Part VI

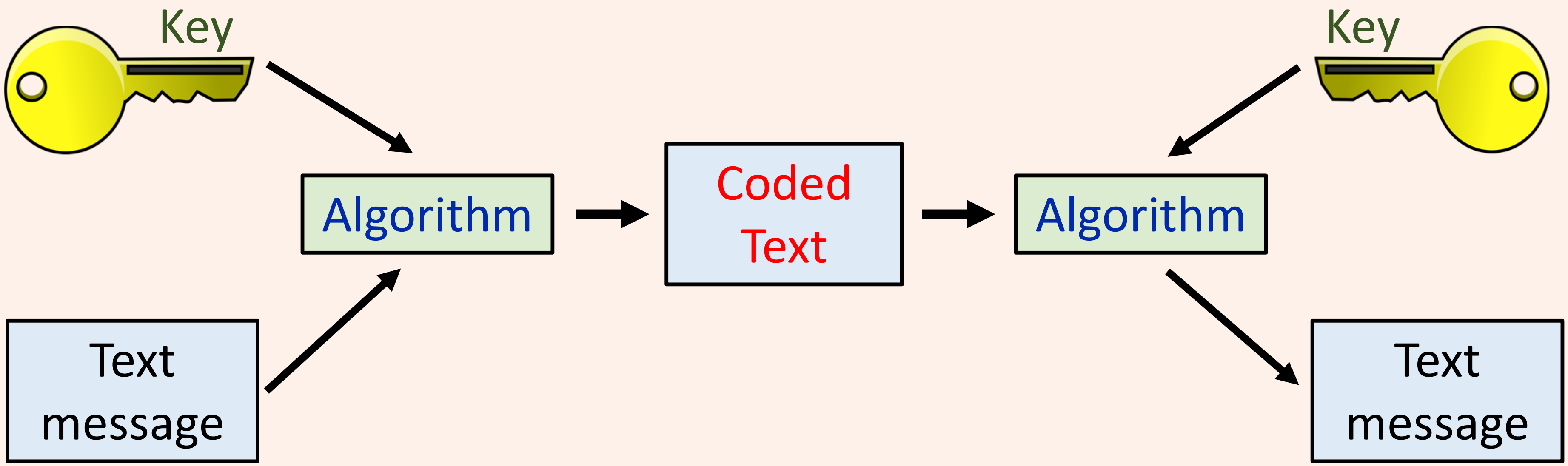
Looking into the future

Lecture 33

Quantum Cryptography
(Quantum key distribution protocol BB84)

Cryptography

Sending a message so that it cannot be read easily by an eavesdropper



Adapted from: *The Code Book, The secret history of codes and codebreaking* by Simon Singh

Example 1: Caesar cipher

Shift each letter of the alphabet in a message by **1**

KEY	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
	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

Text message: Let us watch the movie An Action Hero

Coded message: MFU VW XBUDI UIF NPWJF BO BDUJPO IFSP

Example 1: Caesar cipher

Shift each letter of the alphabet in a message by **3**

KEY	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

Text message: Let us watch the movie An Action Hero

Coded message: OHW XV ZWWFK WKH PRYLH DQ DFWLRQ KHUR

Example 2: Vigenère cipher

- Employ the Caeser cipher algorithm using a different Caeser shift for each letter of the message.
- How to shift the letter is decided by a key , say **HALF** .
- Now use alphabets **A-Z** replaced by **H-Z-A-I** for the first letter of the message, **A-Z** as **A-Z** for then second letter, **A-Z** replaced by **L-Z-A-K** for the third letter and **A-Z** replaced by **F-Z-A-E** for the fourth letter and repeat .

Vigenère square

Key word:
HALF

	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
A	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
B	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
C	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
D	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
E	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
F	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
G	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
H	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
I	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
J	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
K	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
L	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

Text message: Let us watch the movie An Action Hero

Key: hal fh alfha lfh alfha lf halfha lfha

Coded message: SEE ZZ WLYJH EML MZAPE LS HCENVN SJYO

Example 3: ADFGVX cipher

Code the letters and numbers using the following coordinate system as the key

	A	D	F	G	V	X
A	1	i	n	r	4	d
D	h	c	Ø	v	e	m
F	q	2	w	7	l	t
G	9	0	5	b	f	3
V	s	j	g	x	u	z
X	6	a	y	k	p	8

Text message: meet me at canteen at 10

Coded message: DX DV DV FX DX DV XD FX DD XD AF FX DV DV AF XD FX AA DF

ADFGVX contd ……..

Make the code harder by further operations using another keyword. In this case take it to be **EIGHT**

DX DV DV FX DX DV XD FX DD XD AF FX DV DV AF XD FX AA DF

E	I	G	H	T
D	X	D	V	D
V	F	X	D	X
D	V	X	D	F
X	D	D	X	D
A	F	F	X	D
V	D	V	A	F
X	D	F	X	A
A	D	F		

Now arrange the keyword in alphabetical order and transpose the columns accordingly. So arrange **EIGHT** as **EGHIT** and write letters serially column wise to form a code.

