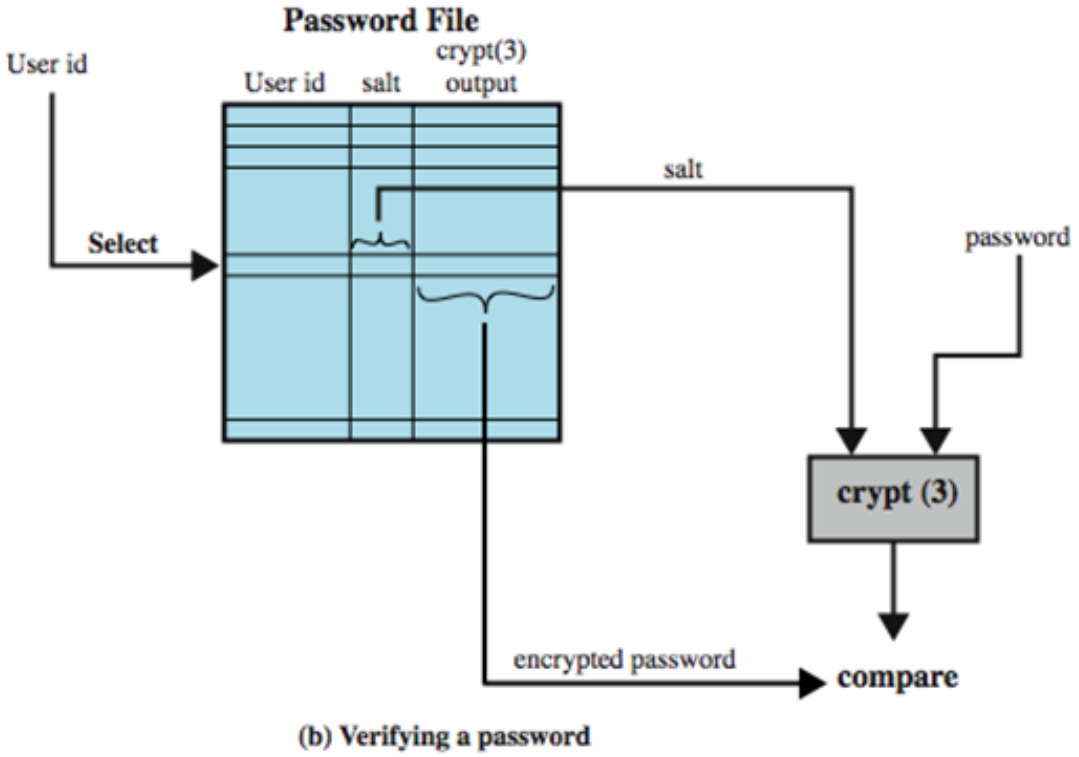
Functionality of Salt—verify a pwd

- Pwd verification
  - Locate the salt value according to user ID; calculate h(pwd||salt); compare it to the stored item in pwd file

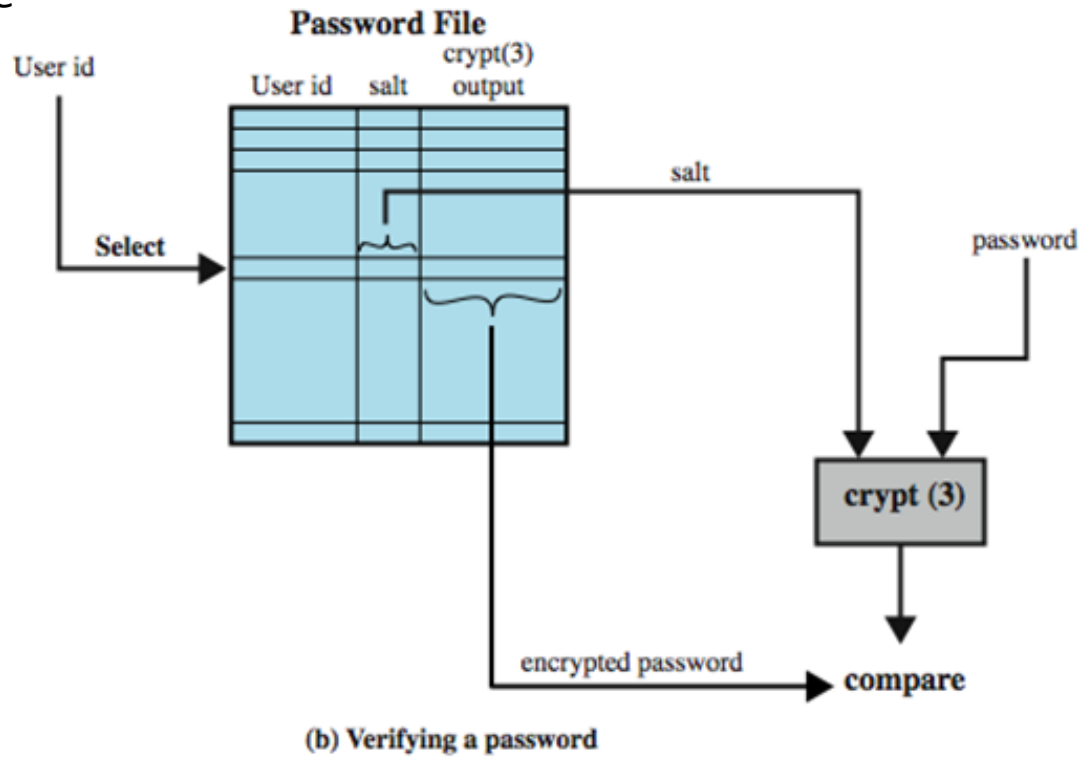


(b) Verifying a password

Why Can Salt Relieve Dictionary Attack?

- Even the attacker knows which user uses which salt (public/available) …

- For each dictionary password, the attacker has to compute Hash(dictionary password||salt), compare it with every entry in pwd file, multiple users→multiple rainbow tables
- without salt: the attacker only computes one rainbow table, compare it with every entry in pwd file
- the efforts differ when the attacker try to compromise multiple pwds



(b) Verifying a password

Questions

- Is it possible to thwart completely all password crackers by dramatically increasing the salt size to, say 24 bits or 48 bits?

Questions

- Is it possible to thwart completely all password crackers by dramatically increasing the salt size to, say 24 bits or 48 bits?
  - the purpose of salt is to let each user have a unique salt, such that even though two users choose the same pwds, their hashed salted pwds are different (different rainbow tables—cannot reuse rainbow table)
  - if 12 bits can guarantee uniqueness of rainbow table for each user, there is no need to increase salt size

Questions

- If there are 1000 users in a system, what is the suitable size of salt?

Questions

- If there are 1000 users in a system, what is the suitable size of salt?
  - 2^10>1000, 10 bits of salt
- The size of salt depends on the user amount in the system

