11	TE	0.98	ATE	0.66	IT	0.93
L	3.60	AT	0.88	VER	0.63	FOR	0.77
C	2.93	ON	0.84	TER	0.62	AS	0.76
F	2.88	HA	0.84	THA	0.62	WITH	0.76
U	2.77	OU	0.72	ATI	0.59	WAS	0.72
M	2.62	IT	0.71	HAT	0.55	HIS	0.71
P	2.15	ES	0.69	ERS	0.54	HE	0.71
Y	1.51	ST	0.68	HIS	0.52	BE	0.63
W	1.49	OR	0.68	RES	0.50	NOT	0.61
G	1.39	NT	0.67	ILL	0.47	BY	0.57
B	1.28	HI	0.66	ARE	0.46	BUT	0.56
V	1.00	EA	0.64	CON	0.45	HAVE	0.55
K	0.42	VE	0.64	NCE	0.43	YOU	0.55
X	0.30	CO	0.59	ALL	0.44	WHICH	0.53
J	0.23	DE	0.55	EVE	0.44	ARE	0.50
Q	0.14	RA	0.55	ITH	0.44	ON	0.47
Z	0.09	RO	0.55	TED	0.44	OR	0.45

Code Mapping

Code mapping can typically be easily cracked by analysing the probability of the mapped letters.

Vigenere code

Moves the mapping depending on a keyword (in this case “GREEN”)

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Hello
GREEN

Coding

Encryption

Coding

Encryption

Vigenere code

Moves the mapping depending on a keyword (in this case “GREEN”)

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	**N**	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Hello
GREEN
N

Vigenere code

Moves the mapping depending on a keyword (in this case "GREEN")

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	**V**	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Hello
GREEN
NV

Coding

Encryption

Coding

Encryption

Vigenere code

Moves the mapping depending on a keyword (in this case “GREEN”)

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	**V**	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Hello

GREEN

NVP

Homophonic substitution code

Number of codes varies with the probability of the letter.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
07	11	17	10	25	08	44	19	02	18	41	42	40	00	16	01	15	04	06	05	13	22	45	12	55	47
31	64	33	27	26	09	83	20	03			81	52	43	30	62		24	34	23	14		46		93	
50		49	51	28			21	29			86		80	61			39	56	35	36					
63			76	32			54	53			95		88	65			58	57	37						
66				48			70	68					89	91			71	59	38						
77				67			87	73						94			00	90	60						
84				69										96					74						
				72															78						
				75															92						
				79																					
				82																					
				85																					

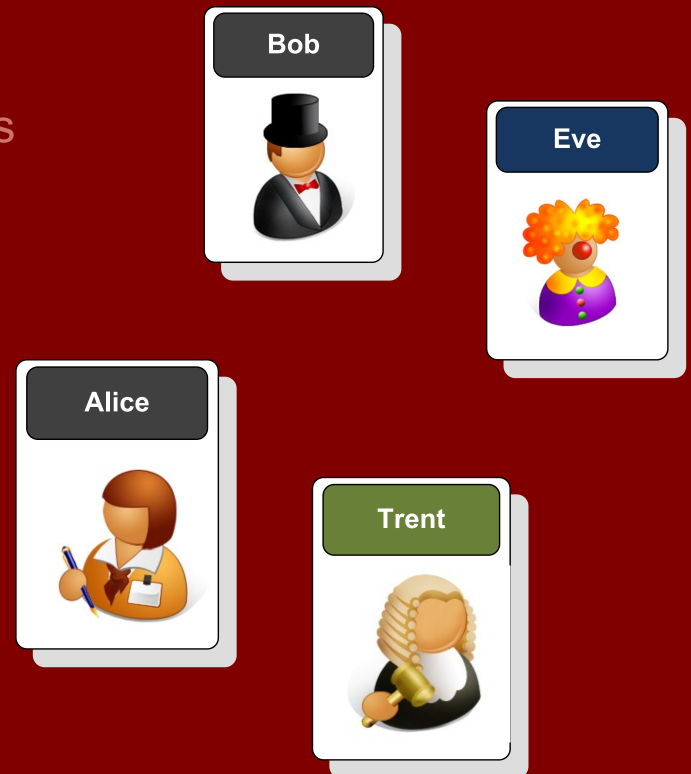
Plaintext	h	e	l	l	o	e	v	e	r	y	o	n	e
Ciphertext:	19	25	42	81	16	26	22	28	04	55	30	00	32

Coding

Encryption

Encryption

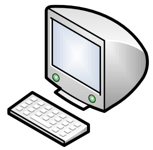
Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption



A few fundamentals

Viewing binary

Binary values are difficult to view/edit, thus encrypted values are typically converted to hex or Base-64.



Bob

'A' 'B' 'C' 'D'

ASCII characters

01000001 01000010
01000011 01000100

Byte values

Encryption

01011110 00100000
11100110 10101010

Hex

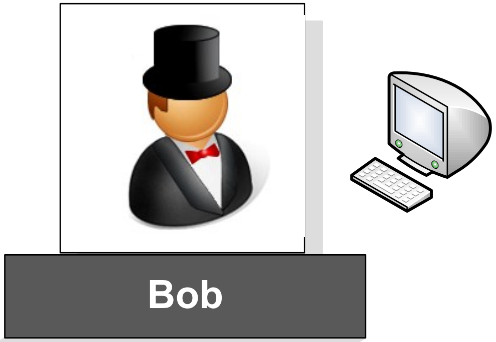
5e 20 e6 aa

Base-64

XiDmqg

Viewing binary

With hexadecimal, the bit stream is split into groups of four, and converted into hex values (0-9,A-F)



0101 1110 0010 0000 1110 0110 1010 1010

Bit stream

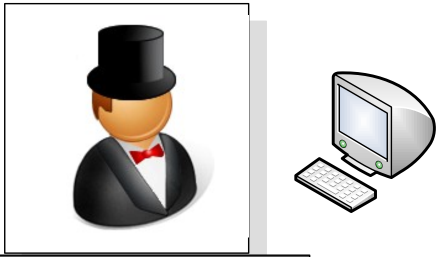
5 e 2 0 e 6 a a

Hex

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Viewing binary

With Base-64, the bits are split into groups of six, and then converted. Base-64 is used extensively on the Internet (such as in email).



Bob

010111 100010 000011 100110 101010 10

Bit stream

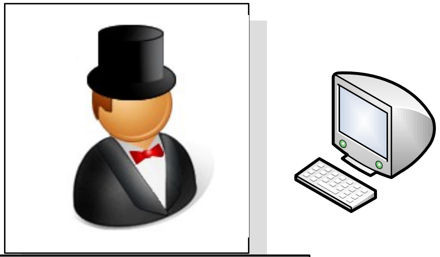
X i D m q g

Base-64

Val	Enc	Val	Enc	Val	Enc	Val	Enc
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

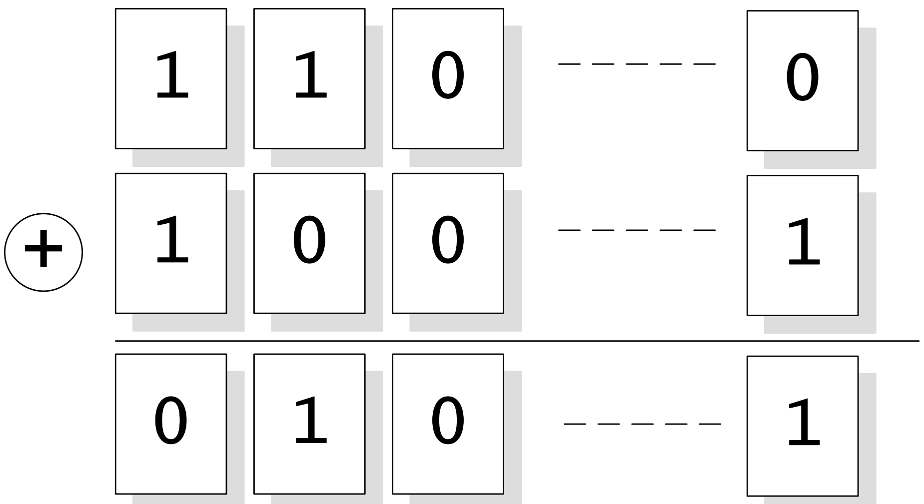
Encryption operators

The two main operators used in encryption are Ex-OR and ROL/ROR, as they are fast, and preserve info.

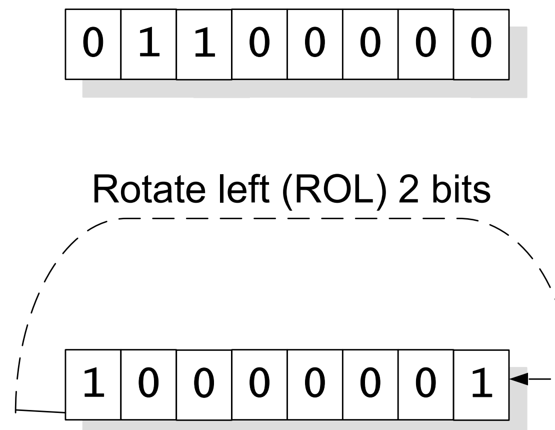


Bob

The two main operators used in encryption are Ex-OR and ROR/ROL



+ **Exclusive-OR operation**

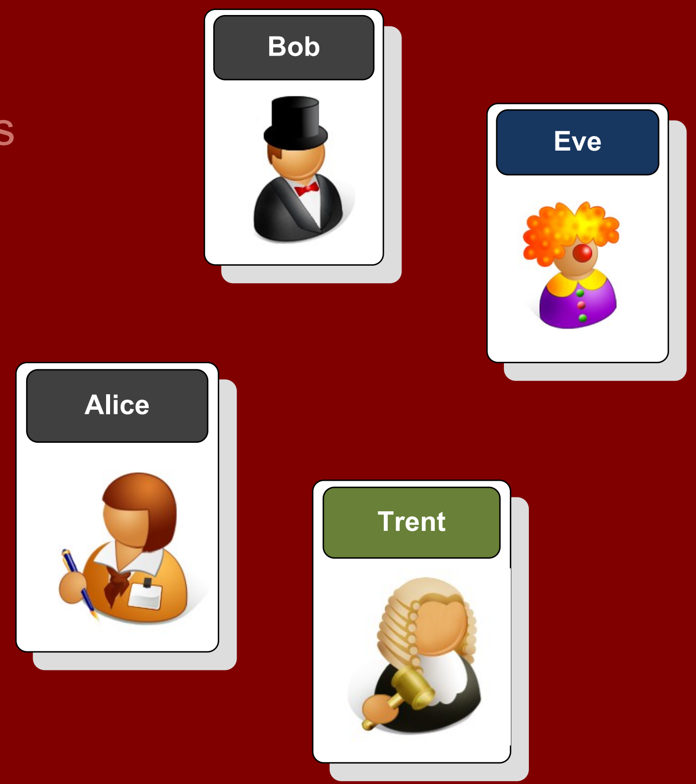


Rotate left (ROL)

Rotate right (ROR)

Encryption

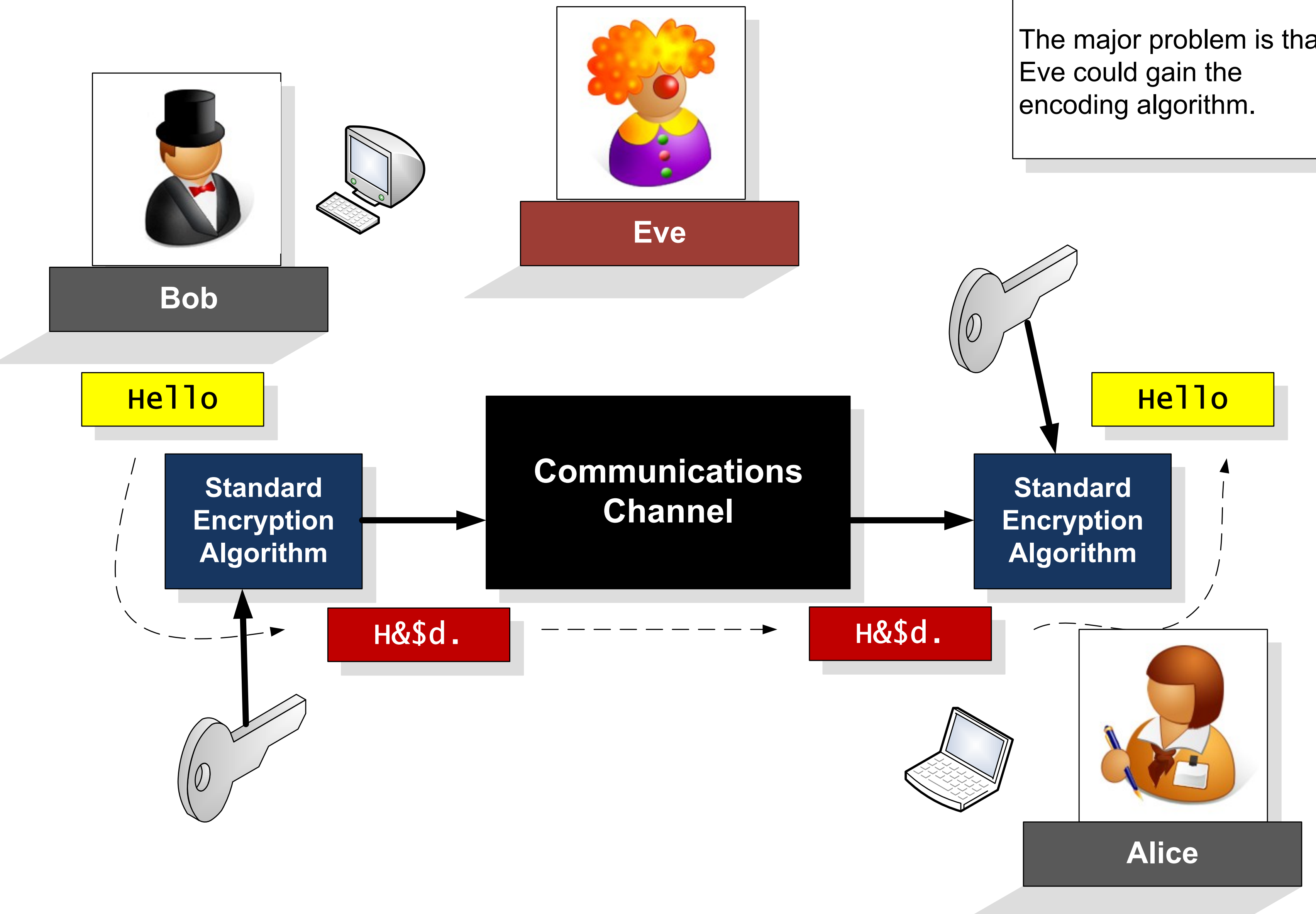
Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption



Key-based encryption

Key-encryption

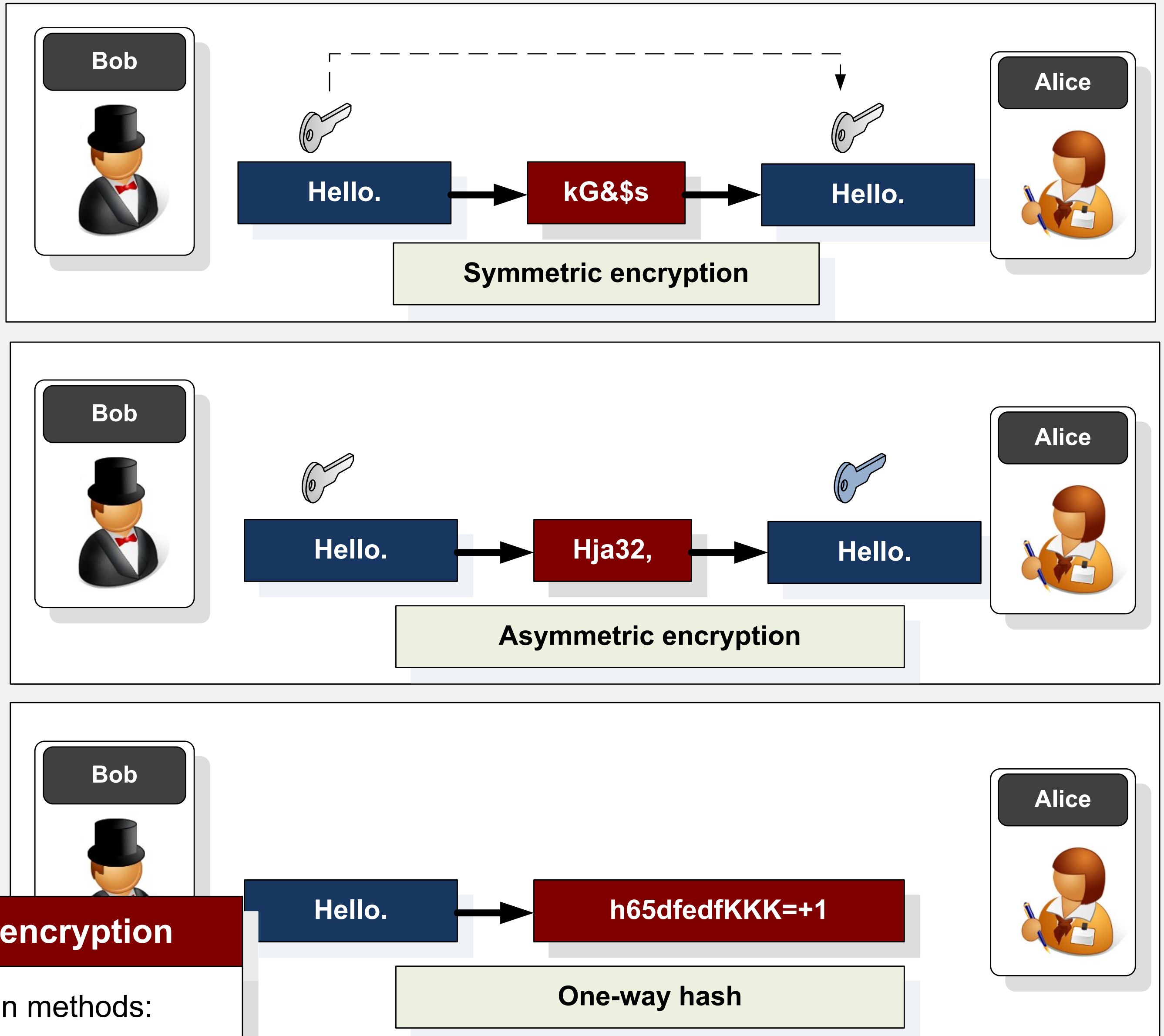
The major problem is that Eve could gain the encoding algorithm.



Key-based Encryption

Encryption

Key-based Encryption



Private-key:
RC2, RC4, DES, 3DES, AES

Public-key:
RSA, DSA (factoring prime numbers)
FIPS 186-2, ElGamal (Elliptic curve)

Hashing:
MD5, SHA-1

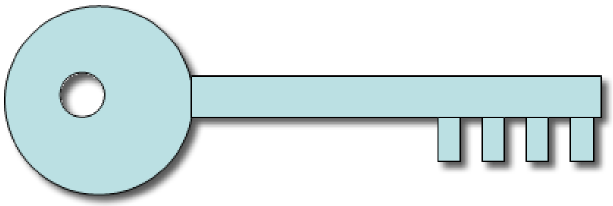
Strength: 80-bit DES -> 1024 RSA -> 160 bit Elliptic

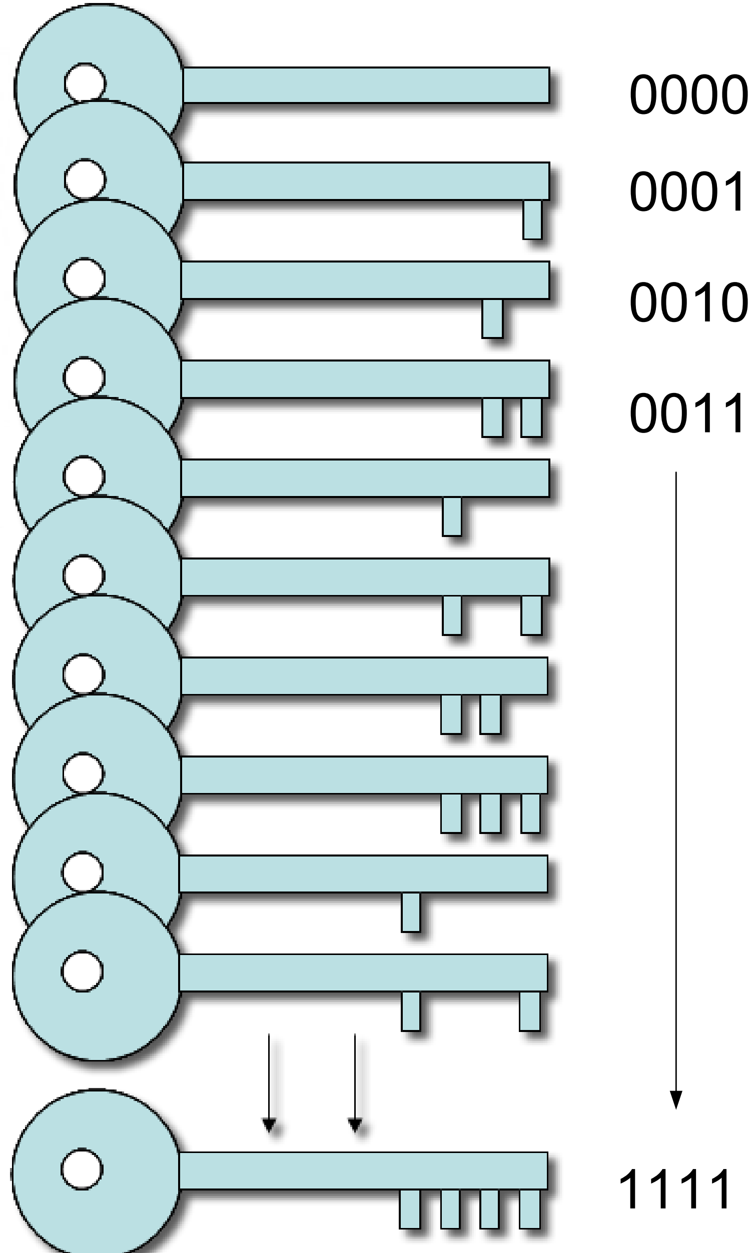
Key-encryption

Three main methods:
Private-key.
Public-key.
One-way hash.