One-factor authentication

- Not safe because
  - Phishing attacks
  - Possible to be cracked

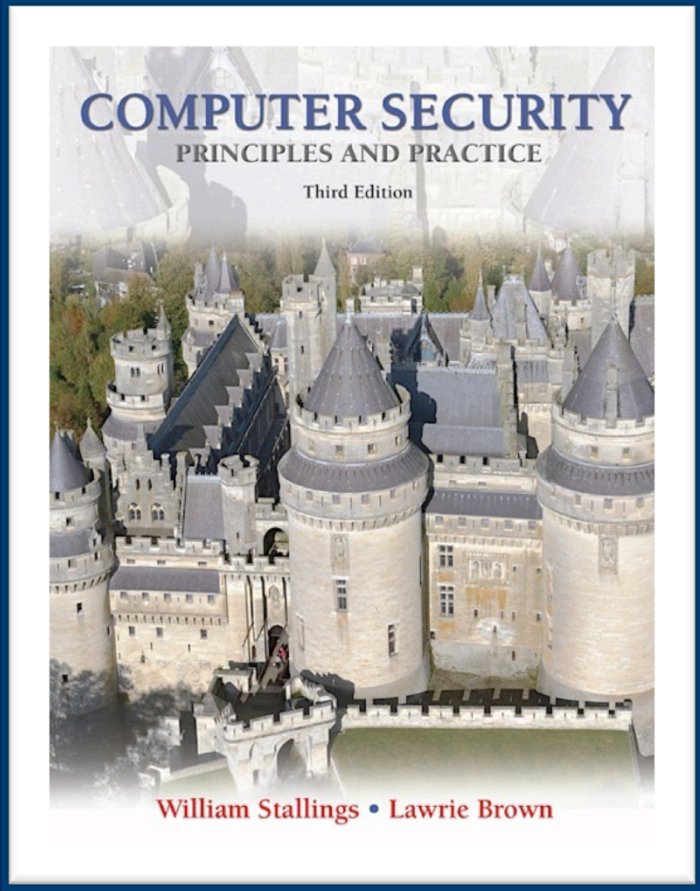
Two-factor authentication

- More secure because
  - Requires physical involvement
  - RSA-fob

Password managers

- Helps organize passwords
  - 1Password, Lastpass
    - https://youtu.be/Srh_TV_J144?t=18s

Lecture 4
Symmetric Encryption Techniques



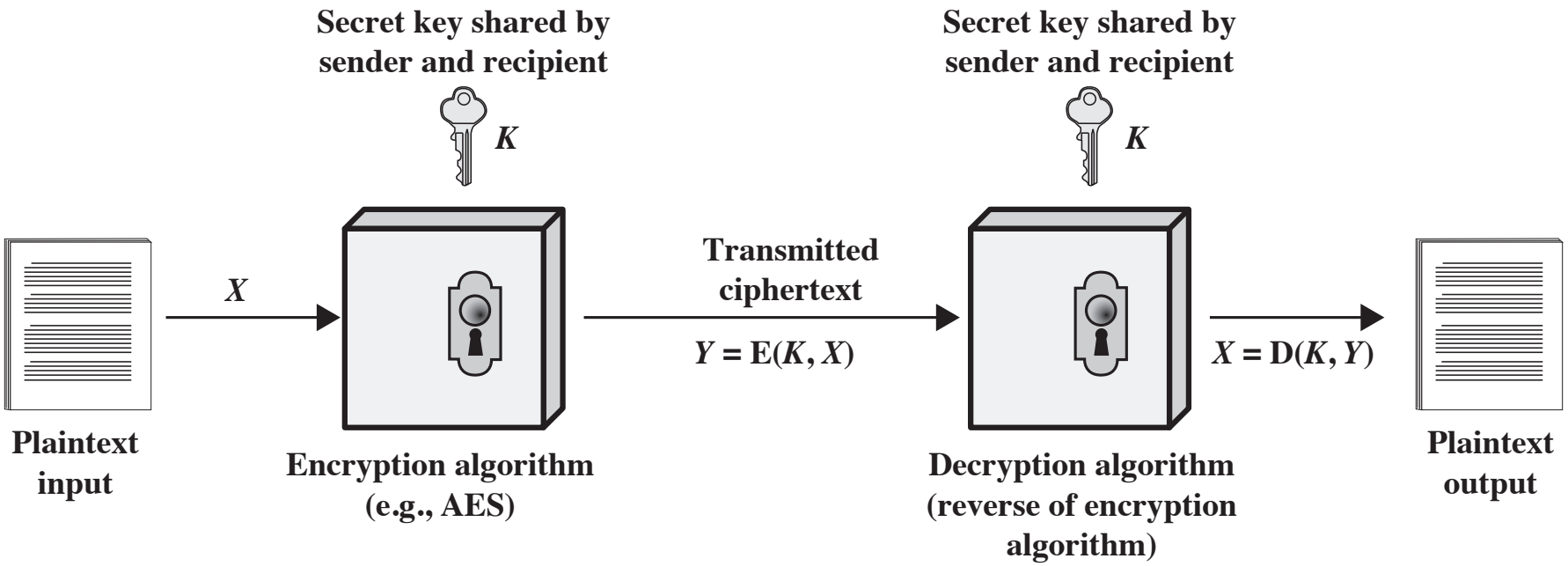
Encryption Technique Overview

- Classic encryption and modern encryption
  - Classic encryption
    - or classic cipher
    - dated back to Greek and Roman times
  - Modern encryption
    - mainly relies on Number Theory
    - was developed in 70s
- Symmetric encryption and asymmetric encryption
  - if the sender and receiver use the same key

Basic Terminology

- Plaintext
  - The original message
- Ciphertext
  - The coded message
- Enciphering or encryption
  - Process of converting from plaintext to ciphertext
- Deciphering or decryption
  - Restoring the plaintext from the ciphertext
- Cryptographic system
  - Schemes used for encryption
- Cryptanalysis
  - Techniques used for deciphering a message without any knowledge of the enciphering details

Model of Symmetric Encryption



Cryptoanalysis and Brute-Force Attack

Cryptanalysis

- Attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext
- Attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used

Brute-force attack

- Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained
- On average, half of all possible keys must be tried to achieve success

- Classic encryption
  - All of them belong to symmetric encryption

Symmetric Encryption Techniques

- Substitution
- Transposition

Substitution Technique

- Is one in which the letters of plaintext are replaced by other letters or by numbers or symbols
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- Simplest and earliest known use of a substitution cipher
- Used by Julius Caesar
- Involves replacing each letter of the alphabet with the letter standing three places further down the alphabet
- Alphabet is wrapped around so that the letter following Z is A
- Coding rule（encryption scheme）:

```
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
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
```

Caesar Cipher

- Exercise

  plain: meet me after the toga party

  cipher:
- Coding rule:

```
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
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
```

Caesar Cipher

- Exercise

  plain: meet me after the toga party

  cipher: PHHW PH DIWHU WKH WRJD SDUWB

- Can define transformation as:

```
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
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
```

Caesar Cipher Algorithm

- Can define transformation as:

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
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

- Mathematically give each letter a number

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- Algorithm can be expressed as:

$$c = E(3, p) = (p + 3) \bmod (26)$$

- A shift may be of any amount, so that the general Caesar algorithm is:

$$C = E(k, p) = (p + k) \bmod 26$$

- where k takes on a value in the range 1 to 25; the decryption algorithm is simply:

$$p = D(k, C) = (C - k) \bmod 26$$

Brute-Force of Caesar Cipher

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Other Substitutions: Permutation

- Alphabet is scrambled, each plaintext letter maps to a unique ciphertext letter

- *For example*

  a, b, c, d, e

  p = b, f, g, z, a

p(a) = b, p(b) = f, p(c) = g, p(d) = z, etc.

Attack substitution ciphers

- Brute force attack
  - 26! possibilities—for each cipher, try 26! times
- Cryptoanalysis
  - Short words,
  - Words with repeated patterns,
  - Common initial and final letters, …
  - English E, T, O, A occur far more than J, Q, X, Z