How safe is the key?
- the more keys … the less likely it is to find the key.

For example, if we have a key with four notches … each which can exist or not … how many keys can we have?

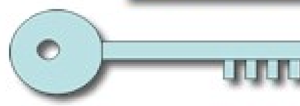




16 key combinations
2 to the power of 4 (2^4)

Width of Napier (100m)

Width of Edinburgh (6 miles)



Earth to the Moon
93,000,000 miles

If each key was 1mm, and each key was laid end-on-end, what is the distance spanned for all the possible 64-bit electronic keys?

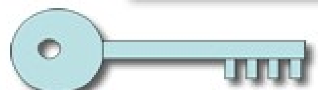
Width of the Milky Way
90,000 light years across

Width of the Solar System
3,666,000,000 miles

- Size would be somewhere between the Milky Way and the Universe

Width of Napier (100m)

Width of Edinburgh (6 miles)

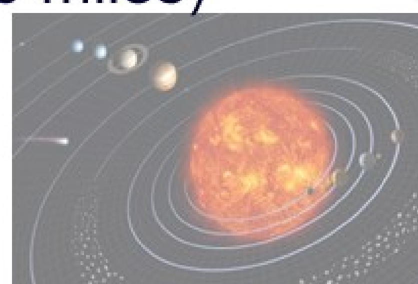


If each key was 1mm, and each key was laid end-on-end, what is the distance spanned for all the possible 64-bit electronic keys?
(1,300,000,000,000,000 miles)

Earth to the Moon

93,000,000 miles

Width of the Milky Way
90,000 light years across

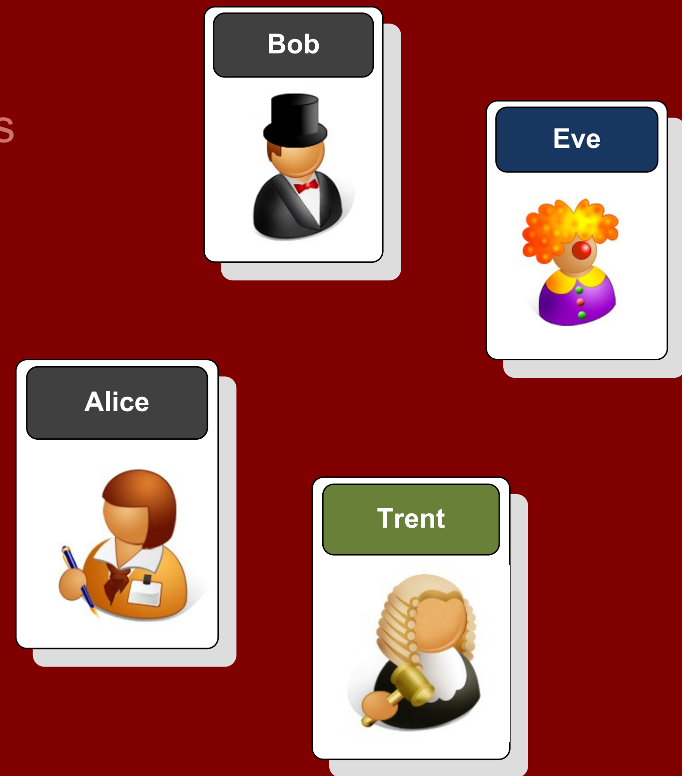


Width of the Solar System

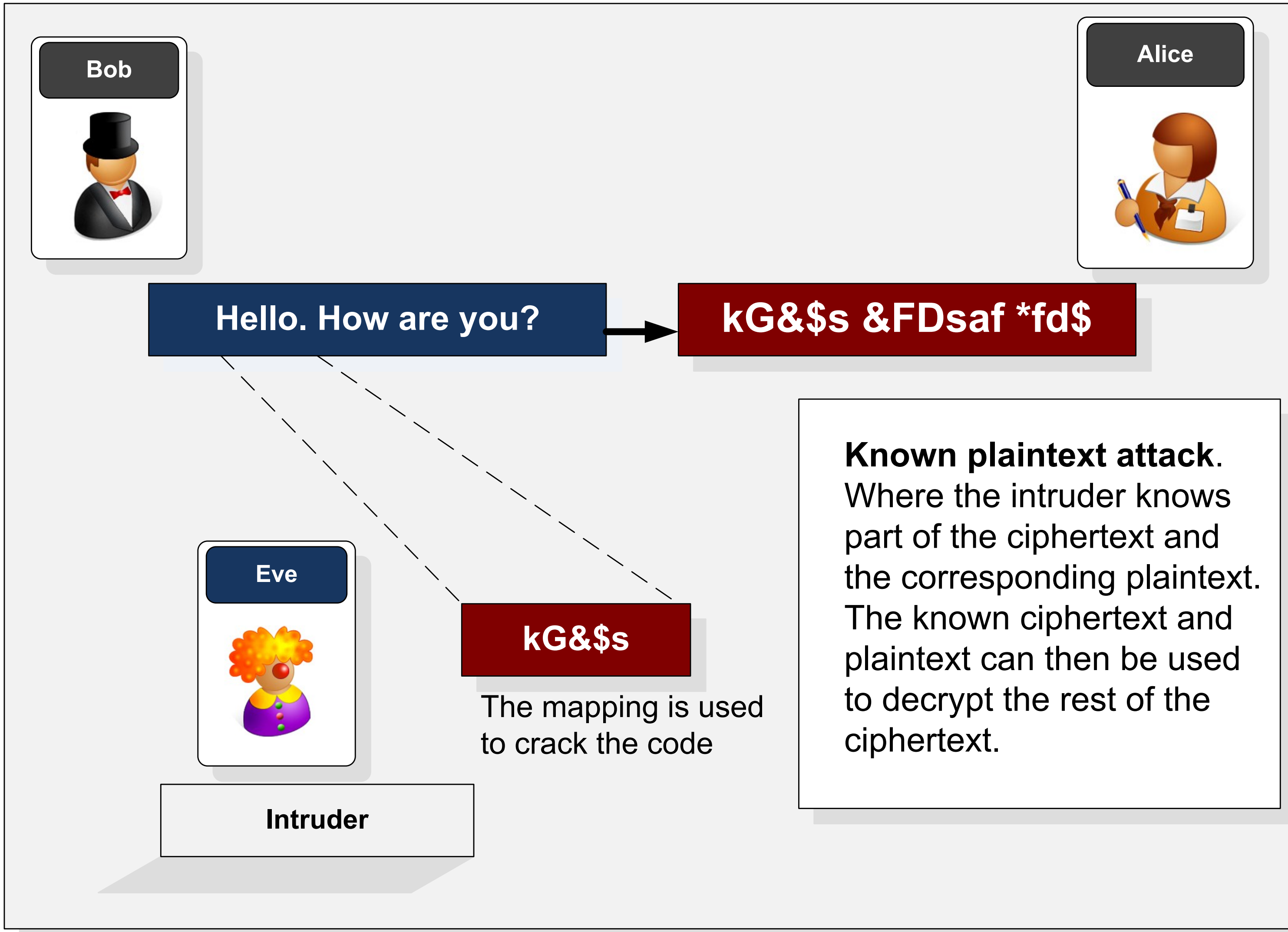
3,666,000,000 miles

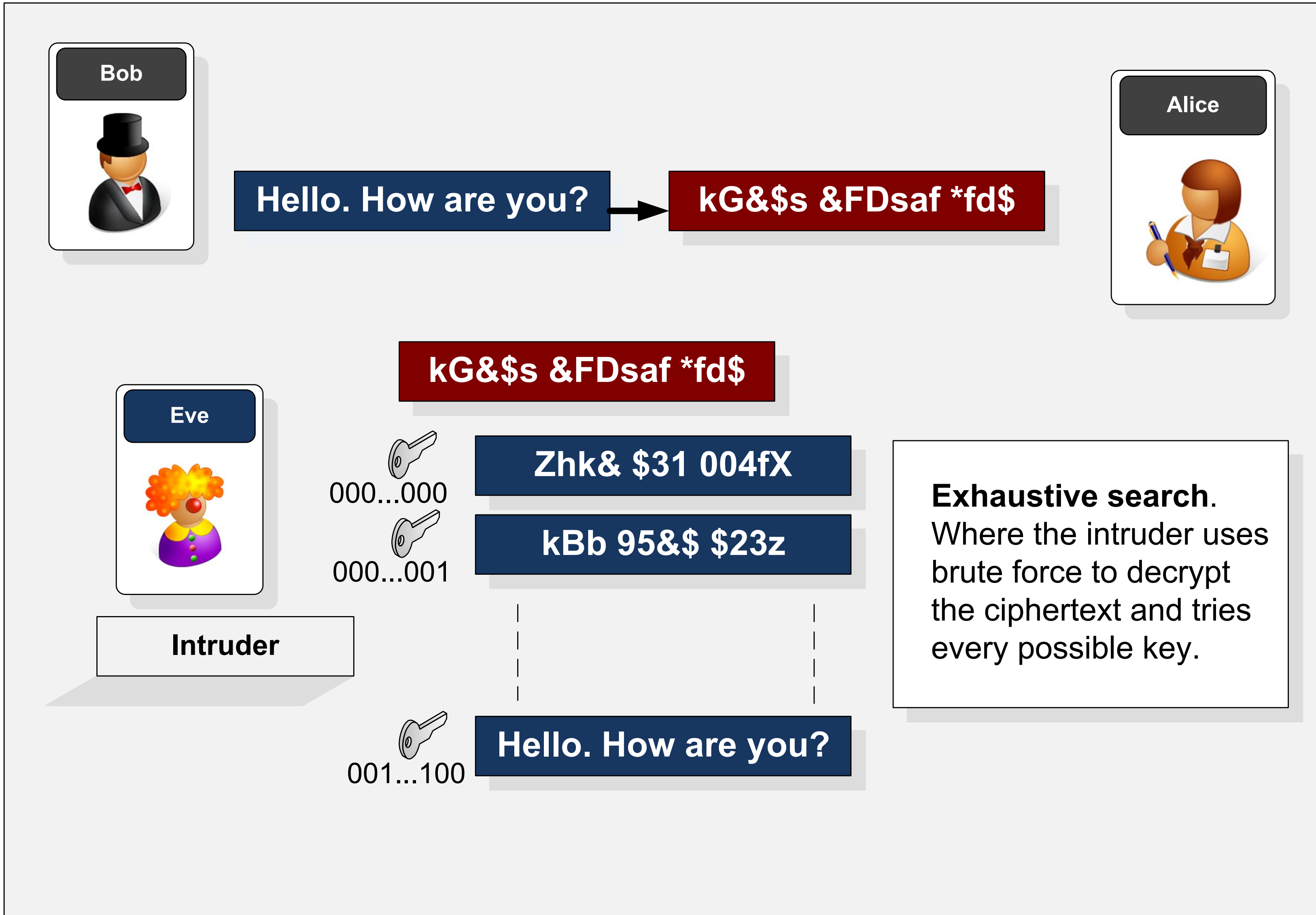
Encryption

Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption

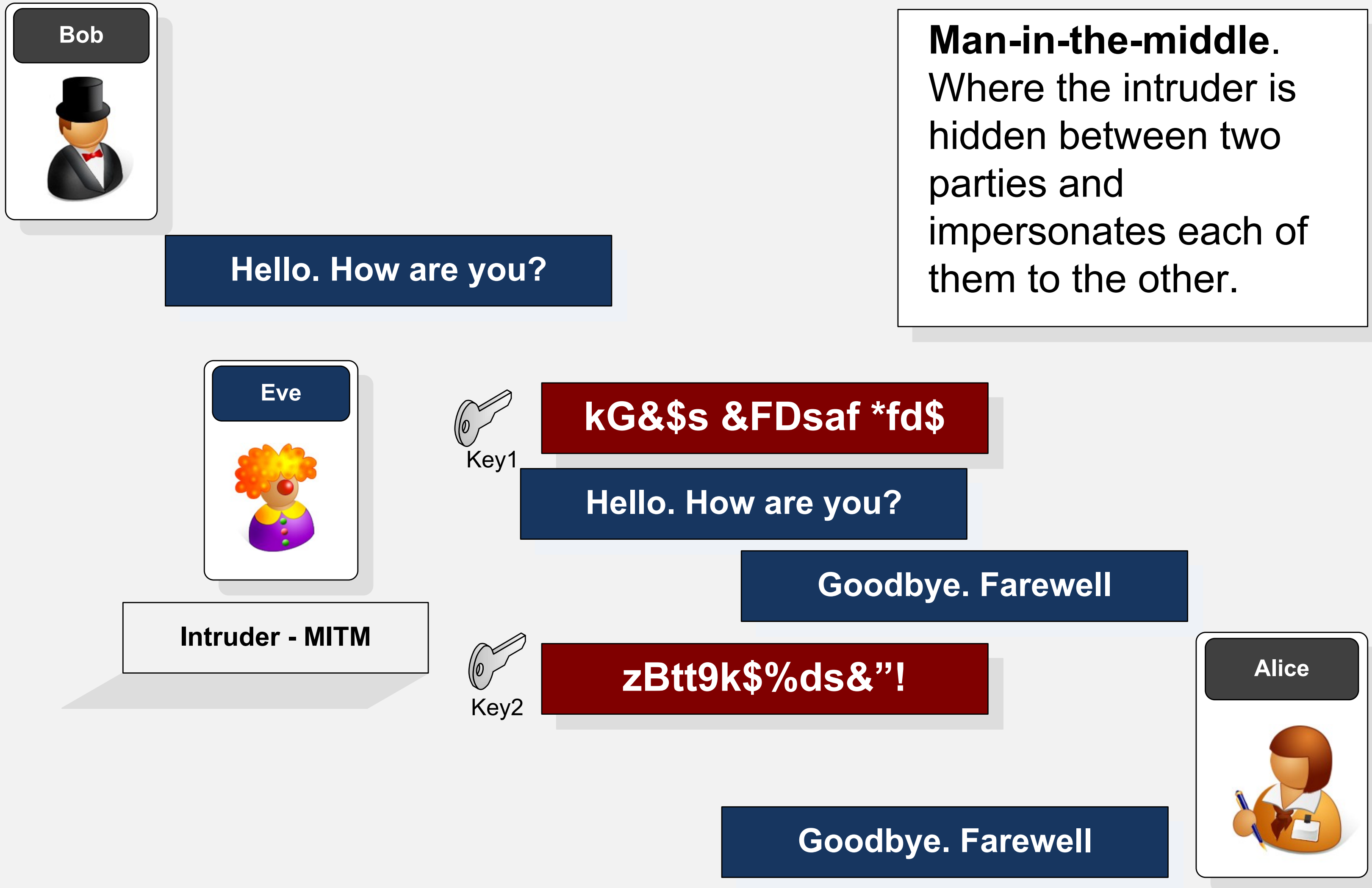


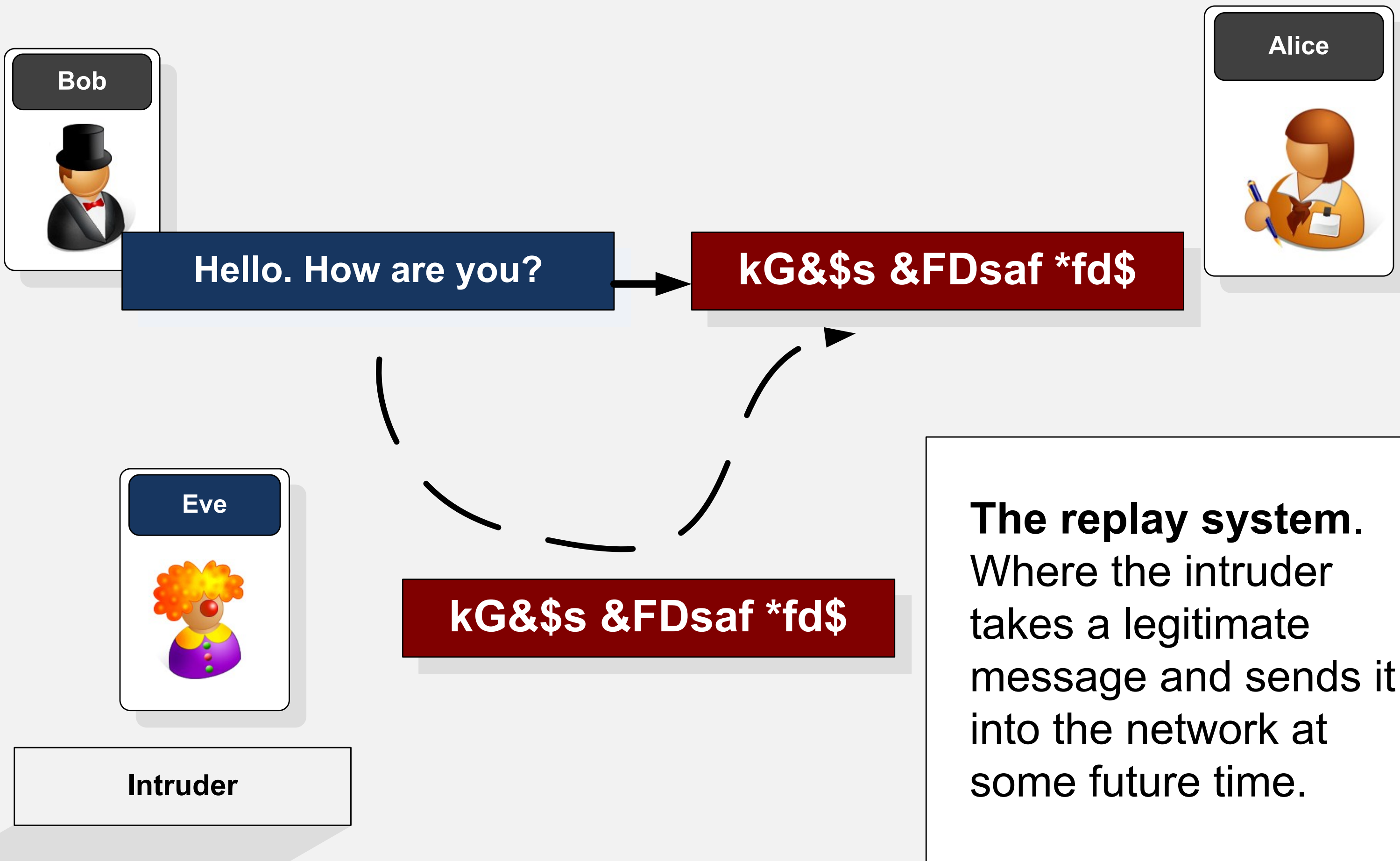
Cracking the code


Bob
Alice
Hello. How are you?
kG&$s &FDsaf *fd$
Eve
kG&$s
The mapping is used to crack the code
Intruder
Known plaintext attack. Where the intruder knows part of the ciphertext and the corresponding plaintext. The known ciphertext and plaintext can then be used to decrypt the rest of the ciphertext.

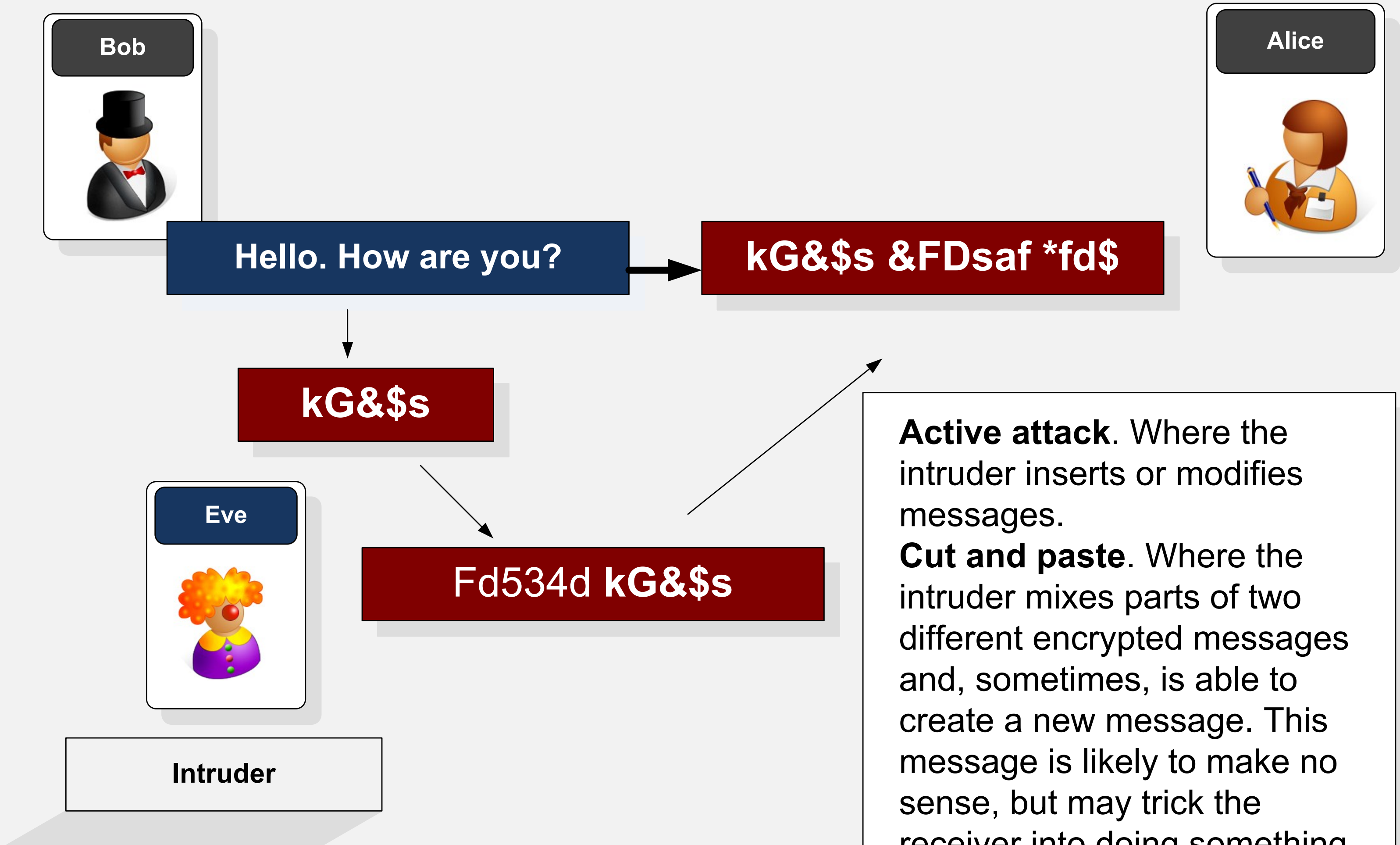

Bob
Hello. How are you?
kG&$s &FDsaf *fd$
Alice
Eve
Intruder
kG&$s &FDsaf *fd$
000...000
Zhk& $31 004fX
000...001
kBb 95&$ $23z
001...100
Hello. How are you?
Exhaustive search. Where the intruder uses brute force to decrypt the ciphertext and tries every possible key.

Man-in-the-middle. Where the intruder is hidden between two parties and impersonates each of them to the other.





The replay system. Where the intruder takes a legitimate message and sends it into the network at some future time.

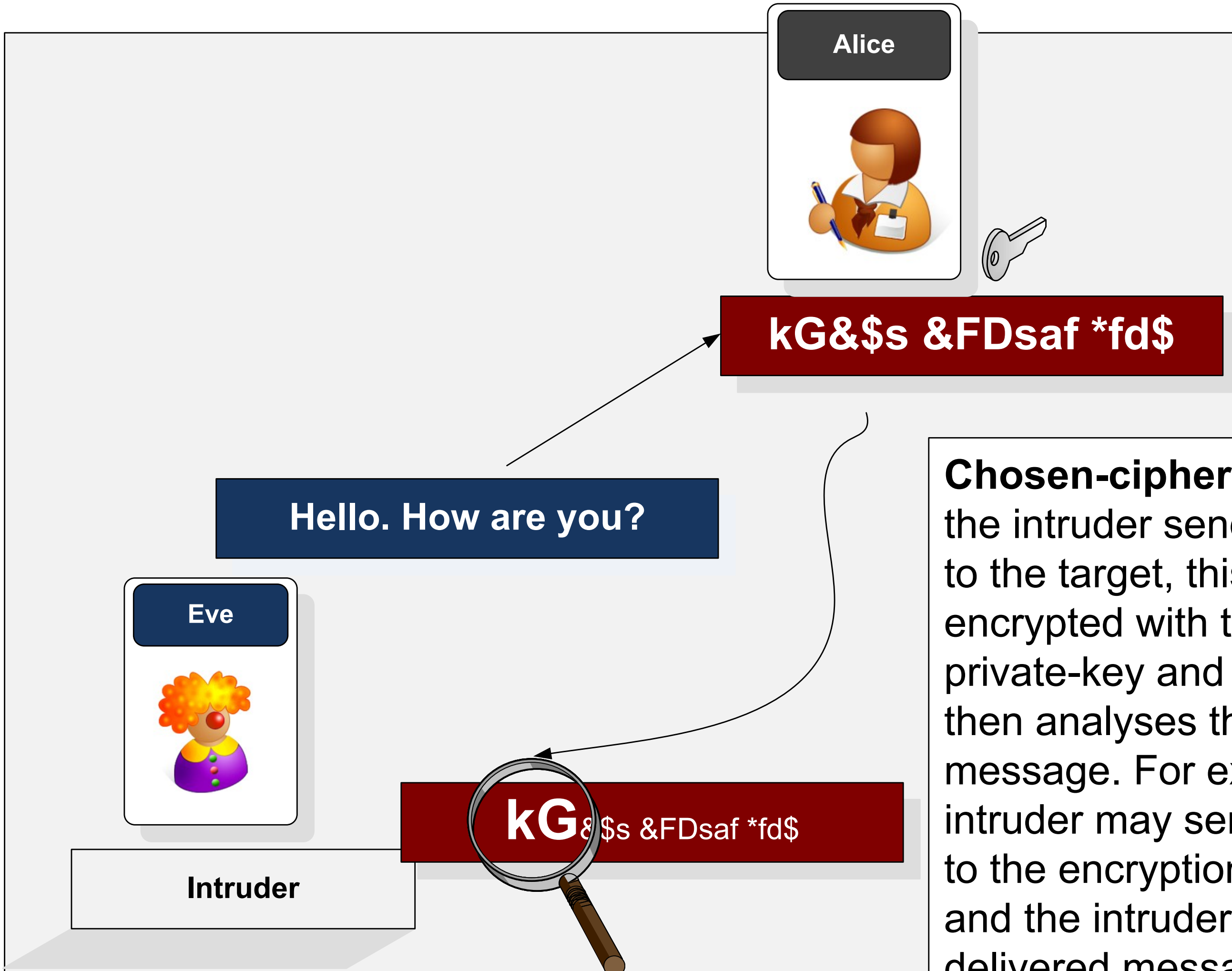


Active attack. Where the intruder inserts or modifies messages.

Cut and paste. Where the intruder mixes parts of two different encrypted messages and, sometimes, is able to create a new message. This message is likely to make no sense, but may trick the receiver into doing something that helps the intruder.

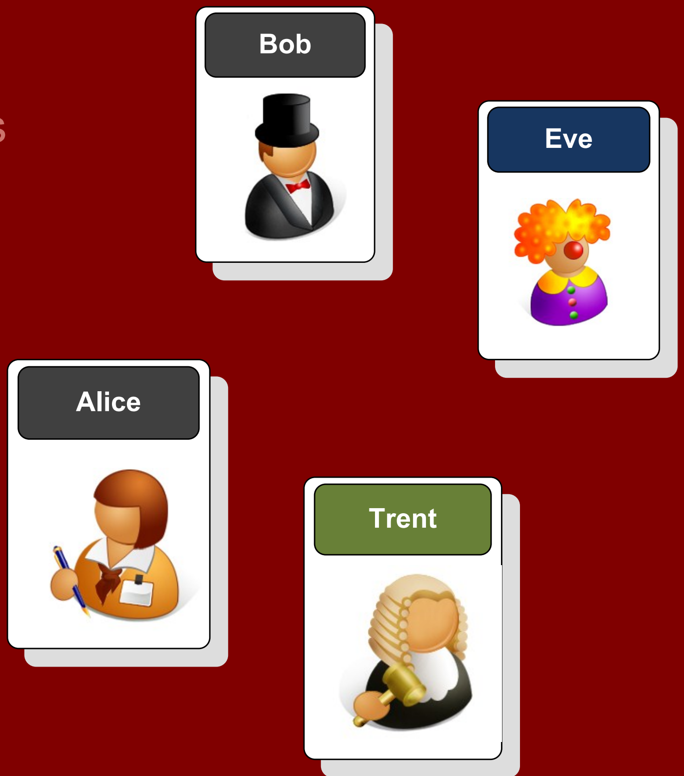
Encryption

Cracking the code

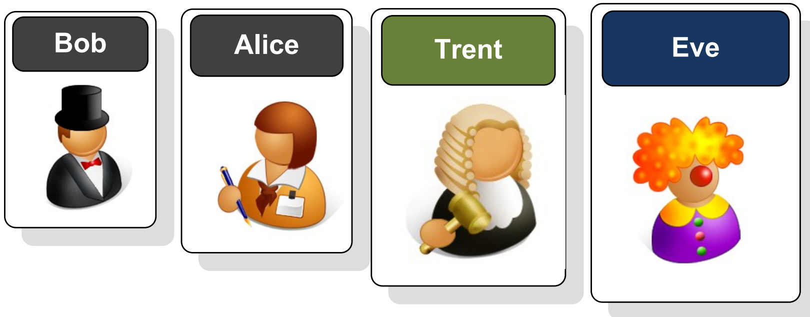


Chosen-ciphertext. Where the intruder sends a message to the target, this is then encrypted with the target's private-key and the intruder then analyses the encrypted message. For example, an intruder may send an e-mail to the encryption file server and the intruder spies on the delivered message.

Encryption



Brute force



Number of keys

The larger the key, the greater the key space.

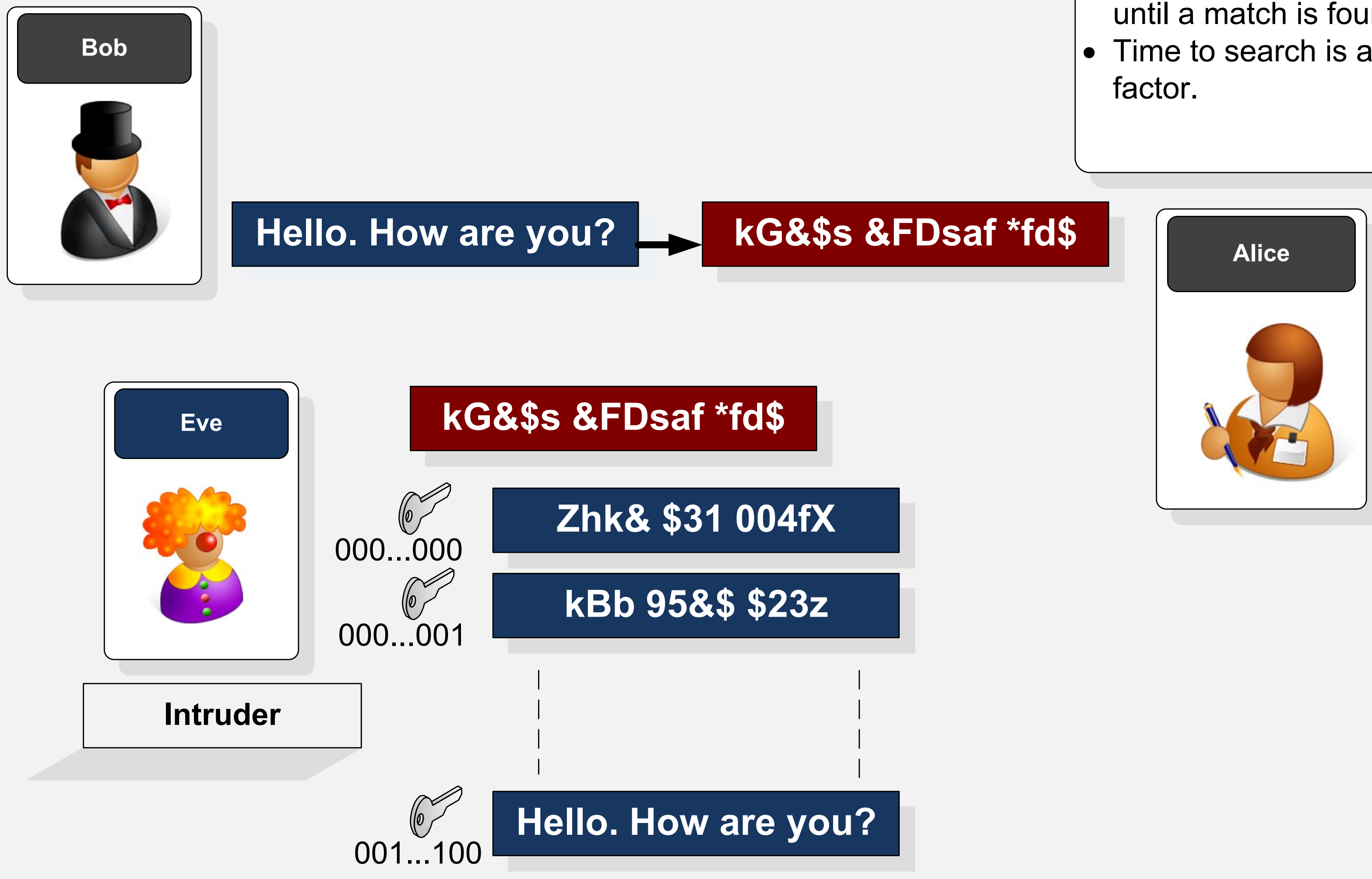
Code size	Number of keys	Code size	Number of keys	Code size	Number of keys
1	2	12	4,096	52	4.5×10^{15}
2	4	16	65,536	56	7.21×10^{16}
3	8	20	1,048,576	60	1.15×10^{18}
4	16	24	16,777,216	64	1.84×10^{19}
5	32	28	2.68×10^{8}	68	2.95×10^{20}
6	64	32	4.29×10^{9}	72	4.72×10^{21}
7	128	36	6.87×10^{10}	76	7.56×10^{22}
8	256	40	1.1×10^{12}	80	1.21×10^{24}
9	512	44	1.76×10^{13}	84	1.93×10^{25}
10	1024	48	2.81×10^{14}	88	3.09×10^{26}

Brute force

Encryption

Brute force

- Eve tries all the keys until a match is found.
- Time to search is a key factor.



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….000000000000000000
To
111111111111….111111111111111111

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

Brute-force

Encryption

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)

Brute-force

Encryption

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.

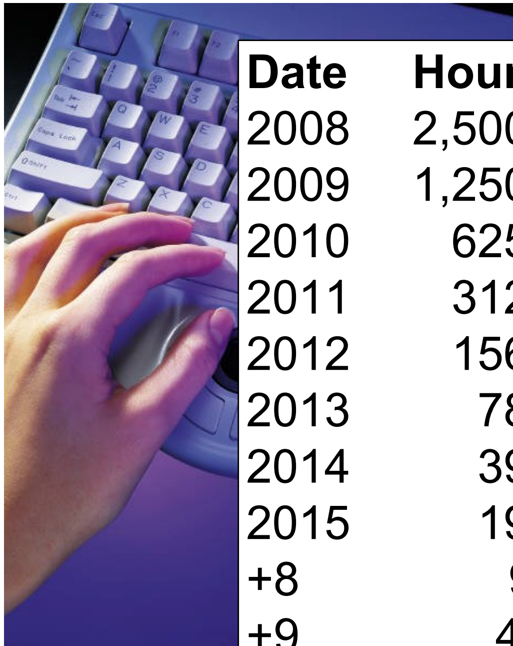
Date	Hours	Days	Years
2008	2,500,000	104,167	285
2009	1,250,000	52,083	143

Brute-force

Encryption

Time to crack

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



Date	Hours	Days	Years
2008	2,500,000	104,167	285
2009	1,250,000	52,083	143
2010	625,000	26,042	71
2011	312,500	13,021	36
2012	156,250	6,510	18
2013	78,125	3,255	9
2014	39,063	1,628	4
2015	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**

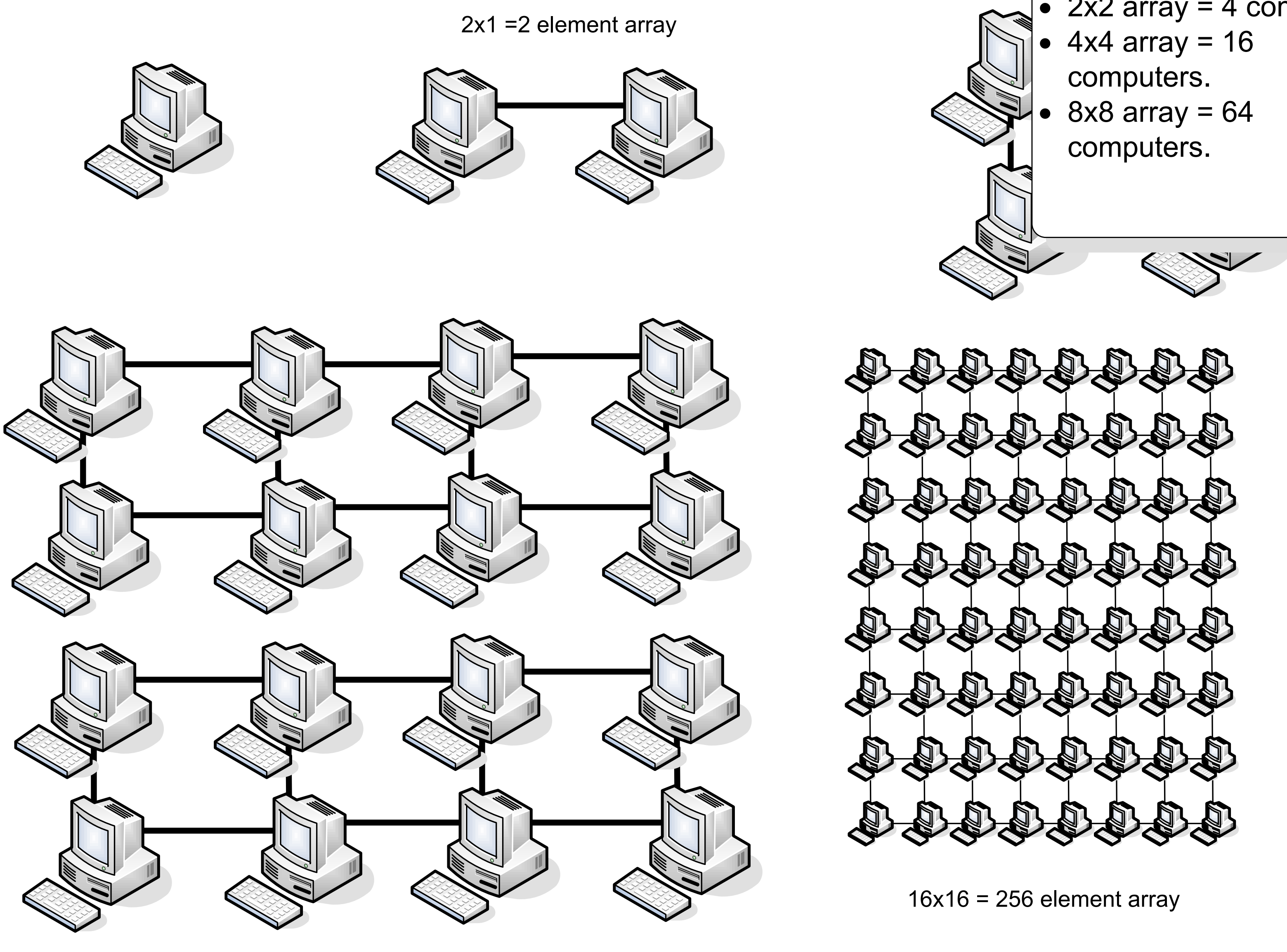
56-bit DES:
Developed
1975
30 years ago!
… now easily
crackable

Brute-force

Encryption

Parallel processing

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

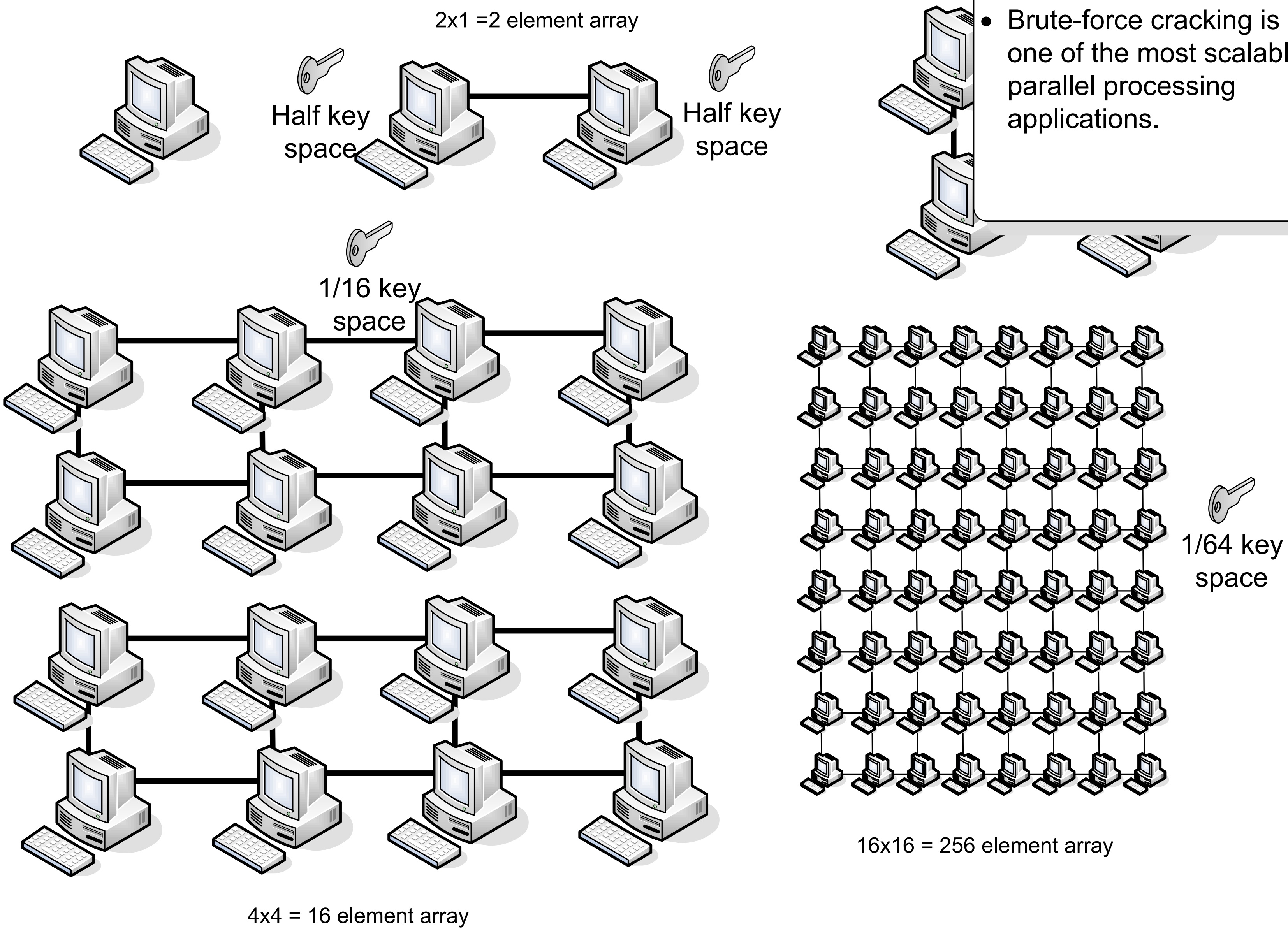


Brute-force

Encryption

Parallel processing

- Brute-force cracking is one of the most scalable parallel processing applications.