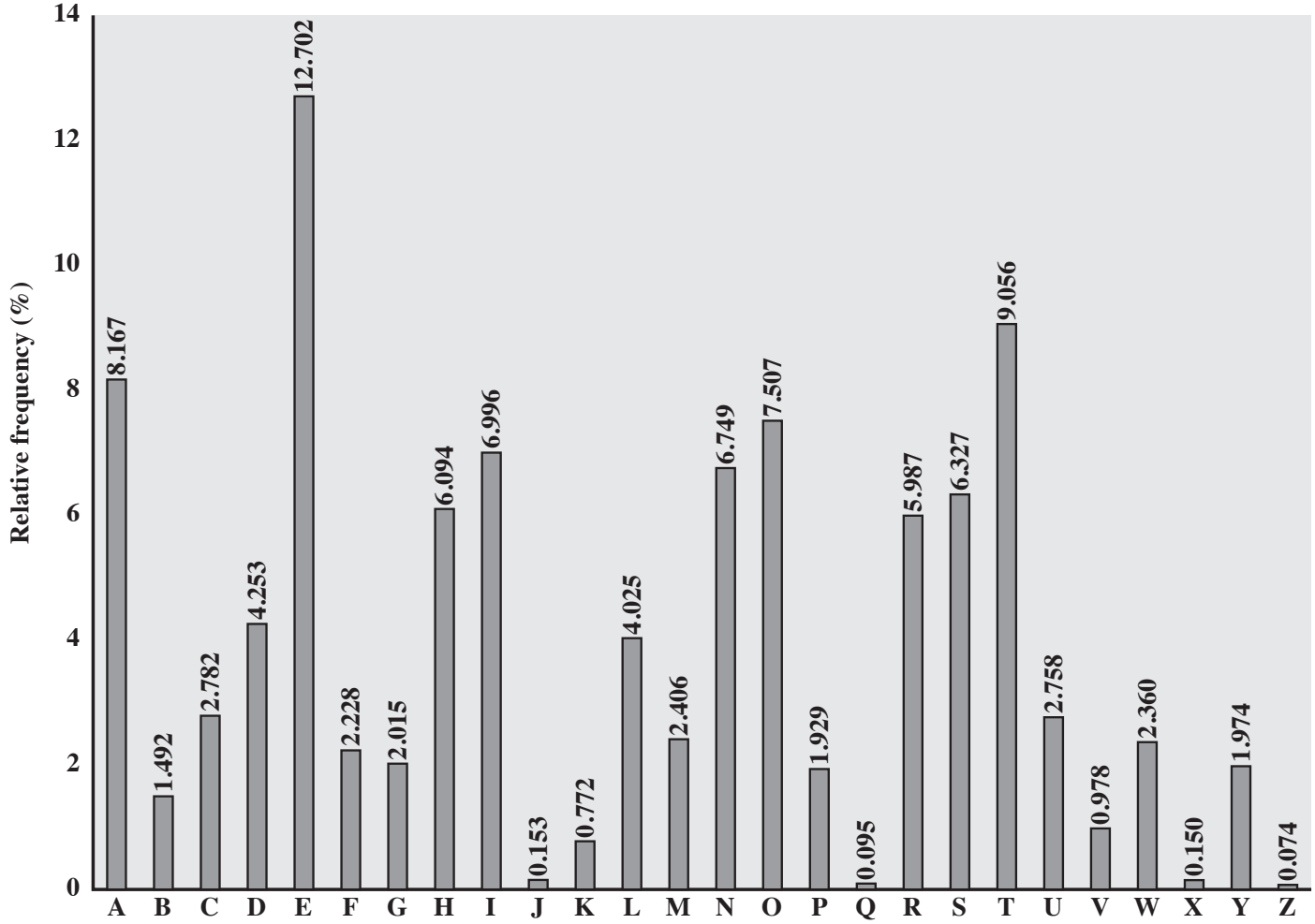


Figure 2.5 Relative Frequency of Letters in English Text

Cryptanalysis

- Example:

wklv phvvdjh lv qrw **wrr** kdug **wr** euhdn

wrr --> see, **too**, add, odd, **off**...

wr --> to, of

wklv --> Txxx,

Best guess: w = T, r = O

Cryptanalysis

wk***lv*** phvvdjh ***lv*** qrw ***wrr*** kdug ***wr*** euhdn

wrr --> see, **too**, add, odd, **off**...

wr --> to, of

Best guess: w = T, r = O

lv --> so, is, in, ...

wk***lv***

T_SO very unlikely...

T_IS likely

Best guess: l = I, v = S

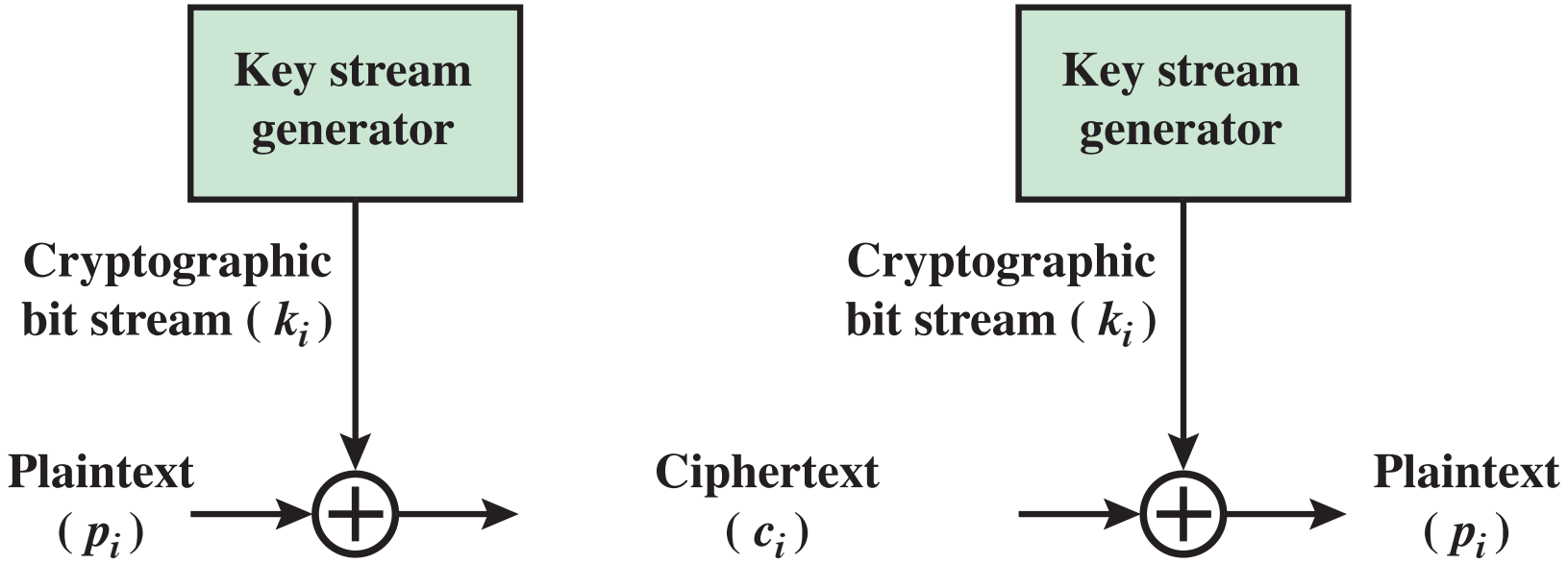
Cryptanalysis

Best guess: w = T, r = O, l = I, v = S

wk*lv* ph**vv**djh ***lv*** q**rw** **wrr** kdug **wr** euhdn

T-IS --SS--- IS -OT TOO ---- TO -----

Vernam Cipher



Example

- Plaintext: HELLO
- Key: DGHBC

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

One-Time Pad

- Improvement to Vernam cipher proposed by an Army Signal Corp officer, Joseph Mauborgne
- Use a random key that is as long as the message so that the key need not be repeated
- Key is used to encrypt and decrypt a single message and then is discarded
- Each new message requires a new key of the same length as the new message
- Scheme is unbreakable (perfect security)
  - Cannot infer the key

Difficulties

- The one-time pad offers complete security but, in practice, has two fundamental difficulties:
  - There is the practical problem of making large quantities of random keys
    - Any heavily used system might require millions of random characters on a regular basis
  - Key distribution problem
    - For every message to be sent, a key of equal length is needed by both sender and receiver
- Because of these difficulties, the one-time pad is of limited utility
  - Useful primarily for low-bandwidth channels requiring very high security
- The one-time pad is the only cryptosystem that exhibits *perfect secrecy* (unconditional security)

Symmetric Encryption Techniques

- Substitution
- Transposition

Transposition Cipher

- Write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns
  - The order of the columns then becomes the key to the algorithm

Key: 4 3 1 2 5 6 7

Plaintext: a t t a c k p

o s t p o n e

d u n t i l t

w o a mx y z

Ciphertext:
TTNAAPTMTSUOAODWCOIXKNLYPETZ

Main disadvantage of Symmetric Enc.