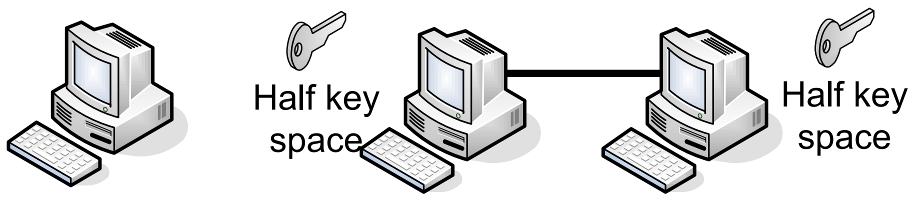


Brute-force

Encryption

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

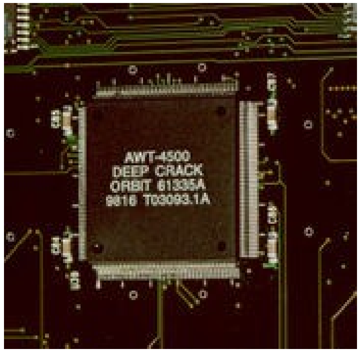
4x4 = 16 element array

Brute-force

Encryption

56-bit DES cracker

- 56-bit DES is seen as insecure as it can be cracked by enhanced processors.

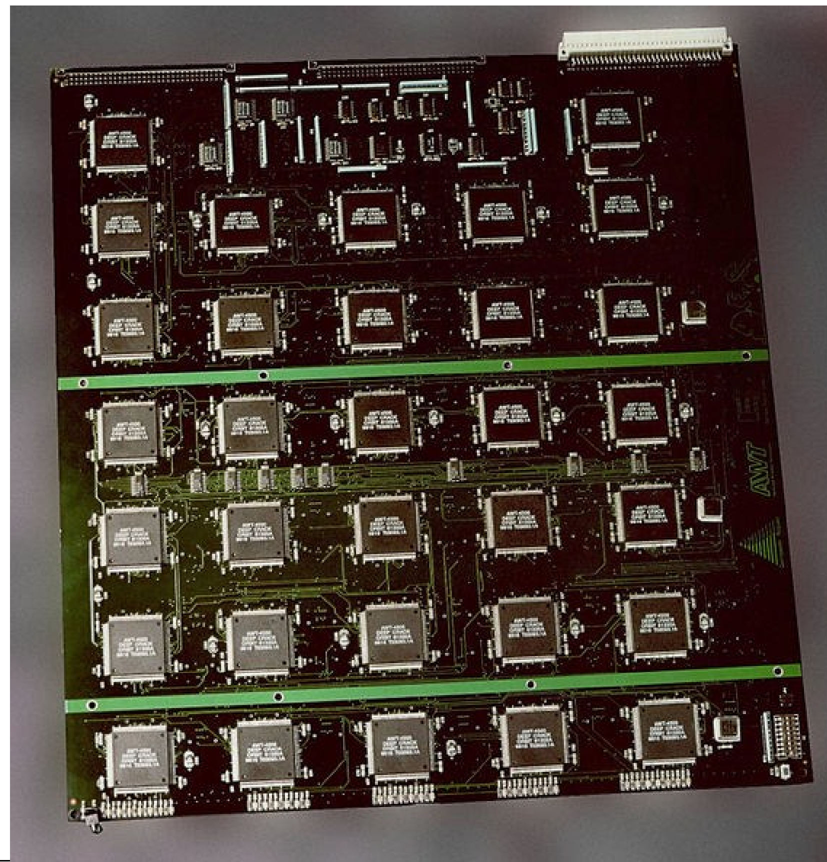


Year: 1998

Electronic Frontier Foundation - Cyberspace Civil Rights Group
90,000,000 keys per seconds

Array: 29 circuits of 64 chips
= 1856 elements

2.5 days

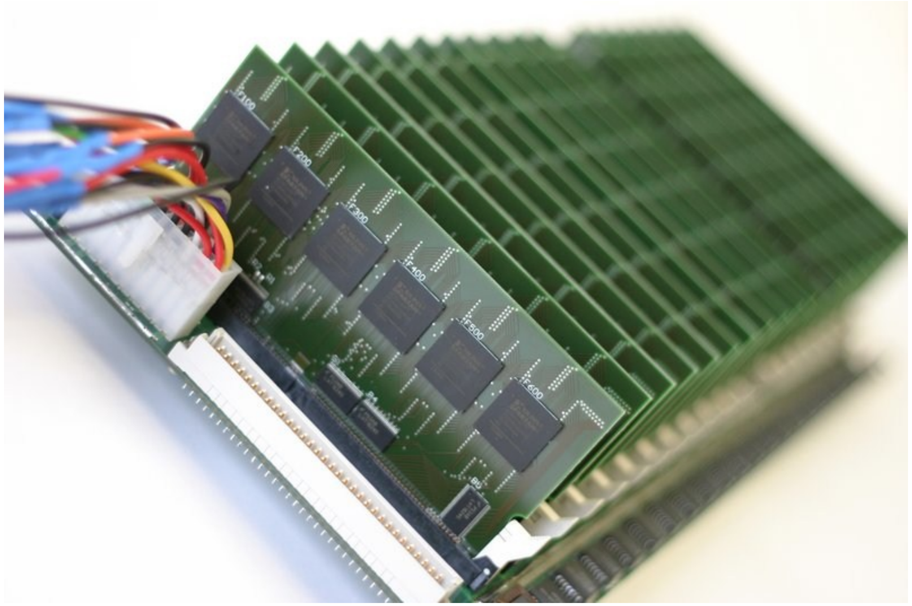


COPACOBANA

- Cracks 64-bit DES in less than nine days for less than $10,000

Now

System: **COPACOBANA** (Cost-Optimized Parallel COde Breaker)
Time to crack: Less than 9 days for DES (64-bit code).
Cost: Less than $10,000



RSA Lab Challenge

- RSA Labs have a number of challenges, each of which have been solved. The present challenge is 72-bit RC5.

1997. RSA Lab's 56-bit RC5 Encryption Challenge - 250 days and 47% of the key space tested) – **distributed.net**

1998. RSA Lab's 56-bit DES II-1 Encryption Challenge - 39 days.
1998. RSA Lab's 56-bit DES II-2 Encryption Challenge - 2.5 days.

1999. RSA Lab's 56-bit DES-III Encryption Challenge - after 22.5 hours using EFF's Deep Crack custom DES cracker.

2002. RSA Lab's 64-bit RC5 Encryption Challenge — Completed 14 July 2002 – 1,757 days and 83% of the key space tested.

RSA Lab's 72-bit RC5 Encryption Challenge - In progress.

Brute-force

Encryption

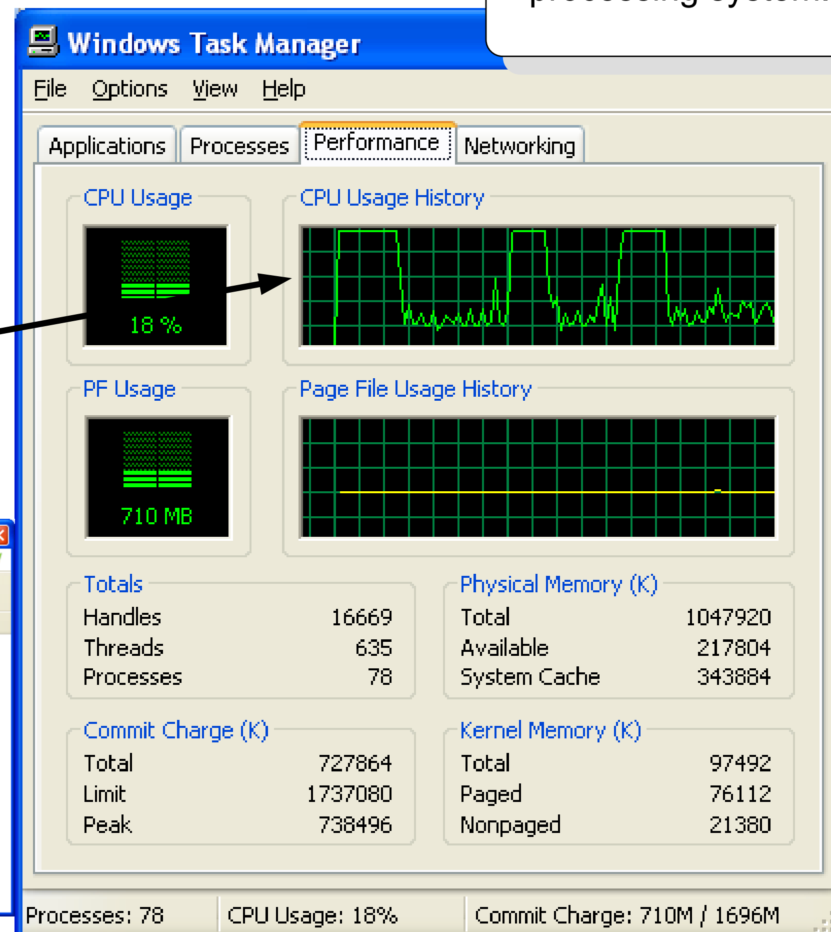
Brute-force

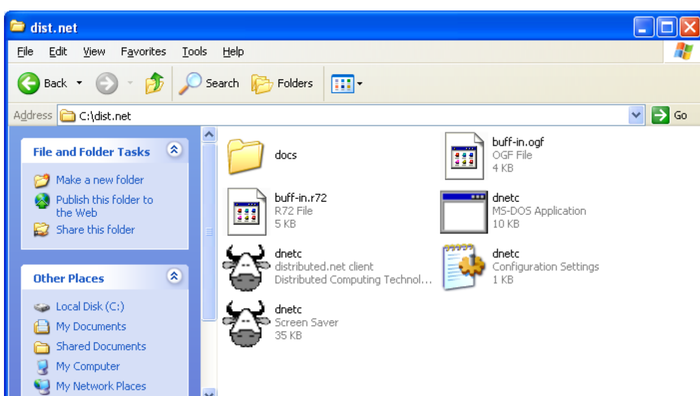
Encryption

distributed.net

- Tries to crack RSA Lab challenge by processing a range of possible keys while the screen save is on.
- Massive parallel processing system.

Distributed.net is starting and stopping (Max CPU when searching for possible keys)





Super Computers

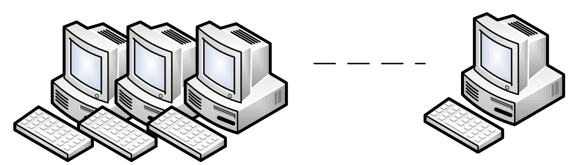
- BlueGene is 1.8million times more powerful than a standard PC.

BlueGene/L – eServer Blue Gene Solution
DOE/NNSA/LLNL, IBM
Department of Energy's (DOE) National Nuclear Security Administration's (NNSA).
131,072 processors
367,000 Gigaflop= 367,000,000 Mflops

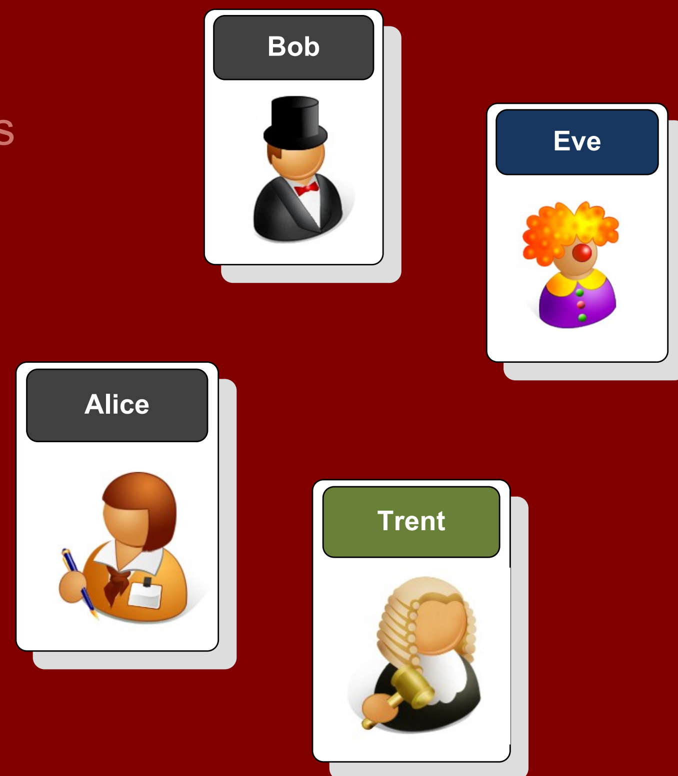
Red Storm - Sandia/ Cray Red Storm
NNSA/Sandia National Laboratory United States, Opteron 2.4 GHz dual core Cray Inc.

26,544 processors
127,000 Gflops

Typical PC: 200 Mflop … BlueGene is **1,835,000** times more powerful than a desktop.



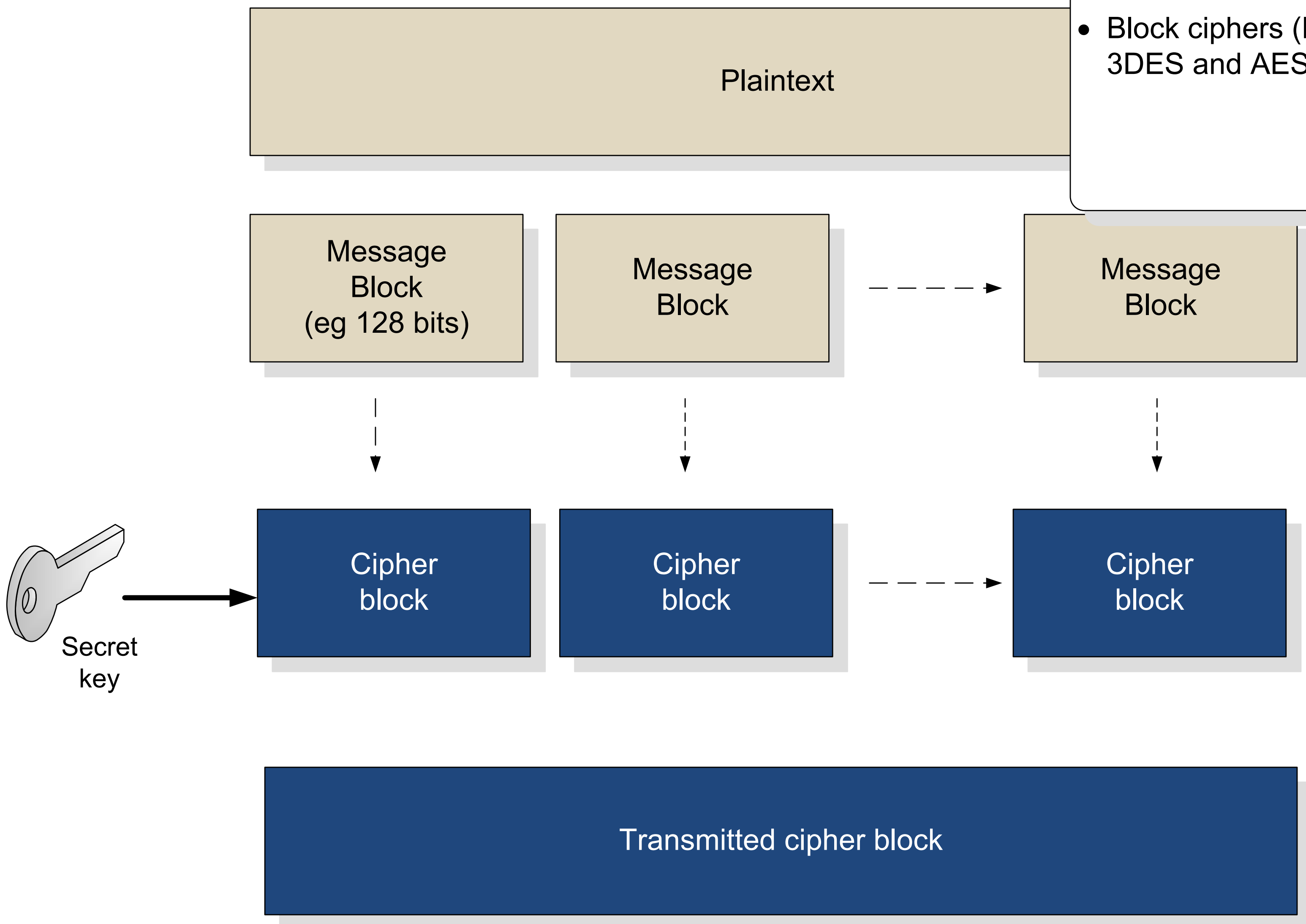
Encryption



Block or stream

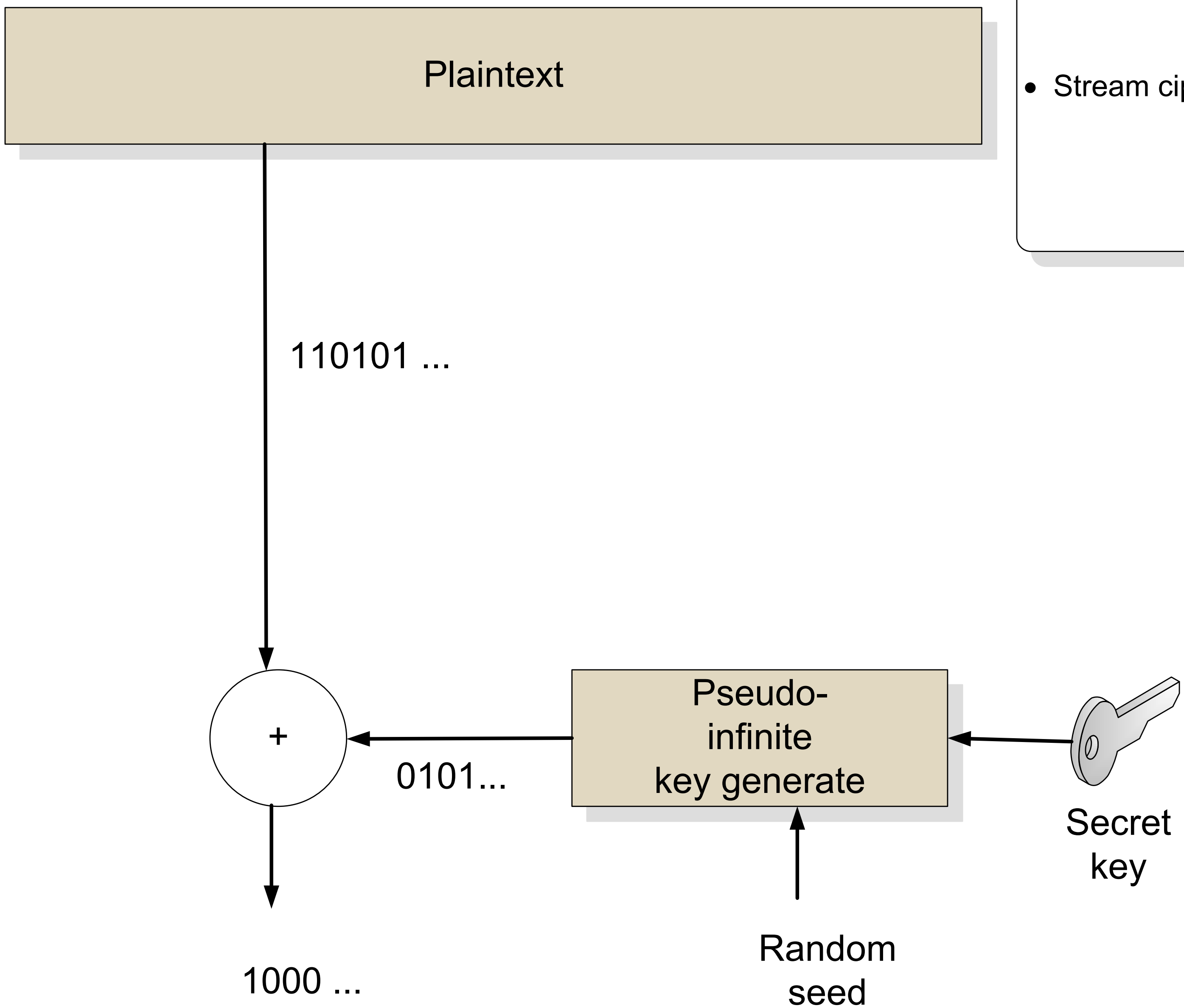
Block cipher

- Block ciphers (DES, 3DES and AES)



Stream cipher

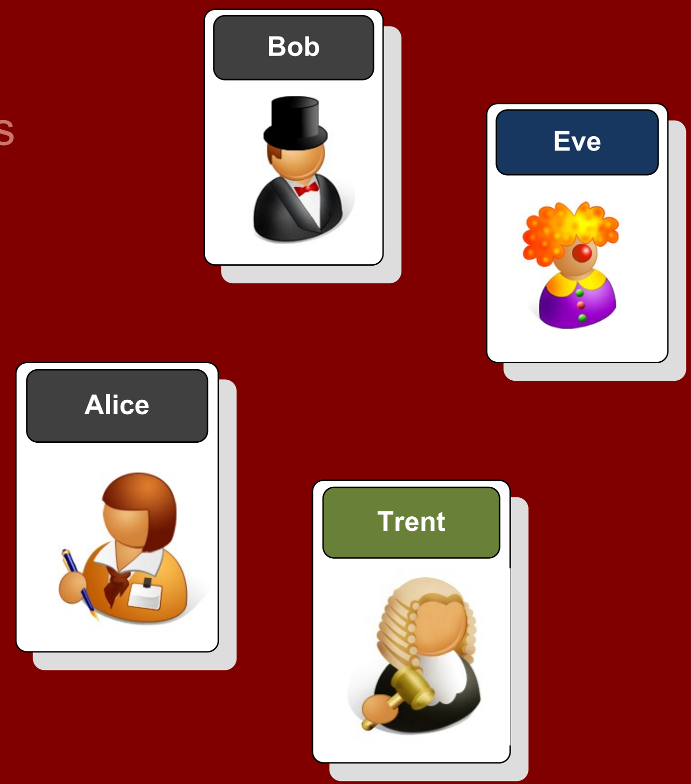
- Stream cipher (RC4)



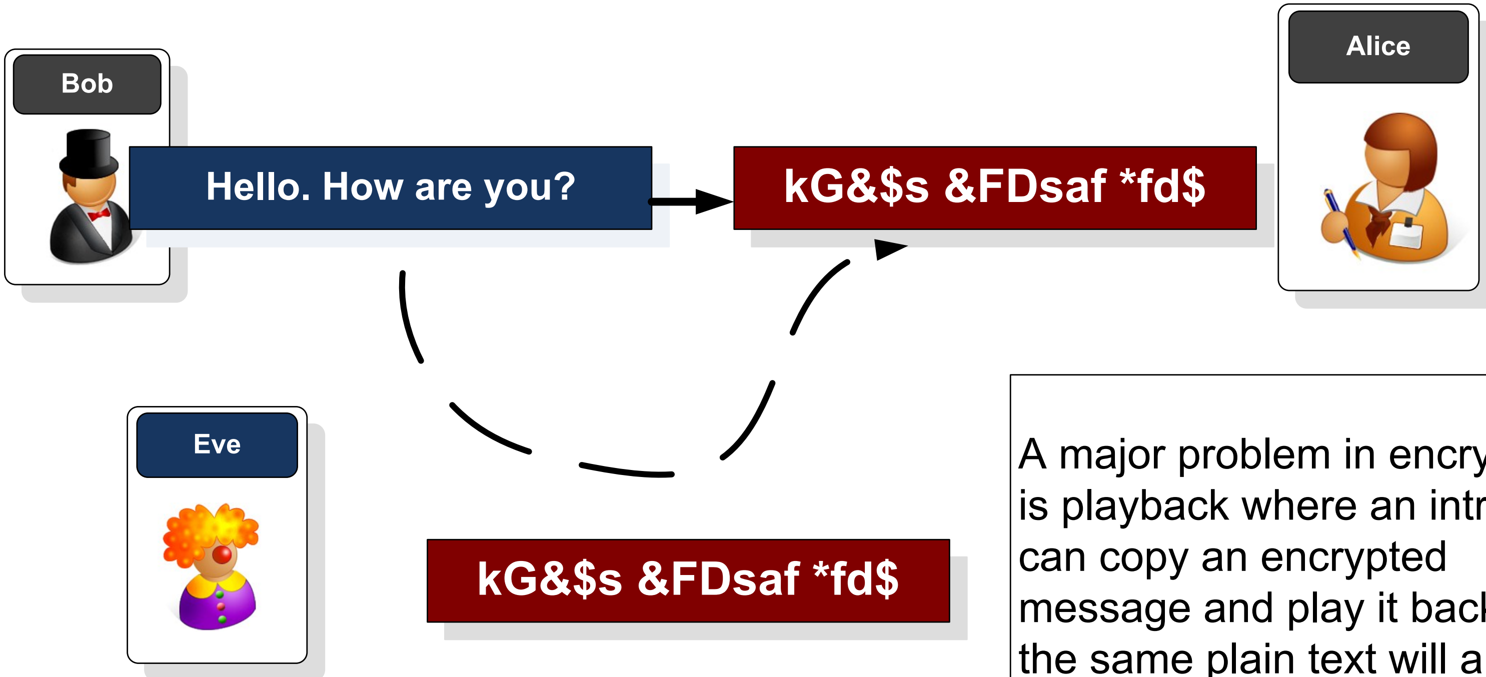
Stream or block?

Encryption

Encryption

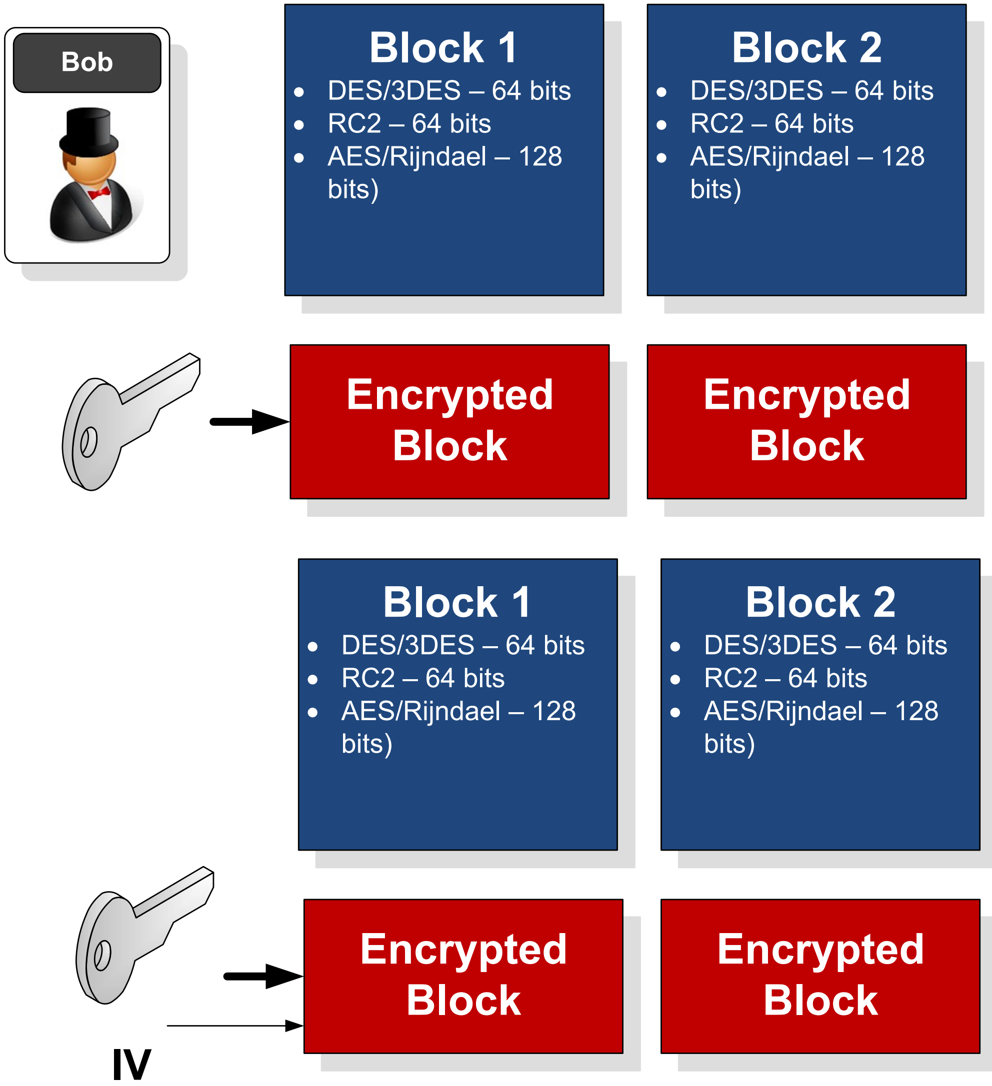


Private-key Methods



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.

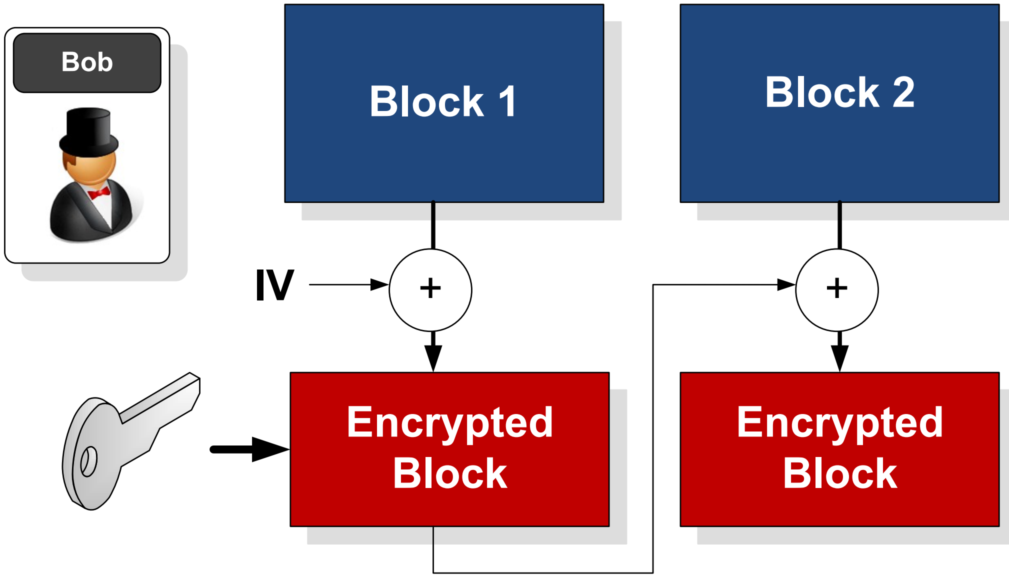
The solution is to add **salt** to the encryption key, as that it changes its operation from block-to-block (for block encryption) or data frame-to-data frame (for stream encryption)



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

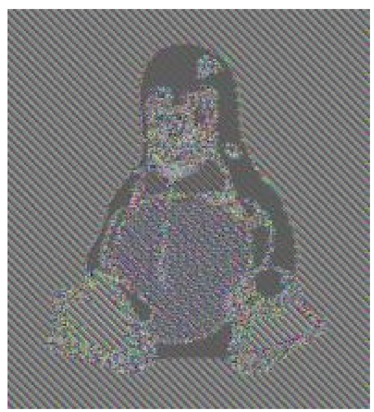


Image with AES using ECB

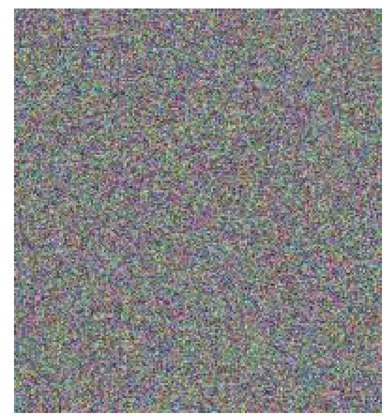


Image with AES using CBC

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

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:

$$\text{Encrypt}_{K3}(\ \text{Decrypt}_{K2}(\ \text{Encrypt}_{K1}(\text{message})))$$

http://buchananweb.co.uk/security07.aspx

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

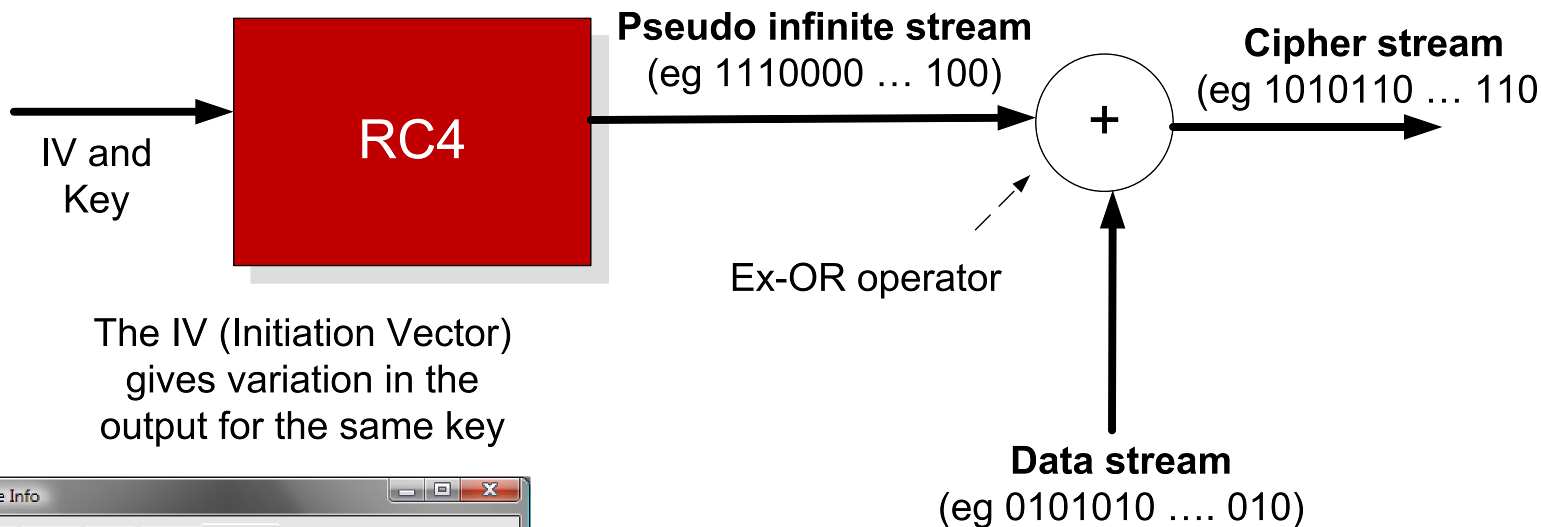
RC-2. RC2 ("Rivest Cipher") is seen as a replacement for DES. It was created by Ron Rivest in 1987, and is a **64-bit block code** and can have a key size from 40 bits to 128-bits (in increments of 8 bits). The 40-bit key version is seen as weak, as the encryption key is so small, but is favoured by 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://buchananweb.co.uk/security06.aspx

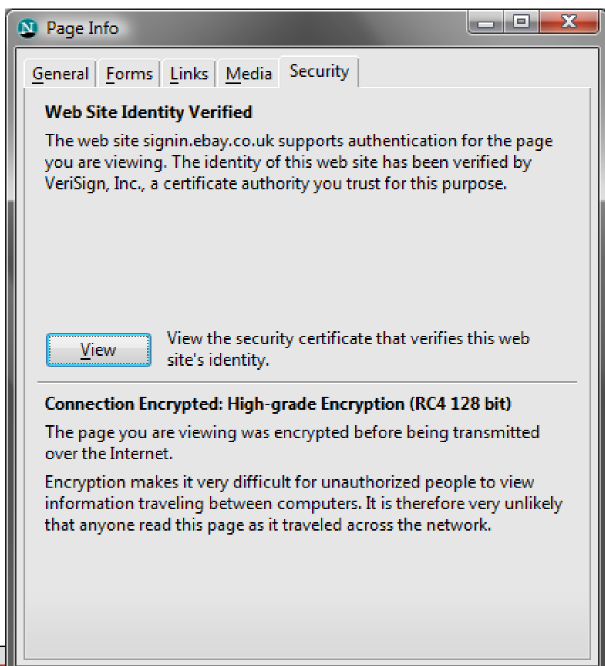
AES/Rijndael. AES (or Rijndael) is the new replacement for DES, and uses **128-bit blocks** with 128, 192 and 256 bit encryption keys. It was selected by NIST in 2001 (after a five year standardisation process). The name Rijndael comes from its Belgium creators: Joan Daemen and Vincent Rijmen.

http://buchananweb.co.uk/security15.aspx

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



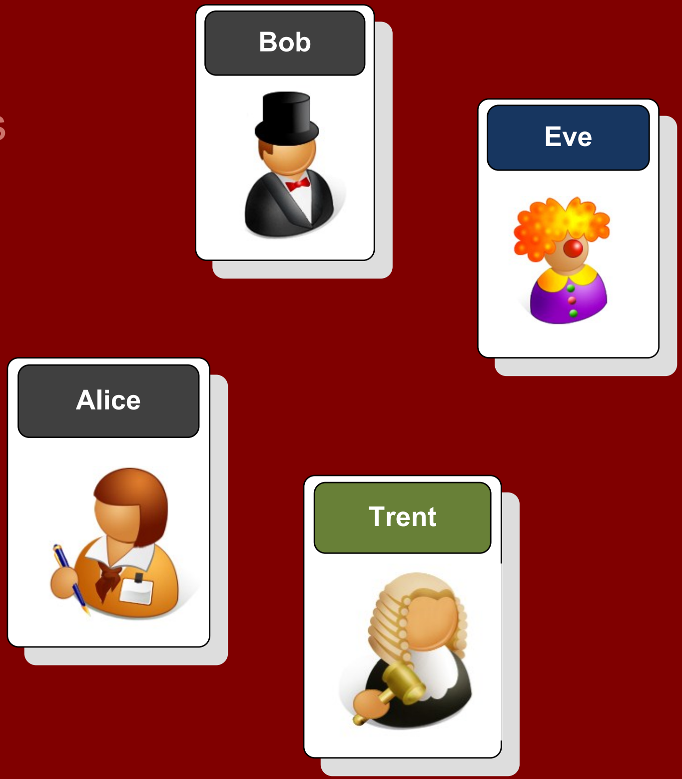
The IV (Initiation Vector) gives variation in the output for the same key



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

Encryption

Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption



Encryption keys

Key enthropy

- 256 phrases -> 8 bit equivalent key.
- 1024 phases -> 10 bit equivalent key.
- 1,048,576 phrases -> 20 equivalent key.

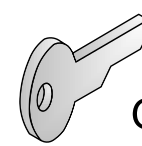
Key entropy: Relates to the equivalent number of bits given the range of phases used.

For example: if there were eight pass phrases – this would be equivalent to a 3-bit key.

Standard English gives 1.3 bits per character. Thus an **8 character word** gives **10.4 bits** for the key entropy.

Key generating method, such as a pass phrase

Key generator

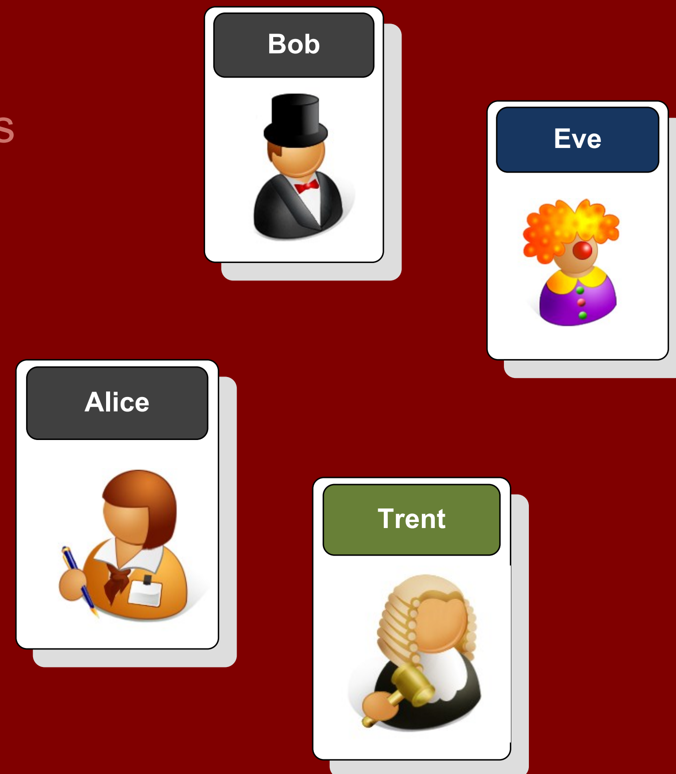


Generate key

Pass phrases might be: Napier, napier, napier1, napier11, napier123, and so on (the range of key will obviously be limited if the number of phrases are limited)