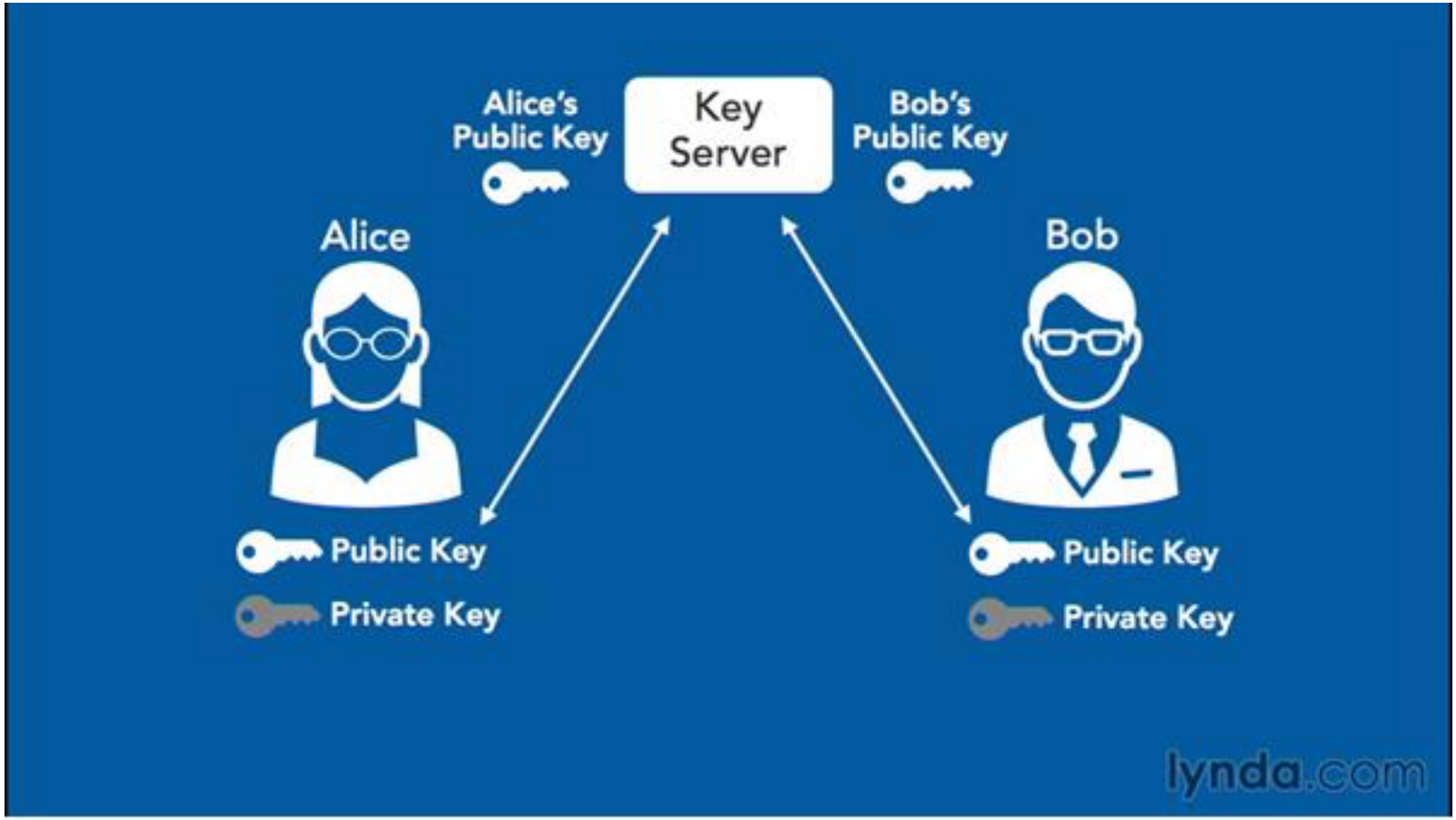
- ?

Main disadvantage of Symmetric Enc.

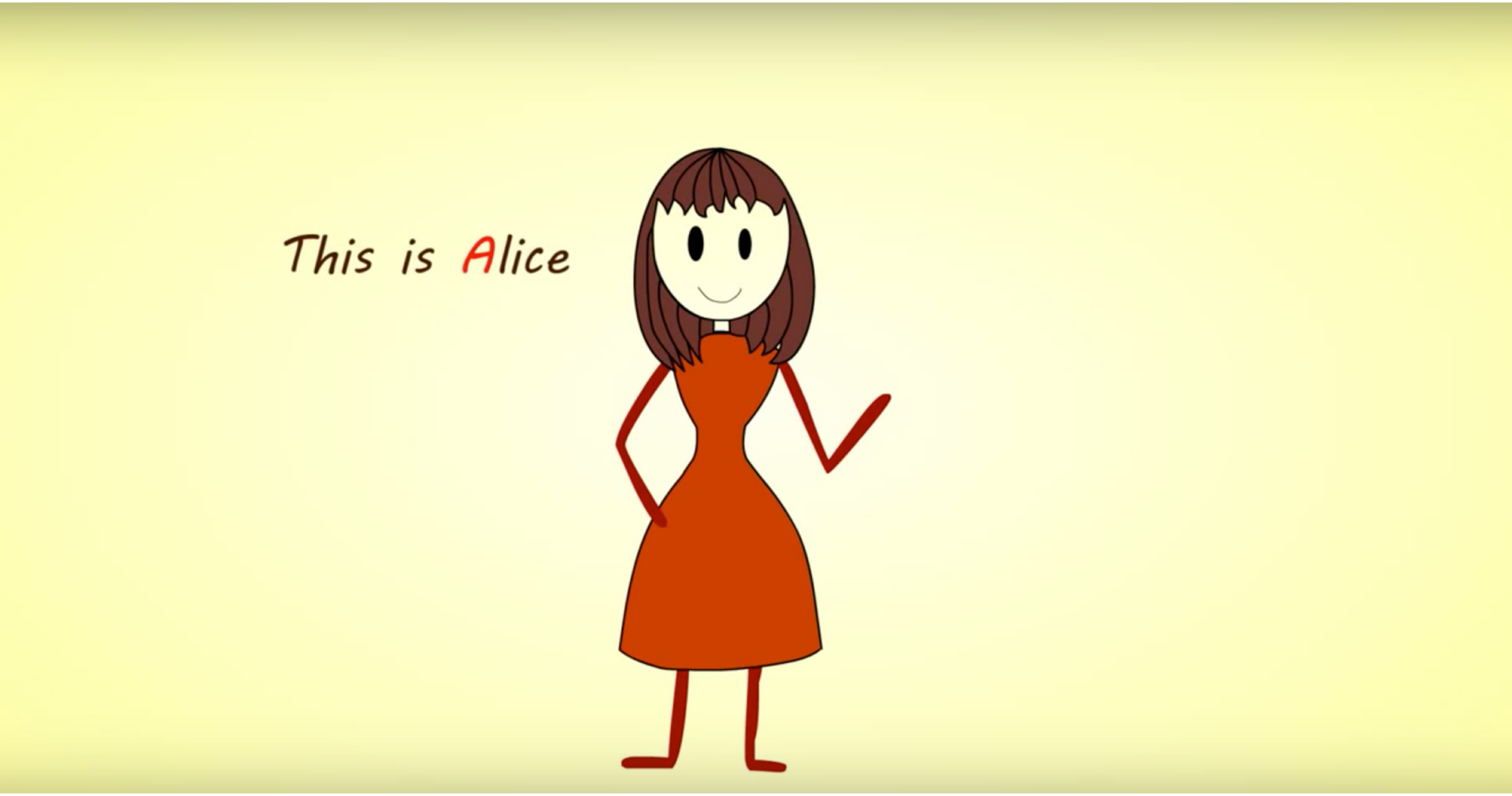
- Key distribution problem
- Chicken and egg problem

Asymmetric Enc. (Public key enc.)

- Public-private key encryption

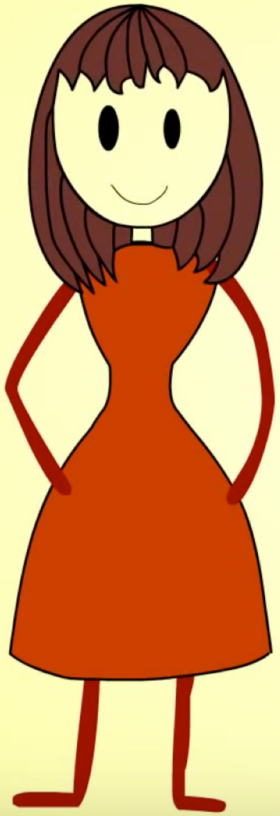


Public key enc.



Public key enc.

First she makes a lock, a key, and a password to protect the key

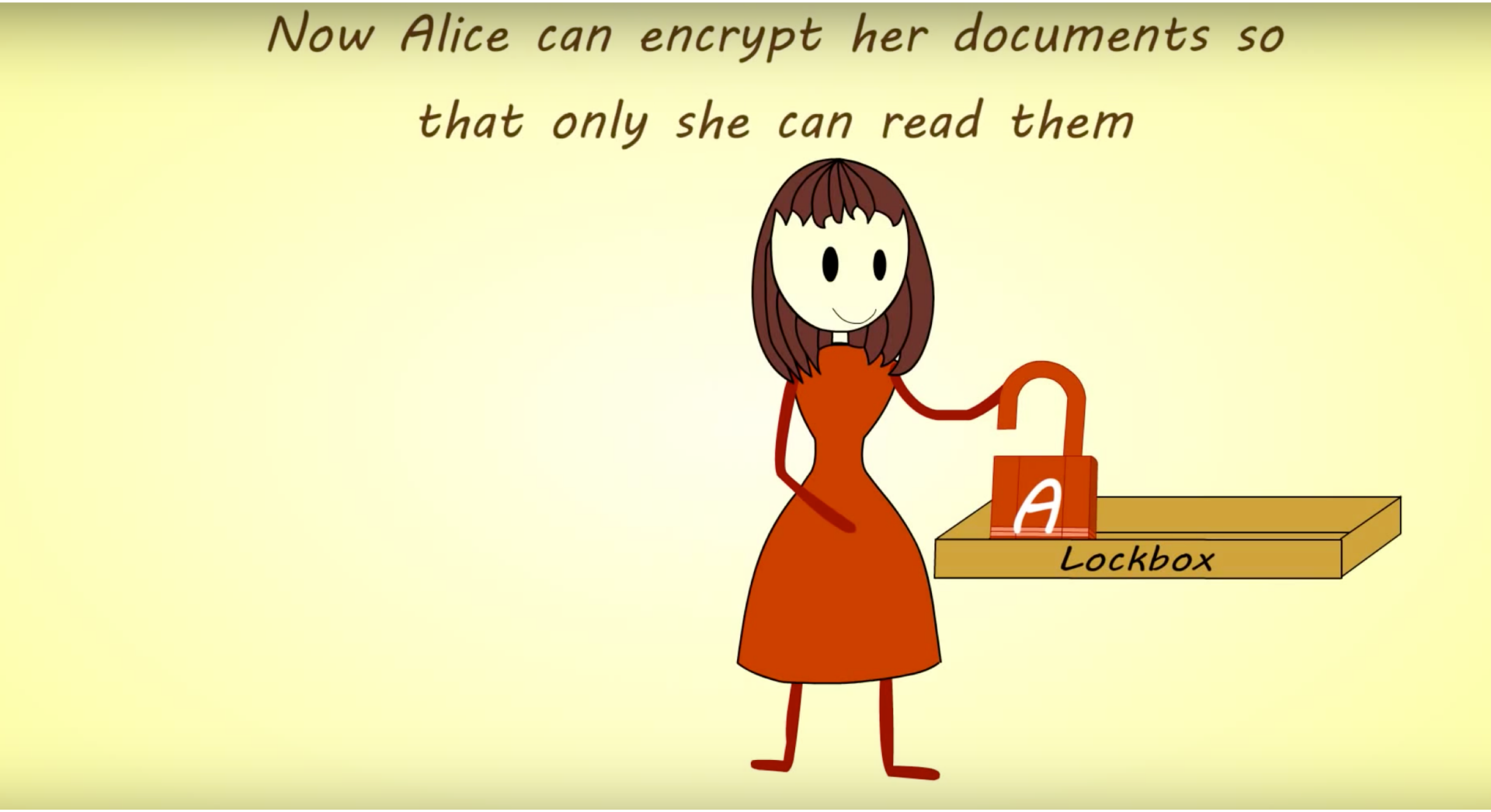


Password

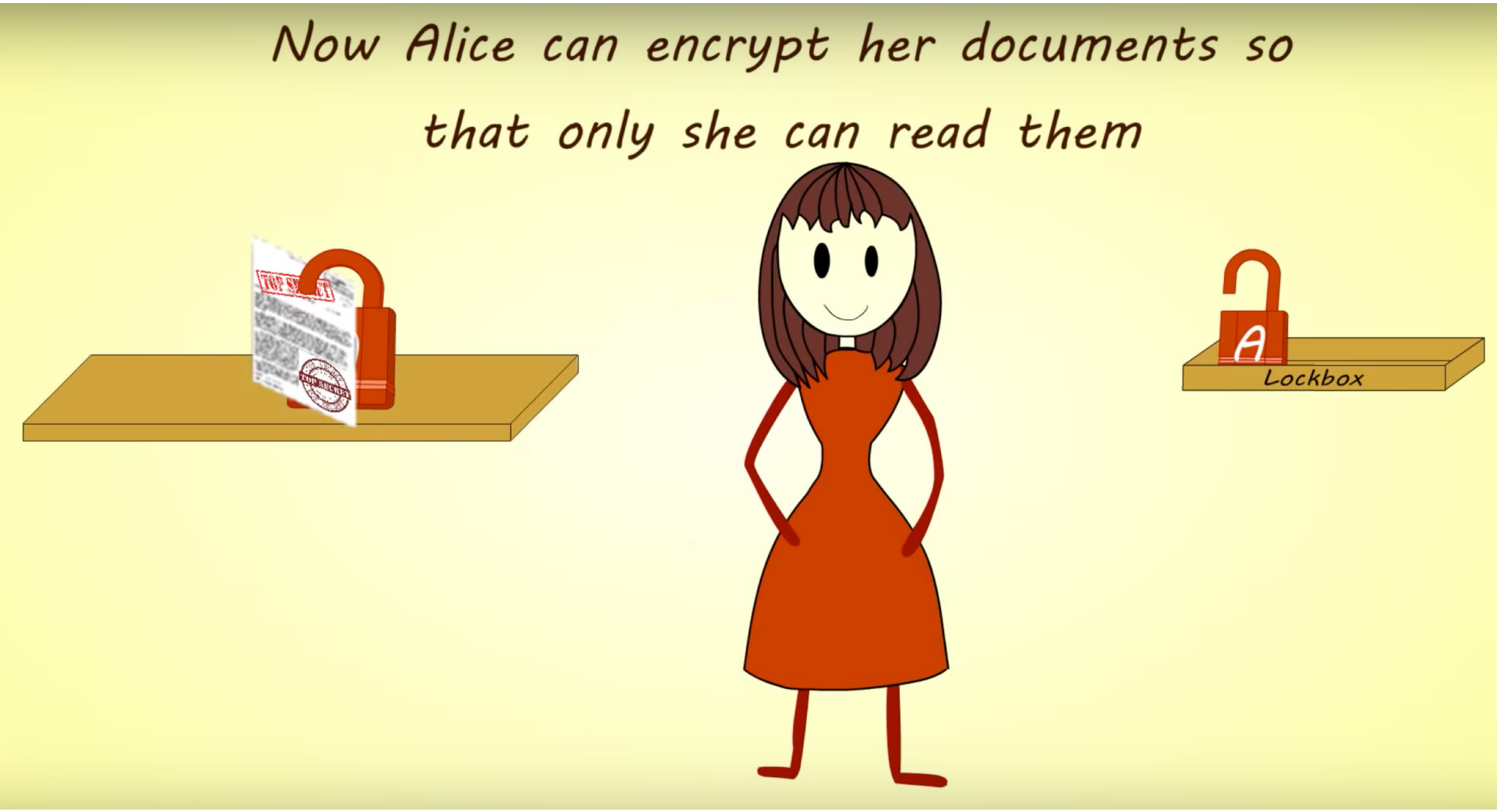
3 0 2 9 3 0 1 4

Public key enc.