Encryption

Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption

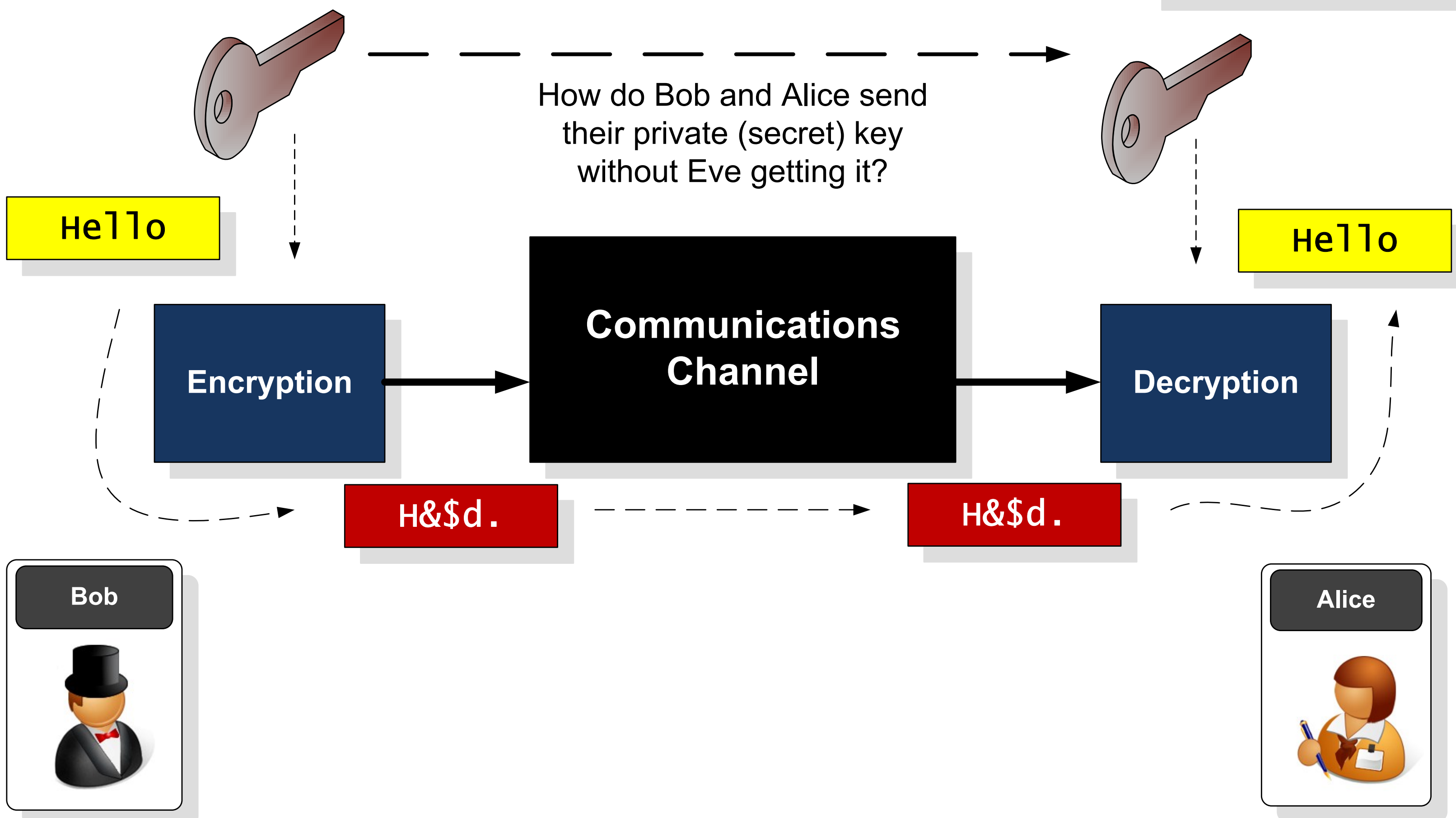


Passing keys

Encryption

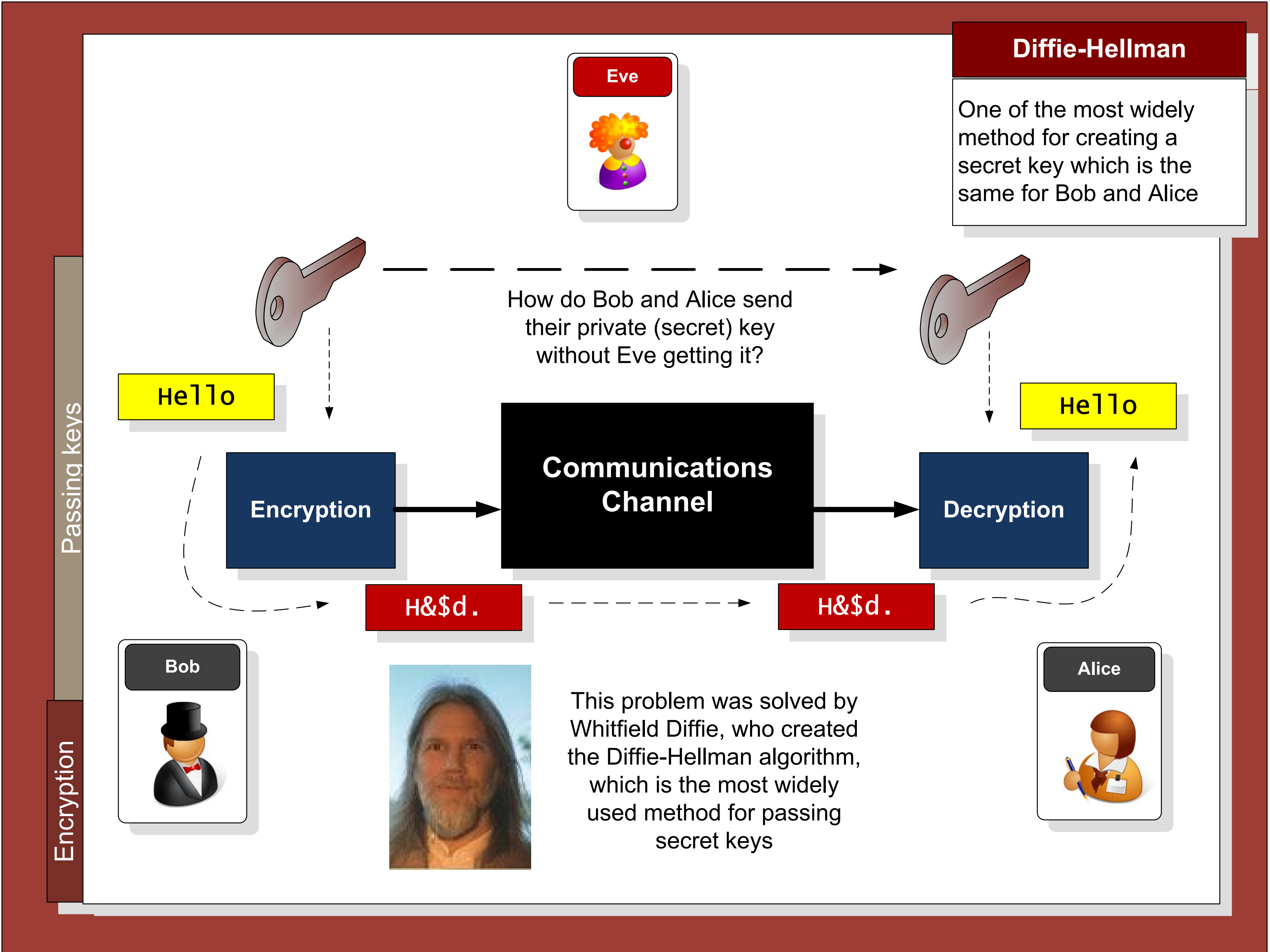
Private key

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



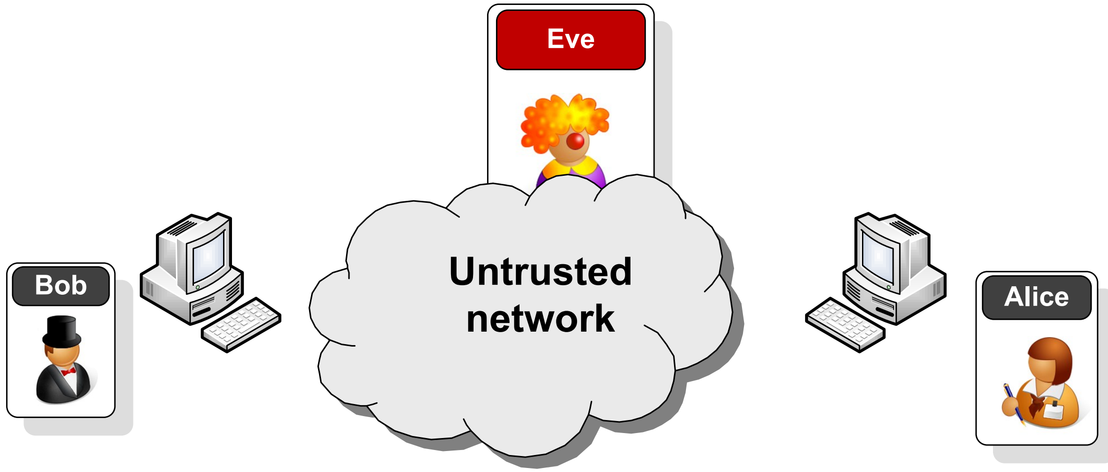
Passing keys

Encryption


Diffie-Hellman
One of the most widely method for creating a secret key which is the same for Bob and Alice
Eve
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
This problem was solved by Whitfield Diffie, who created the Diffie-Hellman algorithm, which is the most widely used method for passing secret keys
Passing keys
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. $B = G^y \bmod n$

4. A and B values exchanged

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

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

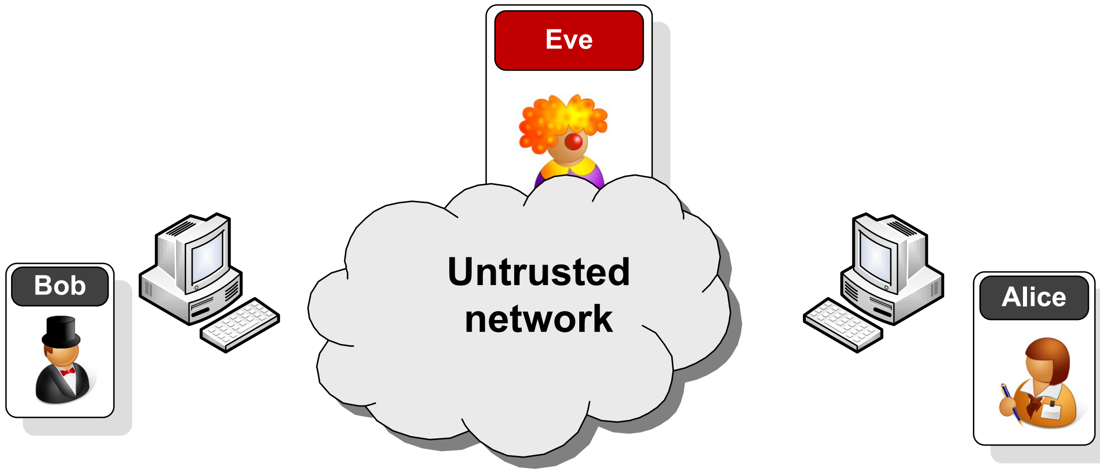
K1 and *K2* should be the **same** and are the secret key

Passing keys

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 (5 and 4)

2. Generate a random value (*3*)

2. Generate a random value (4)

3. $A = 5^3 \bmod 4 = 5$

3. $B = 5^4 \bmod 4 = 1$

4. A and B values exchanged

5. $K1 = 1^5 \bmod 4 = 1$

5. $K2 = 5^4 \bmod 4 = 1$

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

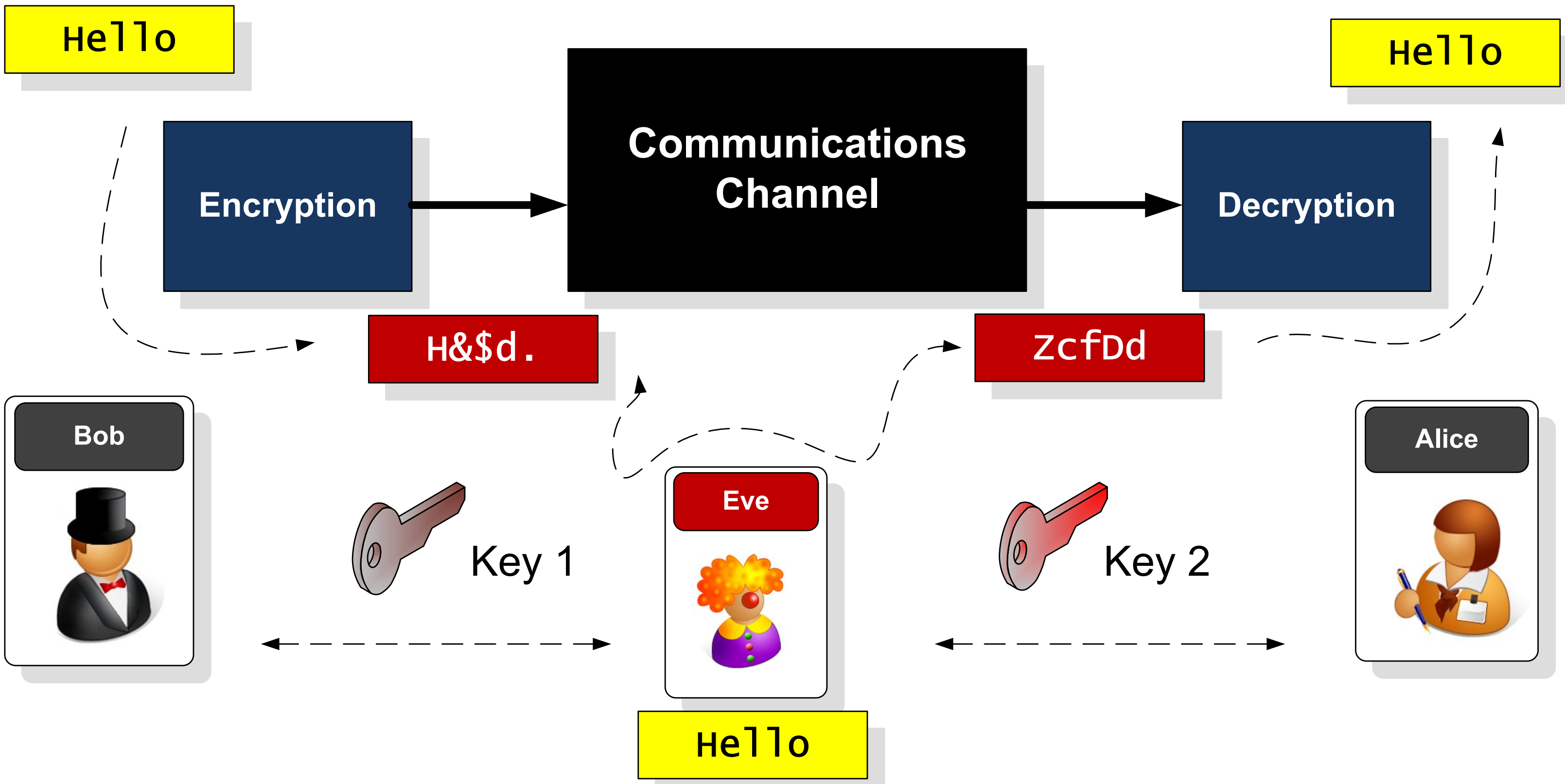
Passing keys

Encryption

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.

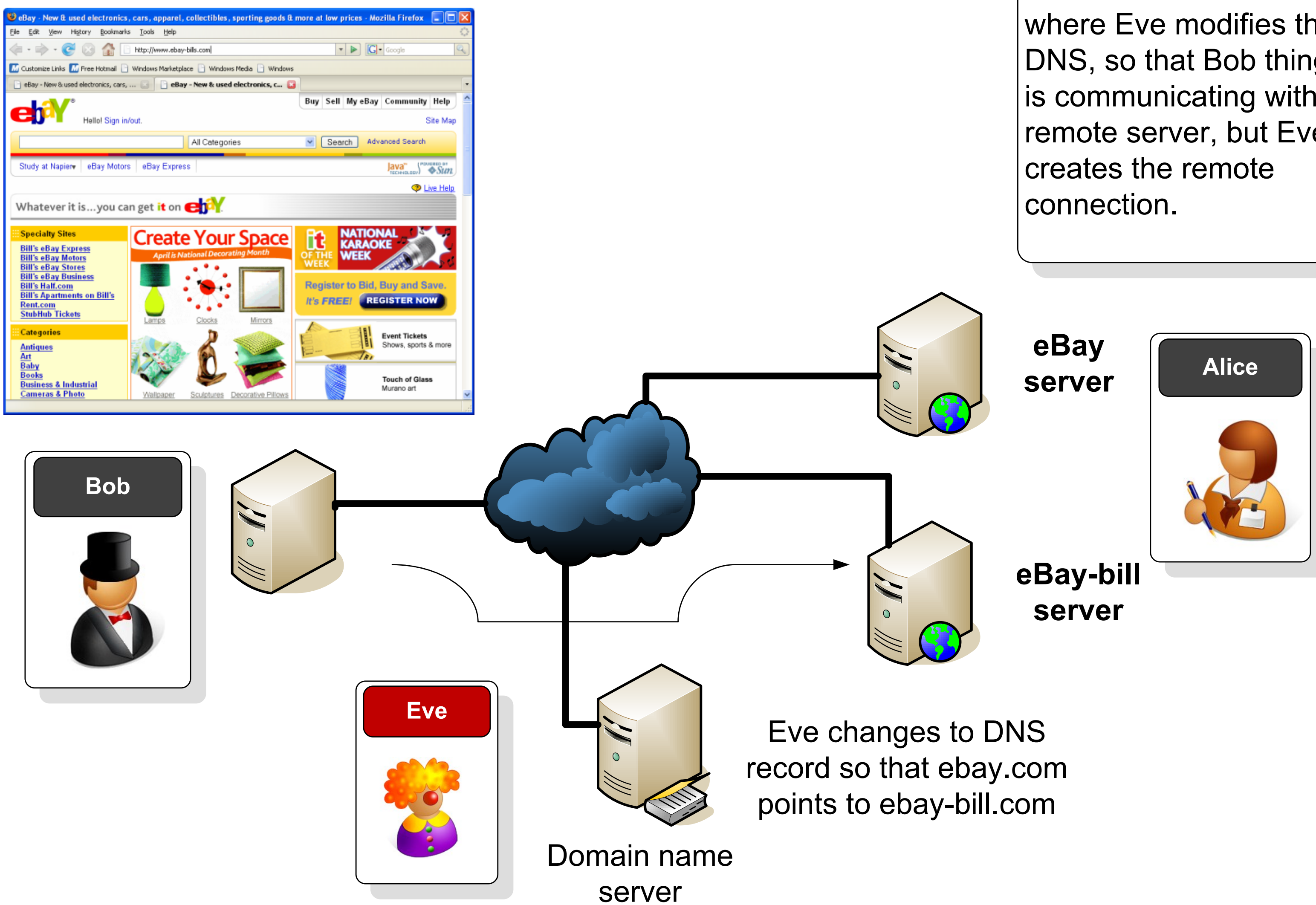


Passing keys

Encryption

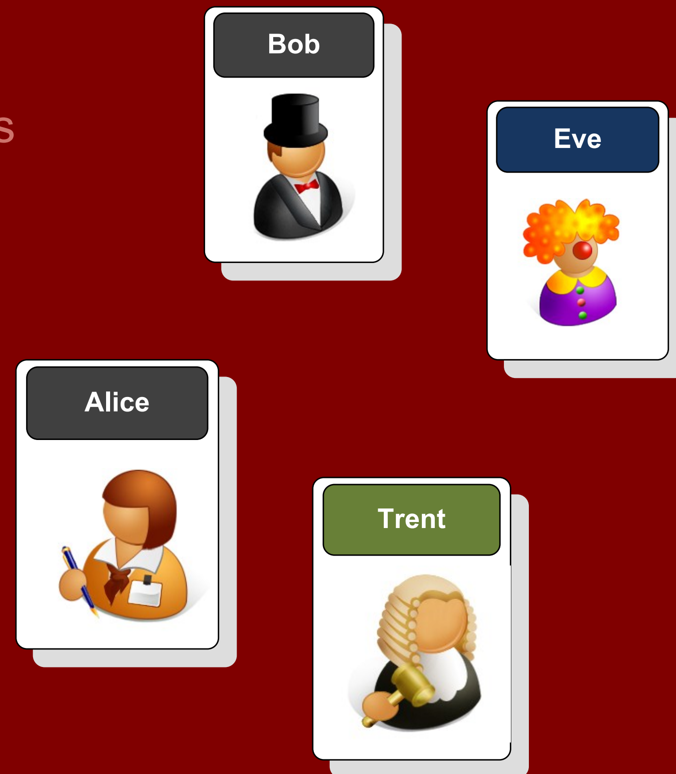
DNS poisoning

A man-in-the-middle is where Eve modifies the DNS, so that Bob things he is communicating with the remote server, but Eve creates the remote connection.



Encryption

Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption



Public-key encryption

Encryption

Public-key encryption

Public-key

RSA is still one of the most widely used encryption algorithms, and still stands up for secure communication, but is relatively slow in encrypting and decrypting.

Eve

With Diffie-Hellman we need the other side to be active before we send data. Can we generate a special one-way function which allows is to distribute an encryption key, while we have the decryption key?

Bob

Alice

Encryption/ Decryption

Communications Channel

Encryption/ Decryption

Solved in 1977, By Ron Rivest, Adi Shamir, and Len Aldeman created the RSA algorithm for public-key encryption.