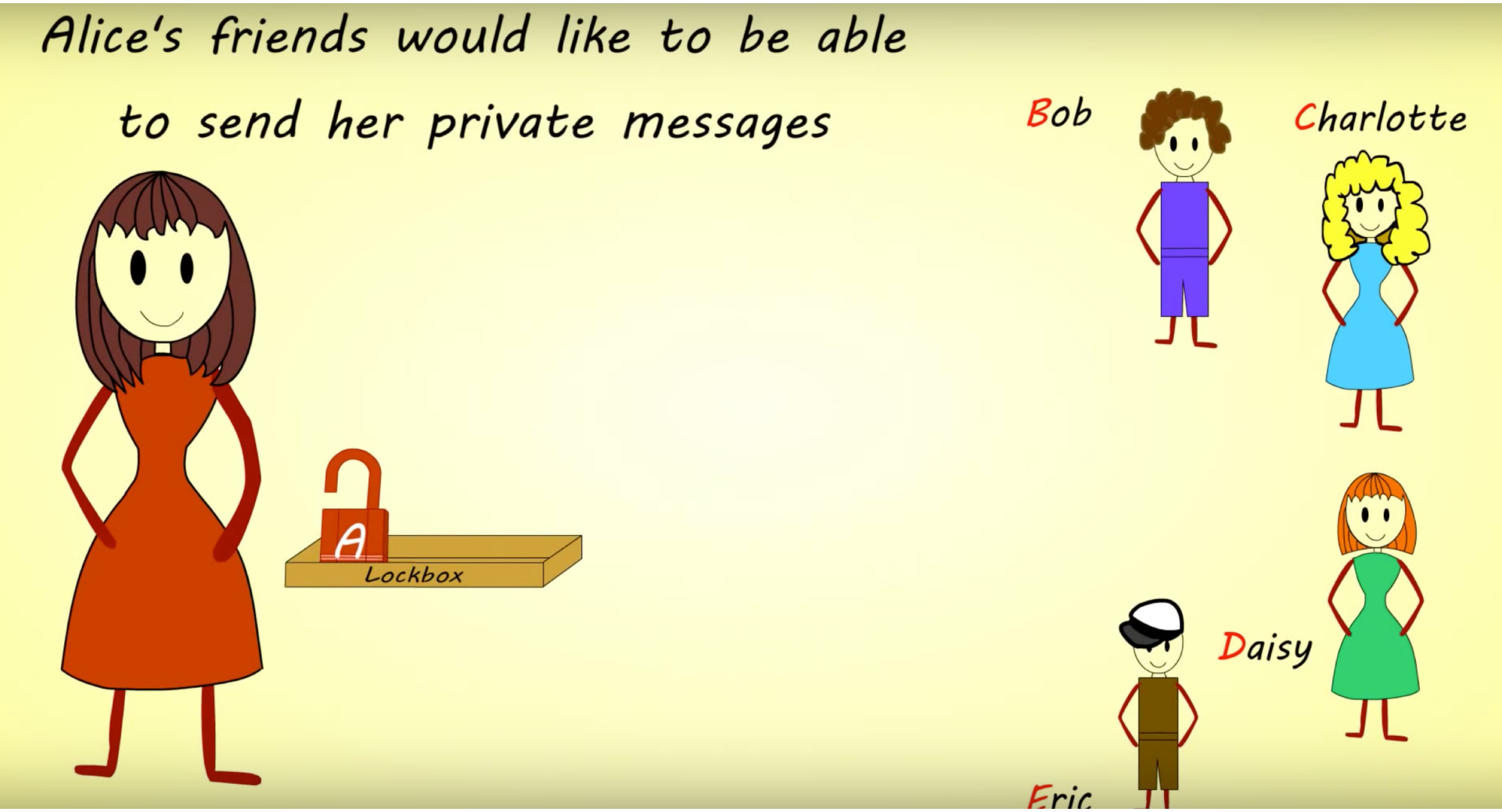
Public key enc.



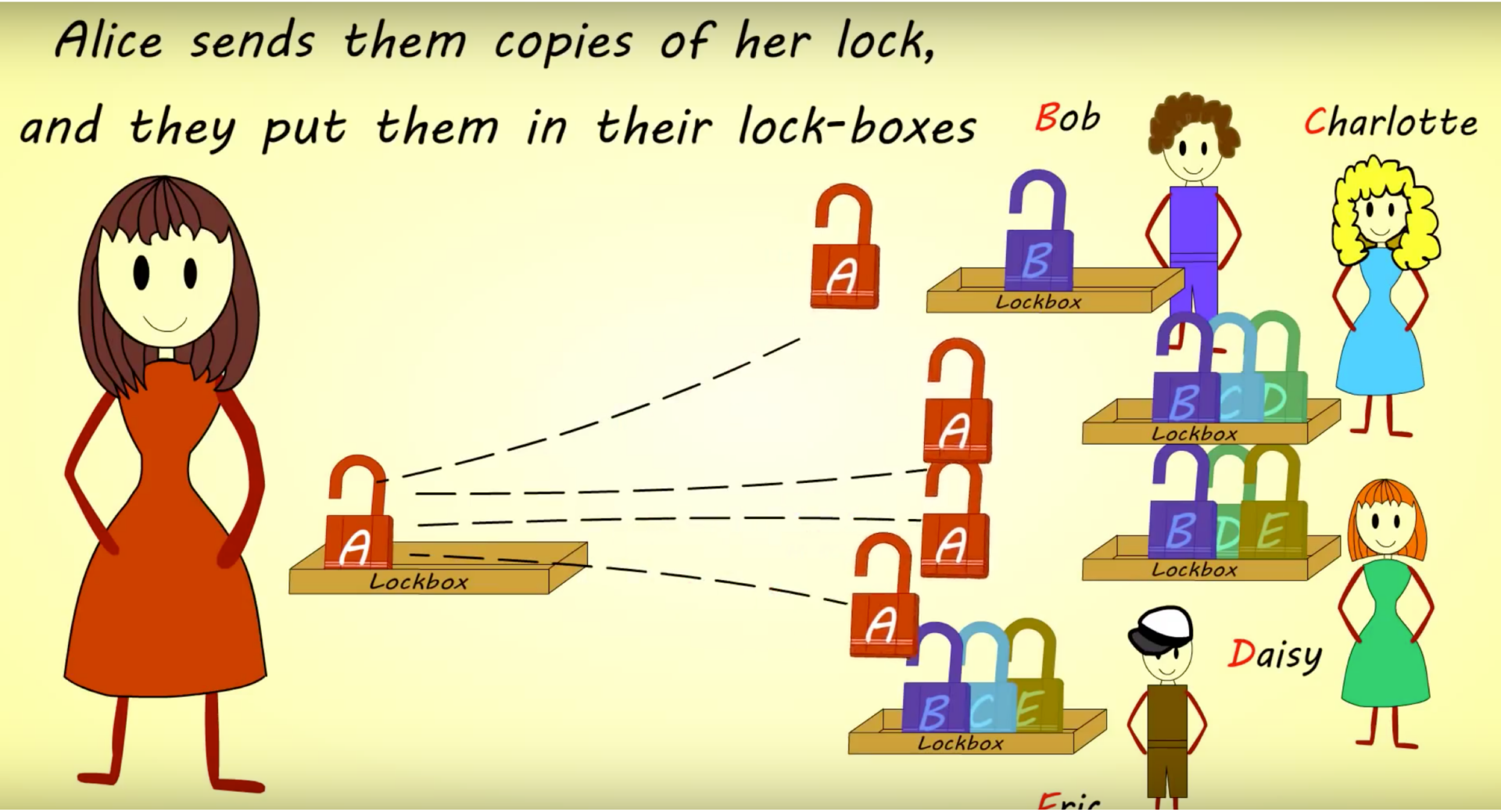
Public key enc.



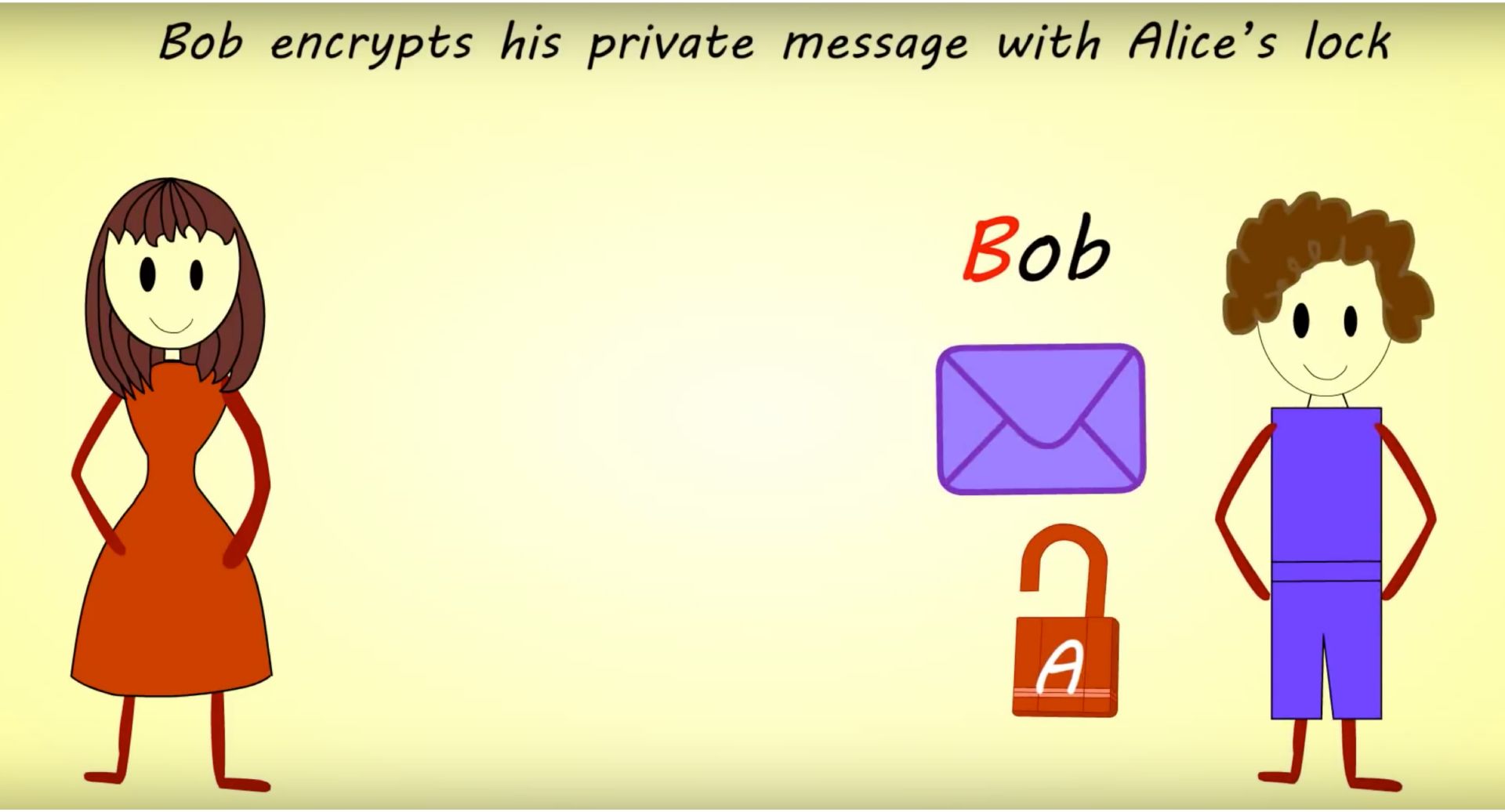
Public key enc.



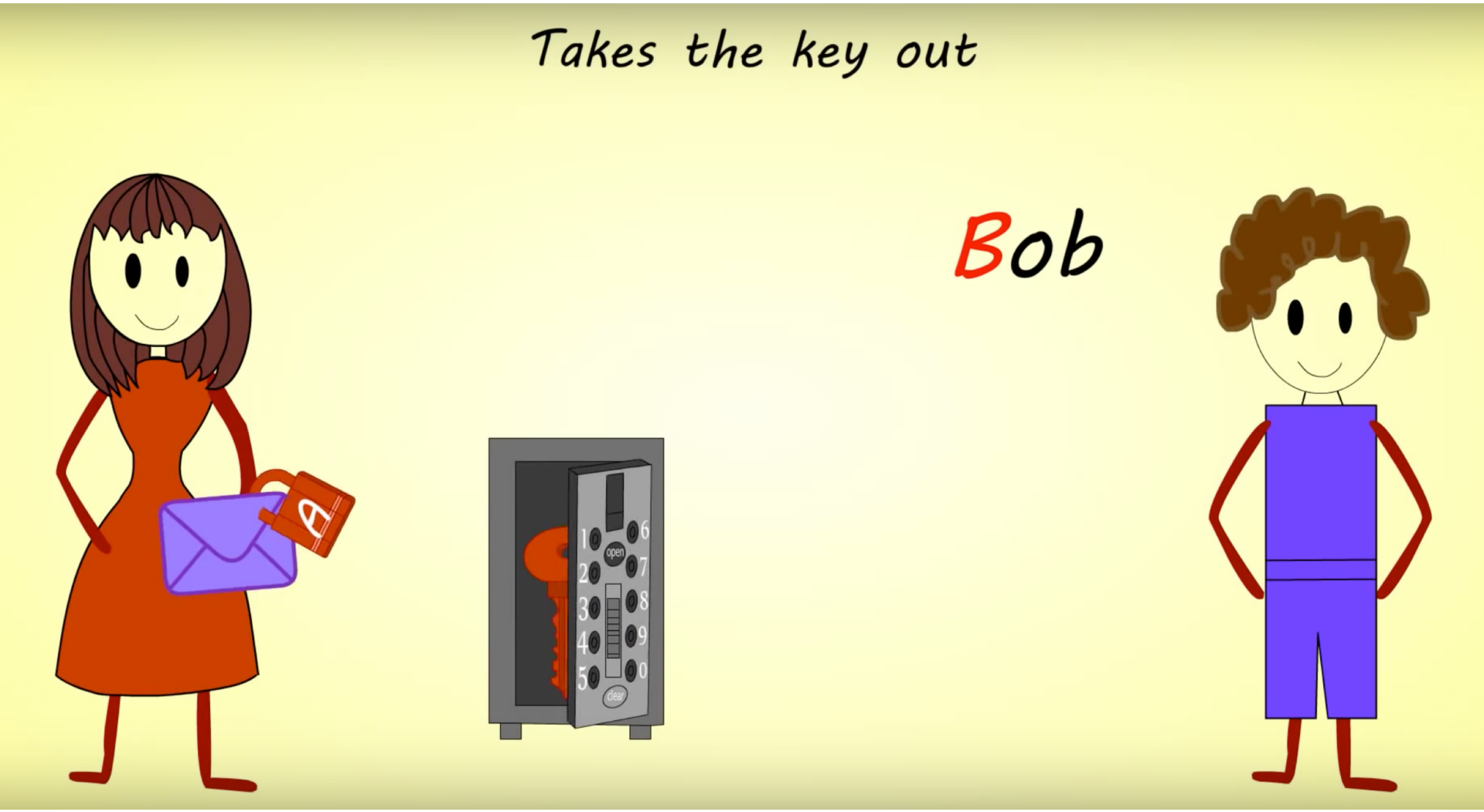
Public key enc.