Bob

Select two prime numbers: **a** and **b**

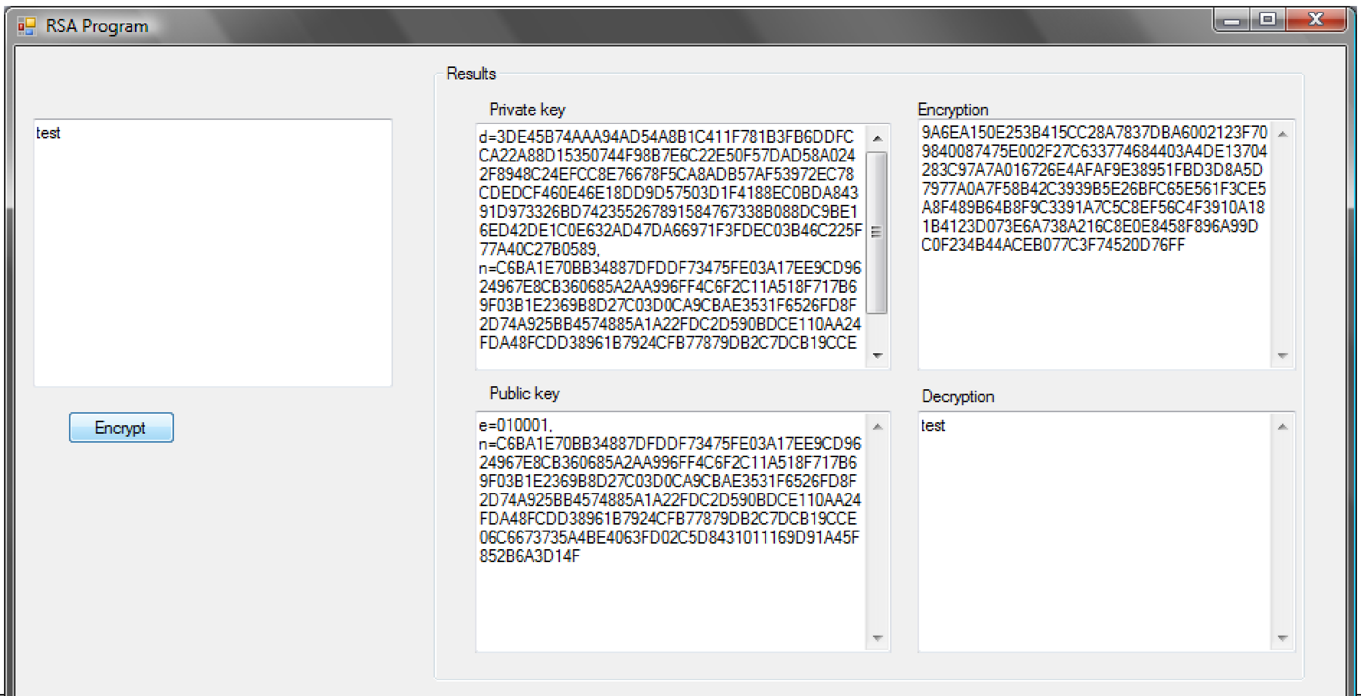
n = a x b

e is chosen so that **e** and **(a-1)x(b-1)** are relatively prime (no common factor greater than 1)

Public key is now: **<e,n>**

d = e-1 mod [(a-1)x(b-1)]

Private key is now: **<d,n>**



Public-key encryption

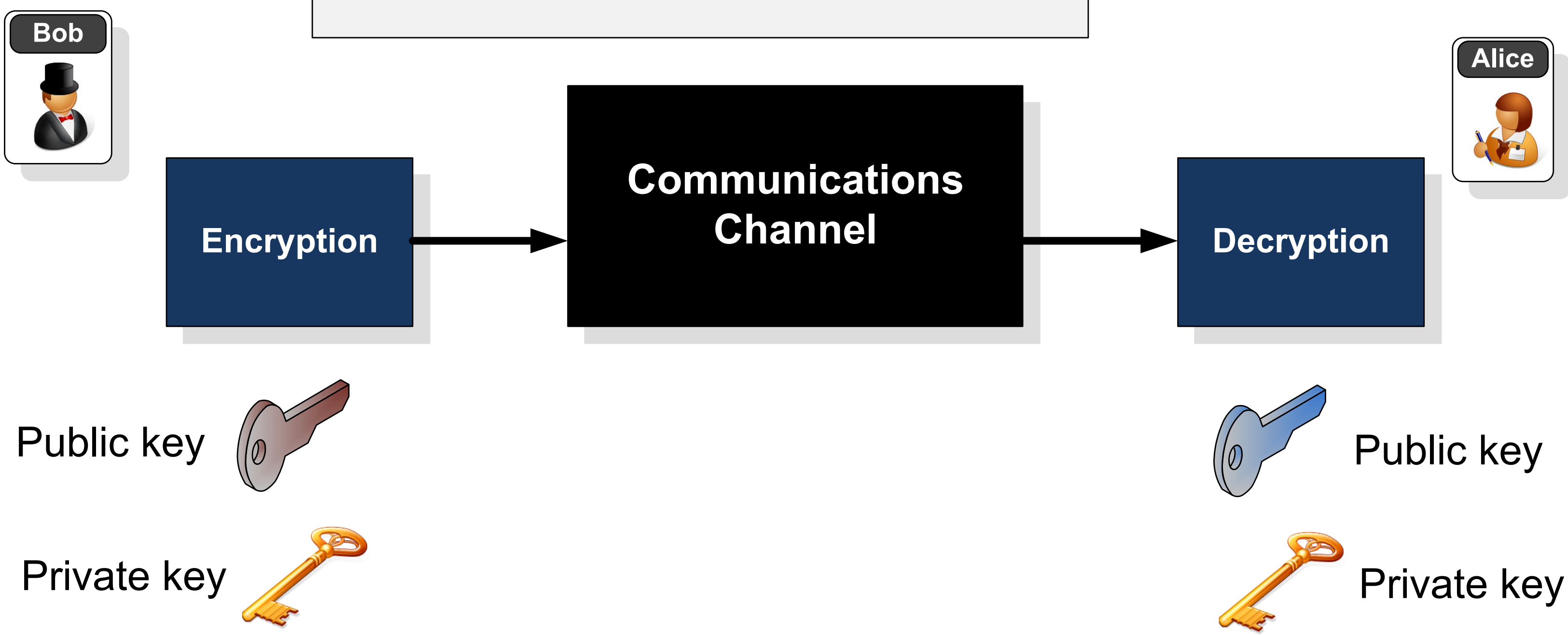
Encryption

Author: Prof Bill Buchanan

Public-key

Public key are keys which relate to extremely large prime numbers (as it is difficult to factorise large prime numbers). It is extremely difficult to determine a private key from a public key.

Public key generates two keys: A public key and a private one. These are special in that if one is applied to encrypt, the other can be used to decrypt



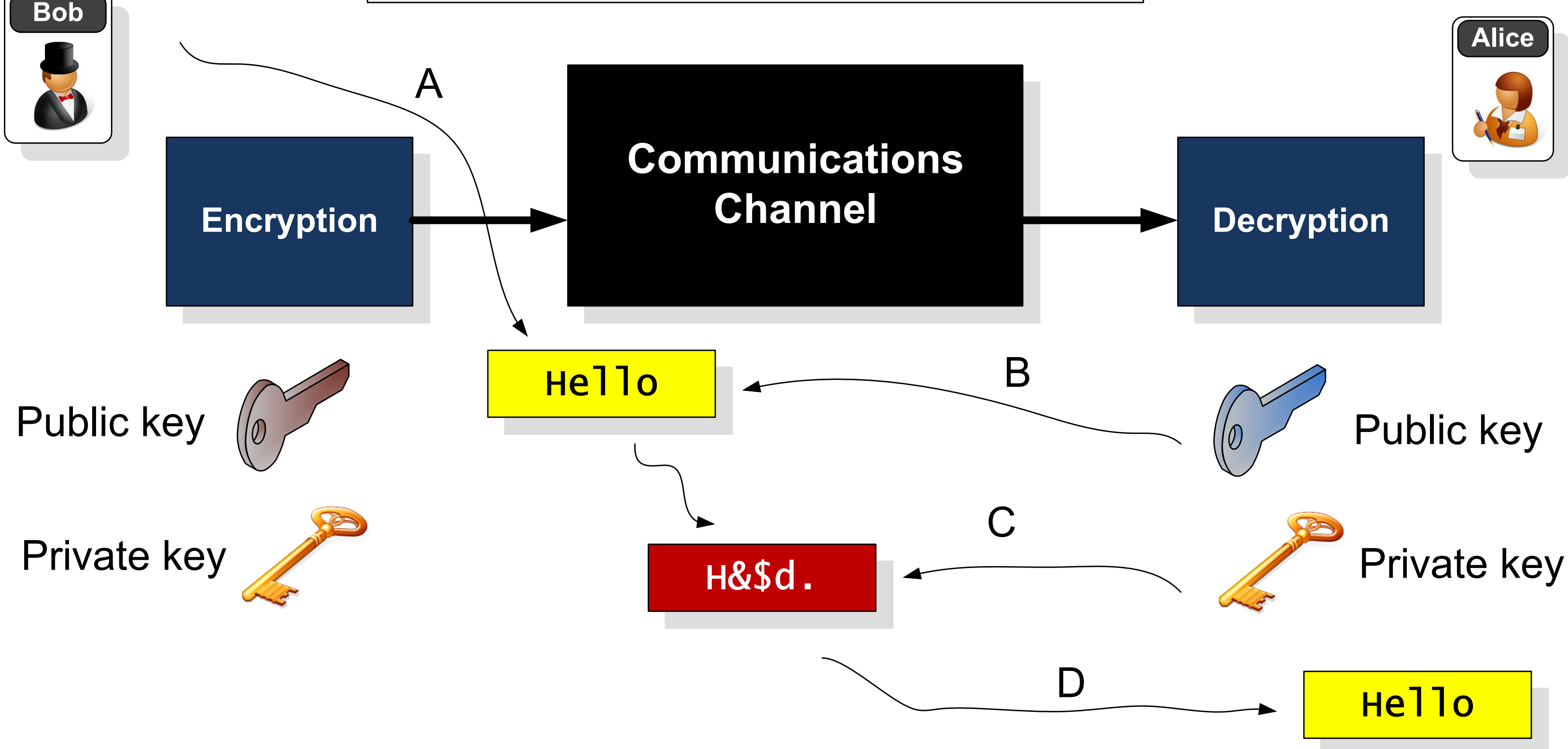
Public-key encryption

Encryption

Public-key

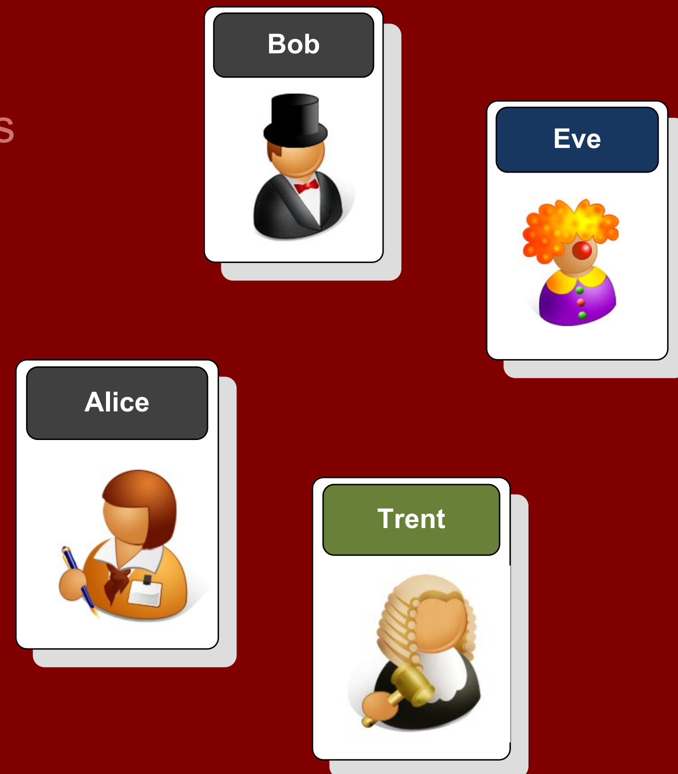
- Once Bob encrypts the message, the only key which can decrypt it is Alice's private key.
- Bob and Alice keep their private keys secret.

A. Bob creates the message.
B. Bob encrypts with Alice's public key and sends Alice the encrypted message
C. Alice decrypts with her private key
D. Alice receives the message

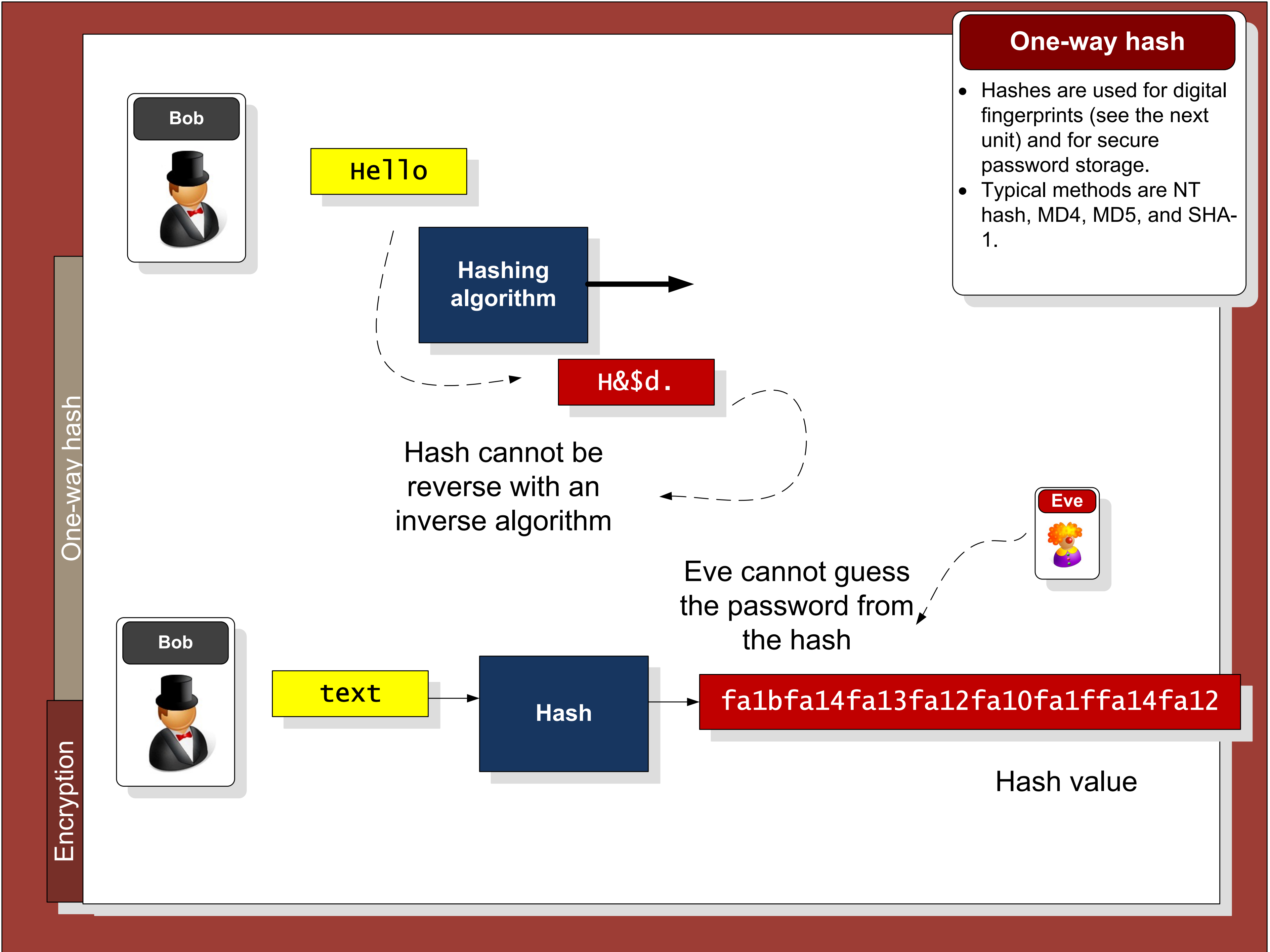


Encryption

Introduction
Before electronic communications
Codes
A few fundamentals
Key-based encryption
Cracking the code
Brute force
Block or stream
Private-key methods
Encryption keys
Passing keys
Public-key encryption
One-way hash
Encrypting disks
PGP encryption



One-way hash

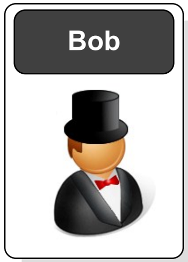

One-way hash
Hashes are used for digital fingerprints (see the next unit) and for secure password storage.
Typical methods are NT hash, MD4, MD5, and SHA-1.
Bob
Hello
Hashing algorithm
H&$d.
Hash cannot be reverse with an inverse algorithm
Eve
Eve cannot guess the password from the hash
Bob
text
Hash
fa1bfa14fa13fa12fa10fa1ffa14fa12
Hash value
One-way hash
Encryption

One-way hash

- Hashes are used for digital fingerprints (see the next unit) and for secure password storage.
- Typical methods are NT hash, MD4, MD5, and SHA-1.

One-way hash

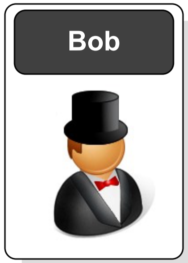
Windows login/ authentication



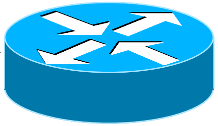
mypass → NT hash (MD4) → fa1bfa14fa13fa12fa10fa1ffa14fa12

NT-password hash for Windows NT, XP and Vista

Cisco password storage (MD5)



mypass →



```
# config t
(config)# enable secret test

Current configuration : 542 bytes
!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
enable secret 5 $1$/Nwk$knsEQYxZVenGjWOGj/TGk0
```

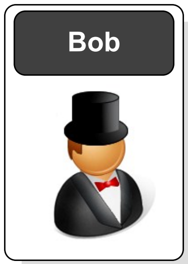
MD5 encoded password →

Encryption

One-way hash

Encryption

Windows login/ authentication



mypass → NT hash (MD4) → fa1bfa14fa13fa12fa10fa1ffa14fa12

NT-password hash for Windows NT, XP and Vista

Hashing suffers from **dictionary attacks** where the signatures of well know words are stored in a table, and the intruders does a lookup on this

mypast	effahd13fa12fa10fgffa1ffa14fa144
mypass	fa1bfa14fa13fa12fa10fa1ffa14fa12
mypose	ff12189043210954defff0123444512d
test1	aabbfce023215546dfeddd0101001cd

One-way hash

- Hashing suffers from dictionary attacks, thus it is important that any passwords are not standard words, such as to change **password** for **pA55wOrd**.

A major factor with hash signatures is:

- **Collision.** This is where another match is found, no matter the similarity of the original message. This can be defined as a **Collision attack**.
- **Similar context**. This is where part of the message has some significance to the original, and generates the same hash signature. The can be defined as a Pre-image attack.
- **Full context**. This is where an alternative message is created with the same hash signature, and has a direct relation to the original message. This is an extension to a Pre-image attack.

In 2006 it was shown that MD5 can produce collision within less than a minute.

A 50% probability of a collision is:

$$\sqrt{N(signatures)} = \sqrt{2^n} = 2^{\frac{n}{2}}$$

where n is the number of bits in the signature. For example, for MD5 (128-bit) the number of operations that would be required for a better-than-50% chance of a collision is:

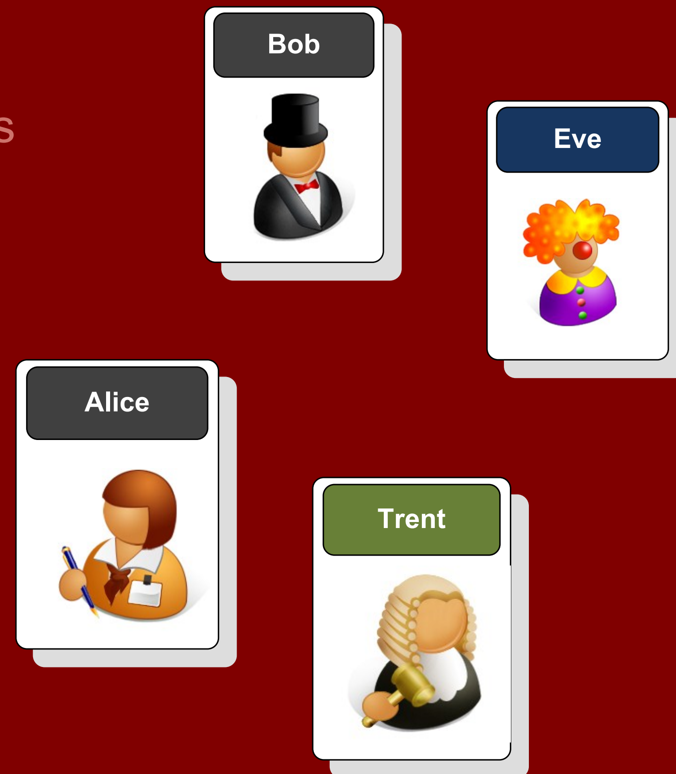
$$2^{64}$$

Note, in 2006, for SHA-1 the best time has been 18 hours

One-way hash

Encryption

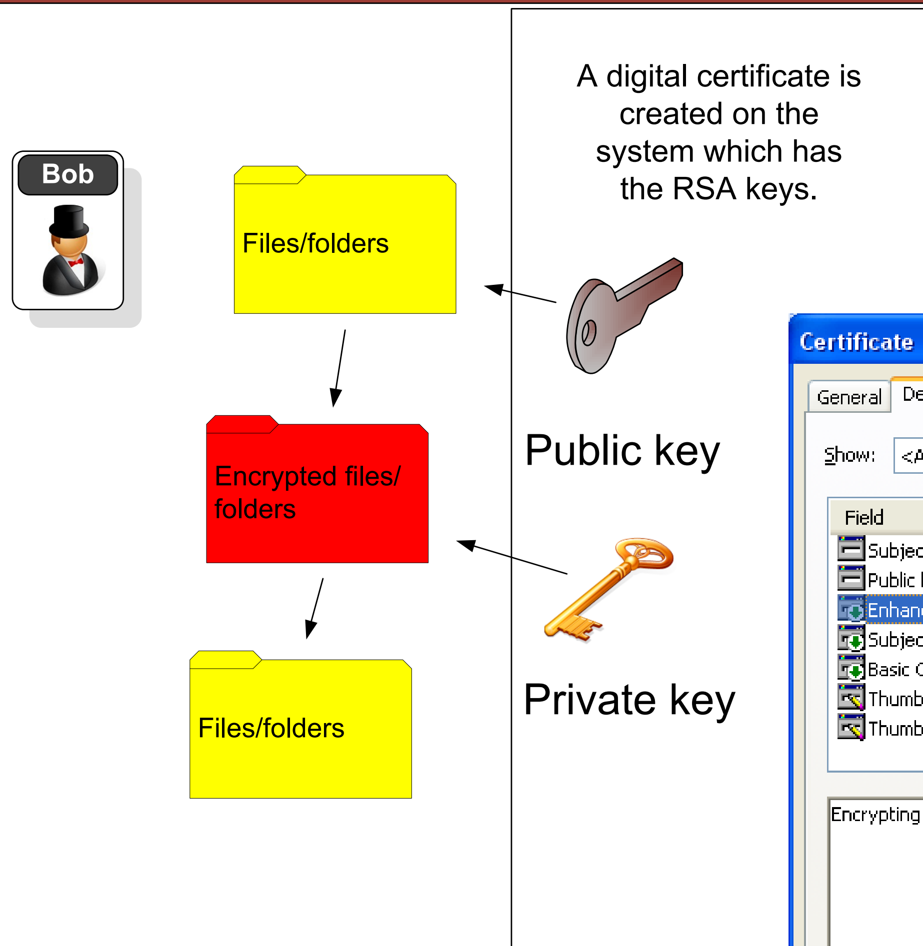
Encryption

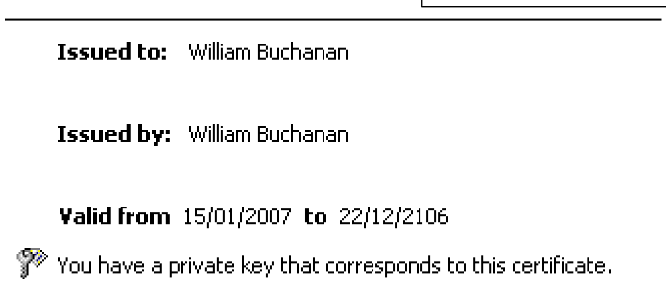


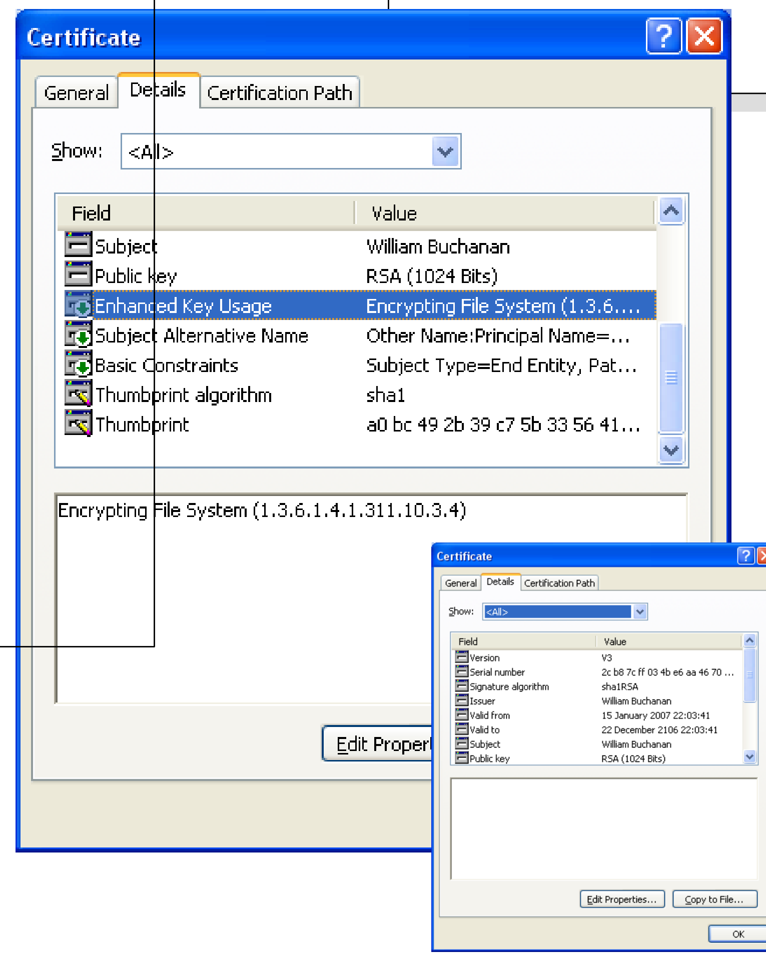
Encrypting disks

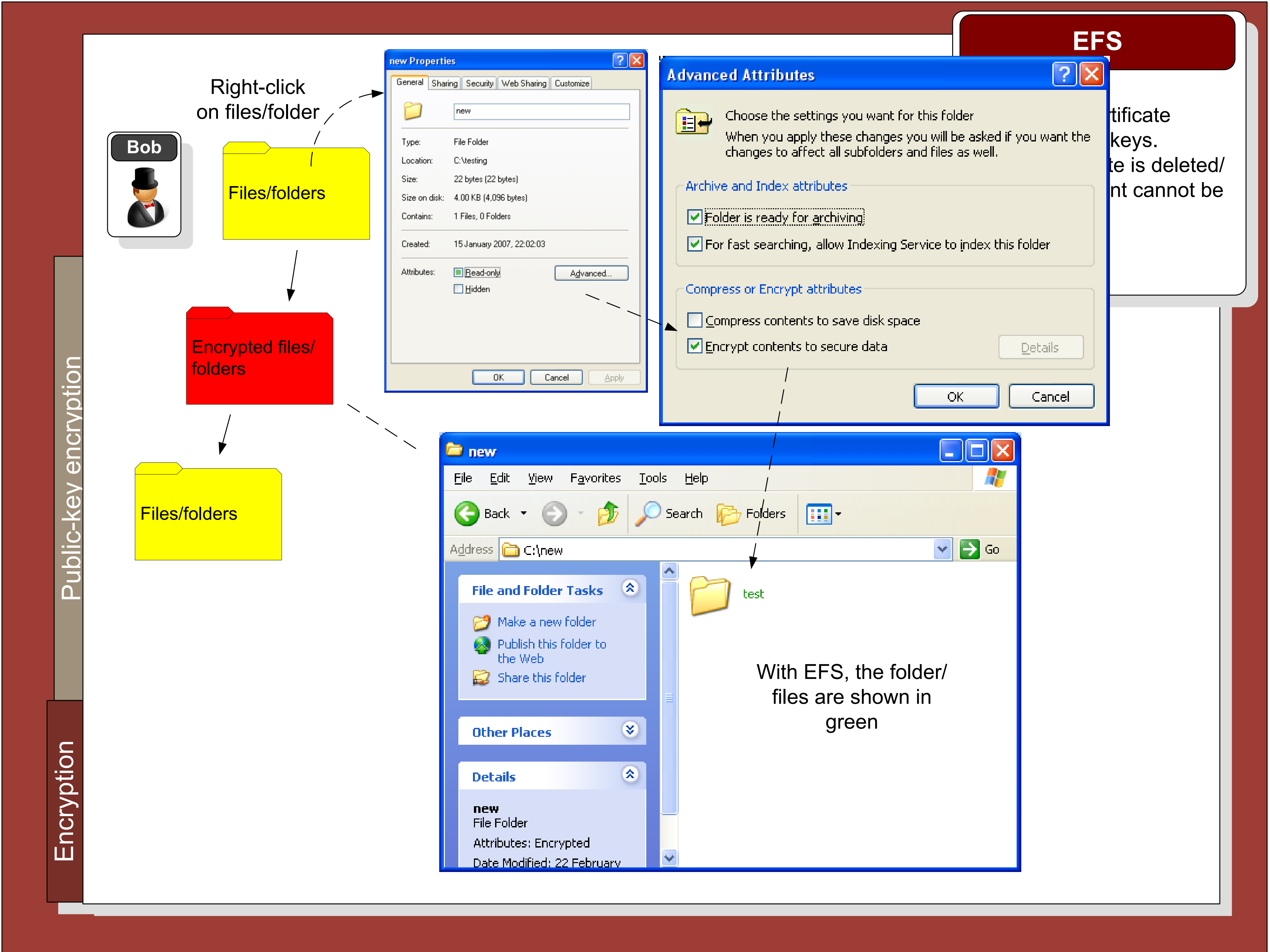
EFS

- The digital certificate contains both keys.
- If this certificate is deleted/ lost, the content cannot be decrypted.



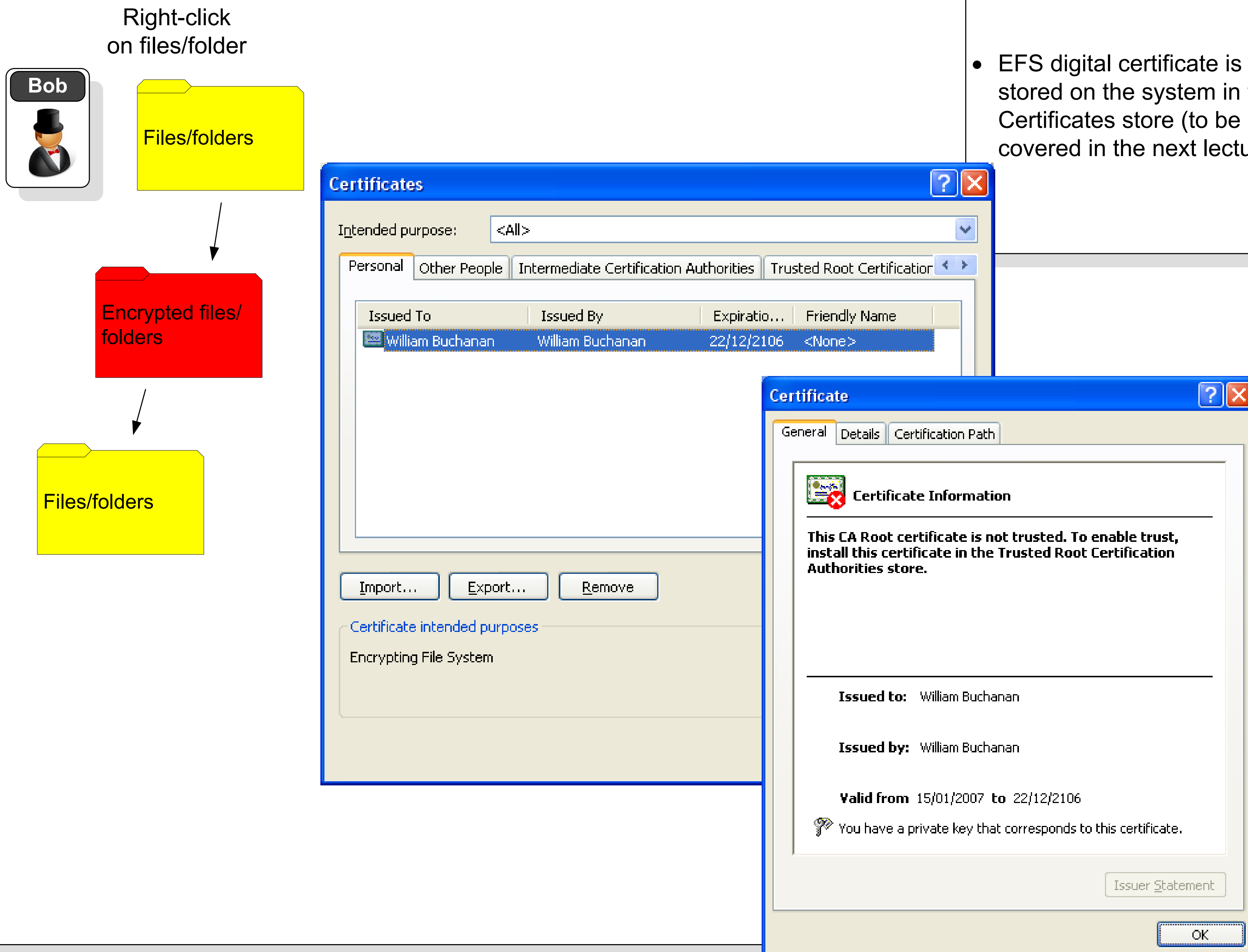





EFS
tificate
keys.
te is deleted/
nt cannot be
Public-key encryption
Encryption
Bob
Right-click
on files/folder
Files/folders
Encrypted files/
folders
Files/folders
new Properties
General
Sharing
Security
Web Sharing
Customize
new
Type: File Folder
Location: C:\testing
Size: 22 bytes (22 bytes)
Size on disk: 4.00 KB (4,096 bytes)
Contains: 1 Files, 0 Folders
Created: 15 January 2007, 22:02:03
Attributes:
Read-only
Hidden
Advanced...
OK
Cancel
Apply
Advanced Attributes
Choose the settings you want for this folder
When you apply these changes you will be asked if you want the changes to affect all subfolders and files as well.
Archive and Index attributes
Folder is ready for archiving
For fast searching, allow Indexing Service to index this folder
Compress or Encrypt attributes
Compress contents to save disk space
Encrypt contents to secure data
Details
OK
Cancel
new
File
Edit
View
Favorites
Tools
Help
Back
Search
Folders
Address
C:\new
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
Details
new
File Folder
Attributes: Encrypted
Date Modified: 22 February
test
With EFS, the folder/
files are shown in
green

EFS

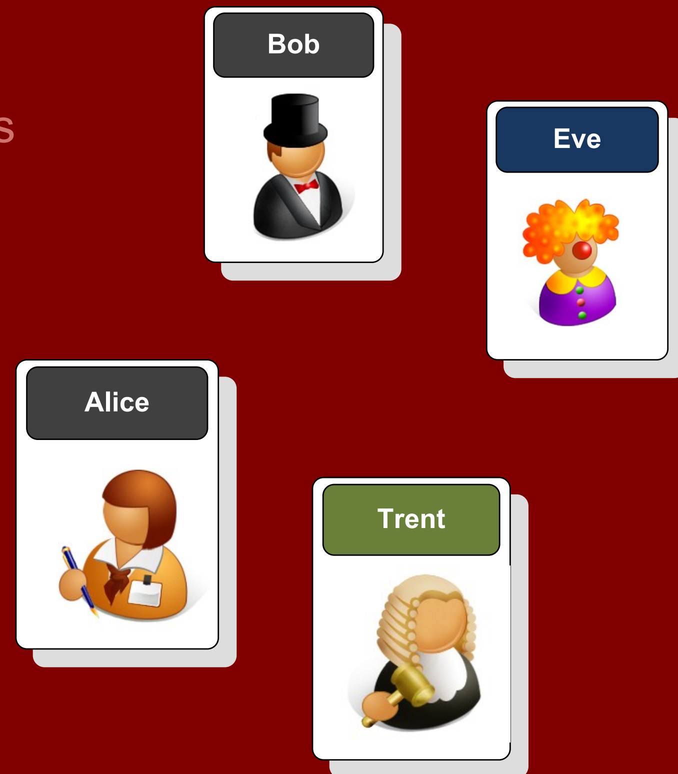
- EFS digital certificate is stored on the system in the Certificates store (to be covered in the next lecture).



Public-key encryption

Encryption

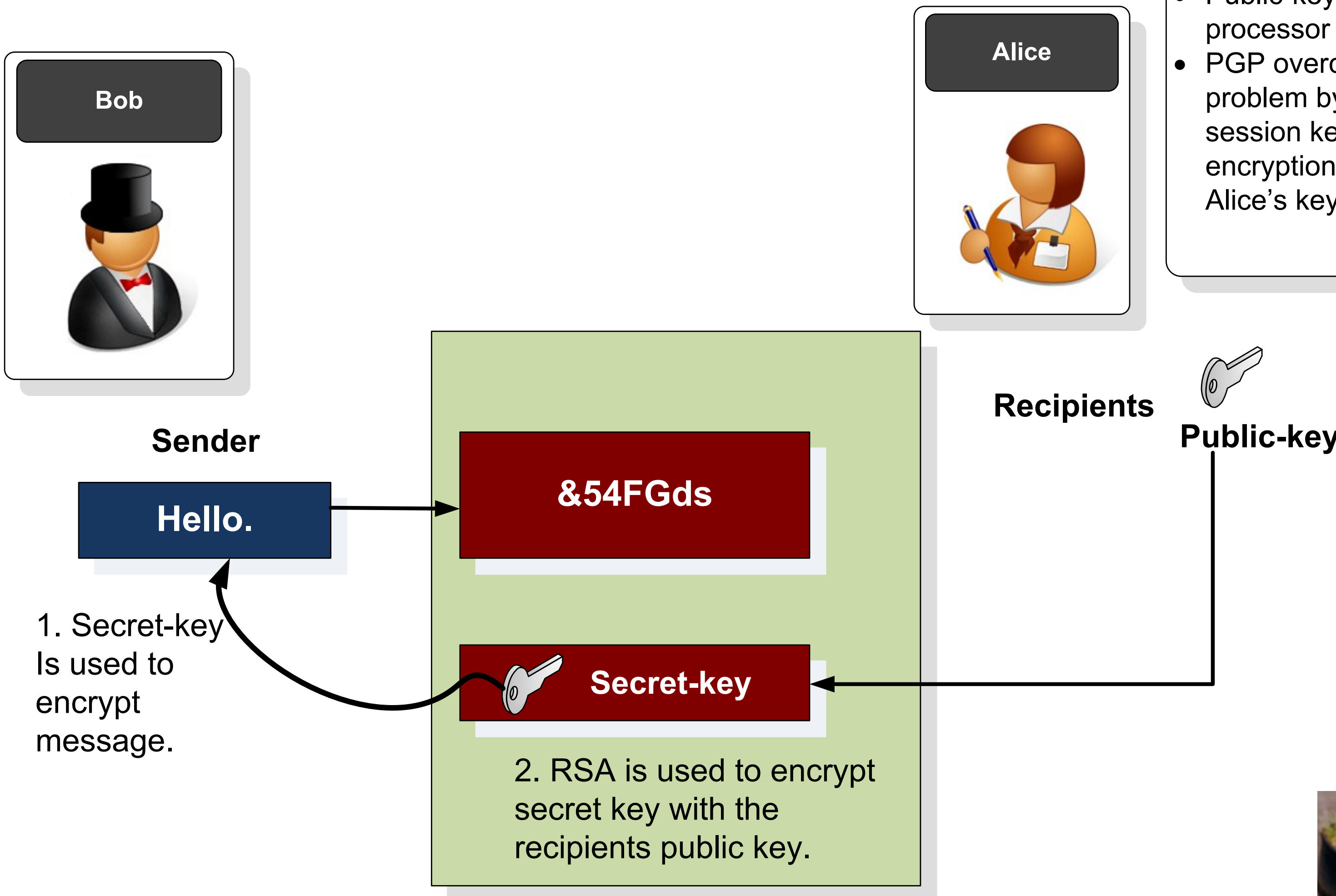
Encryption



PGP encrypting

PGP

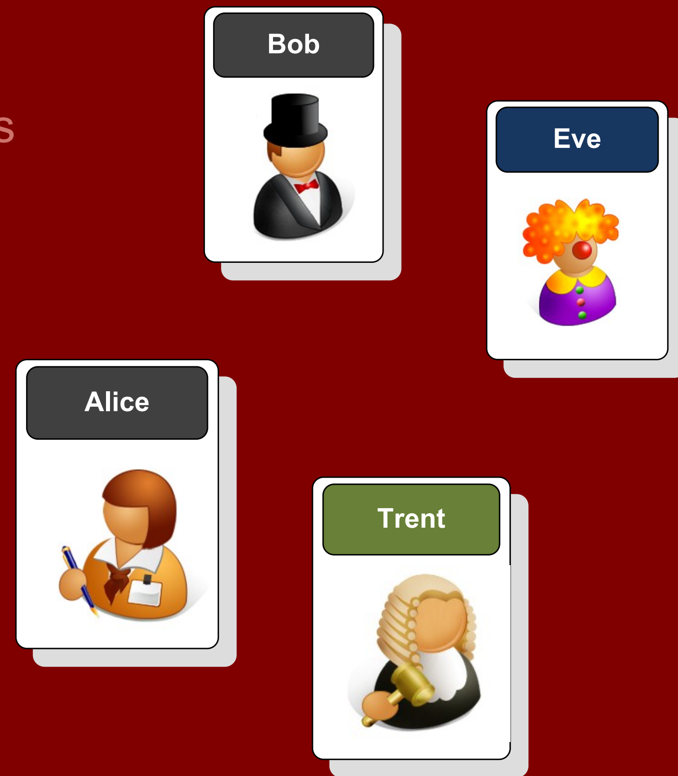
- Public-key is fairly processor intensive.
- PGP overcomes this problem by creating a session key for the encryption, and using Alice's key to encrypt it.



PGP encryption

Encryption

Encryption

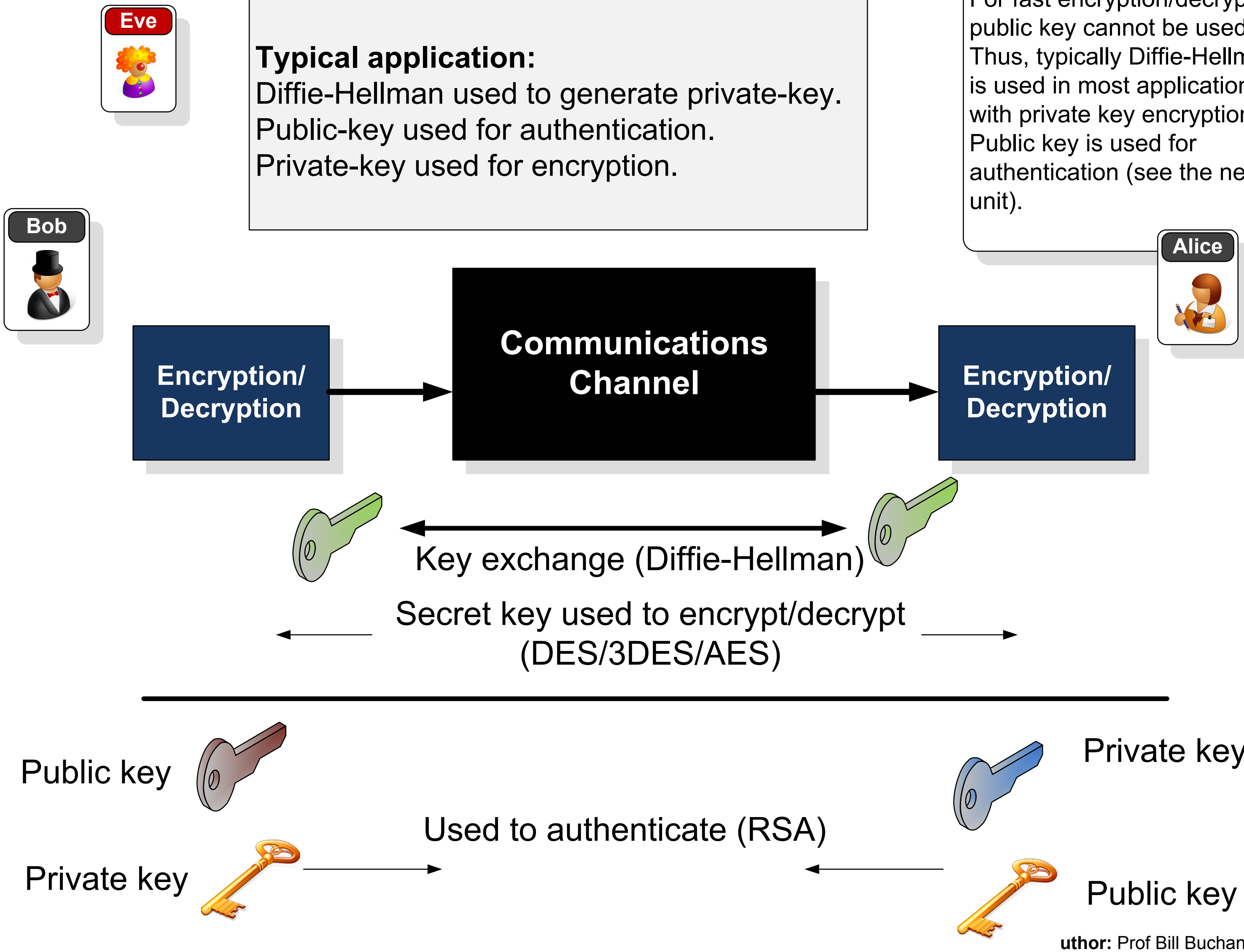


Conclusions

Typical application

For fast encryption/decryption, public key cannot be used. Thus, typically Diffie-Hellman is used in most application, with private key encryption. Public key is used for authentication (see the next unit).

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



uthor: Prof Bill Buchanan