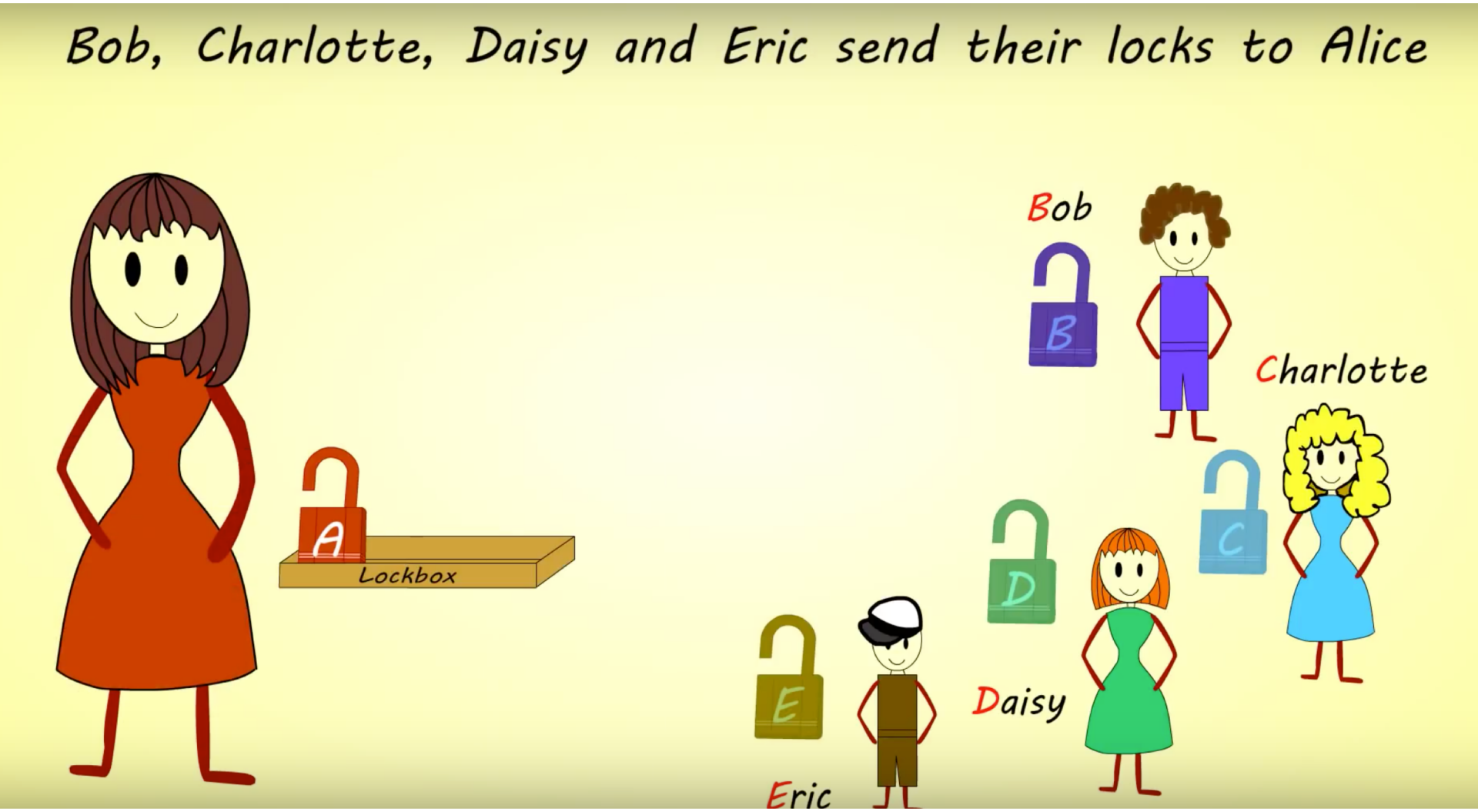
Public key enc.



Public key enc.



Public key enc.



Public key enc.

Public key enc.

And here is how an encrypted message looks

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.11 (MingW32)

HfhcEWfdRlTuiaUaZ8Hs01L/j58Y/+LdwY+Aend9tnJlbqUng7gK+tByVt4bsKw3
d6uv2OeYL8K7g6DO6idaDf0B01ZcO8uBWv5g23qB+ScFsj0stC7BhOgcajtmodYD
rQLgdUXsRzRdH3iaW3ftG4PxbTmjUuZgNMp5qrYXh3clGZ5W3WpKbfOy50AqY5mx
jTVPIugYPjgHLBzHeMCQ4nCRA+r1nXUdArNJbAIIsoGWMwcnRazZHi2z3NQrjCGD
RnxF0zu+2TmiZTJHy81FOwlz4x0d73Y81jFY/K9GQyT1YN0C11sR3aBDlpXQ0zlo
l/PvVf5e4XGnwVHDmn/taJLzoGMts9mfbWConJXI9h2ZAEHn12o5VSqVYA2fR0Bg
cmuK8KJ3s36FH1Q/AwHMh3hOc/4gcR7GJ57C0dOf8rYtnqc/XNmwRZelcXDk0v3H
L3gDa6xkF6ZQDCObF58tL8qKK34cdd5LNO3r9/sLwp4s1F1sYJToWRv/hGL0X5Q8
qElydy6/g4vl2CgPDvW6QDGDwQACDtzzOz0lt9F4GX0ETzdNwrp5R14re5X/NBLZ
wymER11HEMGYLWeGNy3TM7/oQWPS6B7sllbrQ1b3a//Yf31tyleCBK1lzUKRdN1R
2x+01E/iM+N1+uFnQWPuZjj0mj4k618u3dW0A9VU+xvm0gicsJa9TNzG9D5NUXaL
VuHh5rzSeQVDNW0rmd5pWVEb1NJlfwv2is2LnIytZGQFnt/2TmQMMO5yLmh4nVUY
dQoE3YKAKCTZKEcJ0eXCr1aotH6SL9B2TMLwb3epyJNpHwdnIdvT938VjMpx6KcM
4jJuzPmiMJCUdoy8OiJXfJ72aiiN2WWLix4rrH9lpesmRy0zMSsZ99qE26oHa50g
4HdTtUzN3iZsmWMj5i9w85bY+trSsPUkDprhF1HQO6DatqfetG7RaB5fSA7ht2q7
tm1N1WNDGARTG7BYJEJshaDt1txkA8gciWR1hSyhfEr41W19Ae4qD5EsAMNr4xof
v/vpM26OlEcWUR6S00DOXAtToU5fOwPkyZQvFa/OJTeuHaq2cblTWUOFruXP3Srt
3x0QKOpEtoLMQ8fnfagIYcnlY7sSeUMtvj+VBDXw2WR8ojQL
=9Aao
-----END PGP MESSAGE-----
```

Asymmetric Enc. (Public key enc.)

- Dropbox
  - Server sync
  - Sharing with others
  - Same key enc.
  - Deduplication

Weakness

- There does not exist an alg. as secure as OTP
- Computational cost
  - For efficiency, hybrid systems can be used