E	**G**	**H**	**I**	**T**
D	D	V	X	D
V	X	D	F	X
D	X	D	V	F
X	D	X	D	D
A	F	X	F	D
V	V	A	D	F
X	F	X	D	A
A	F		D	

Coded message: DVDXAVXADXXDFVFFVDDXXAXXFVDFDDDDXFDDFA

To decipher a code, one must know what **algorithm** has been used and what are the **keys** used to encode a message. So the important aspect of making good codes is to have good algorithm and a **secret key**. If one makes a key in such a way that it cannot be found in a reasonable time by an eavesdropper, the code is safe.

We now discuss how modern coding is done using **bits** (0 and 1). We will then see how the key for it is generated and transmitted. We will then learn how this key is made using quantum mechanics and is almost impossible to break.

Coding and decoding a message using bits

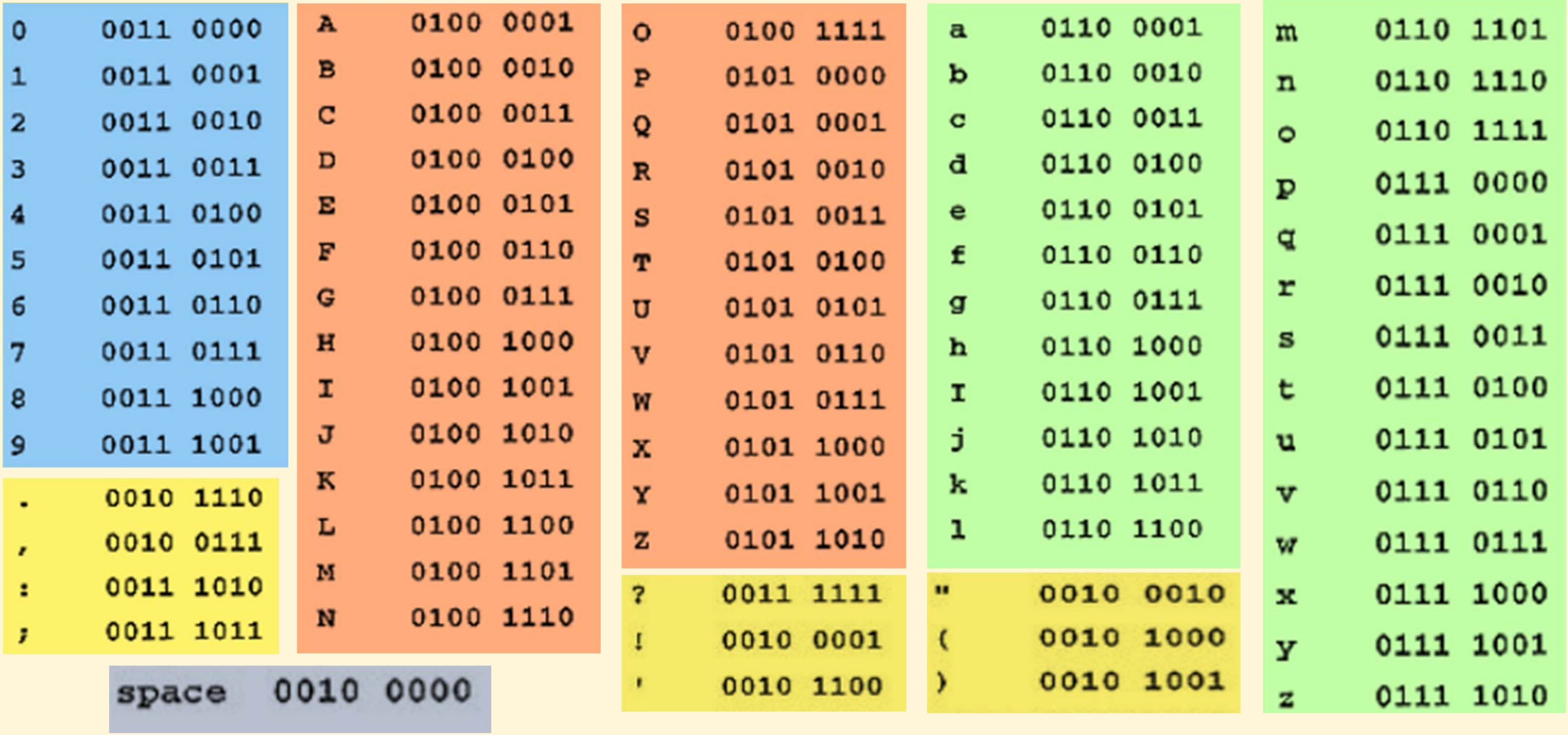
To code:

- Step 1 - Translate the message into bits using **ASCII** (American Standard Code for Information Interchange) code for character to binary conversion;
- Step 2 - "Add" a binary key to it to generate the binary code.

To decode:

- Step 1 - Use the key to get back the original message in binary;
- Step 2 - Translate the message back from binary to regular text message

ASCII (American Standard Code for Information Interchange) code for character to binary conversion



0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001

.	0010 1110
,	0010 0111
:	0011 1010
;	0011 1011

space	0010 0000

A	0100 0001
B	0100 0010
C	0100 0011
D	0100 0100
E	0100 0101
F	0100 0110
G	0100 0111
H	0100 1000
I	0100 1001
J	0100 1010
K	0100 1011
L	0100 1100
M	0100 1101
N	0100 1110

O	0100 1111
P	0101 0000
Q	0101 0001
R	0101 0010
S	0101 0011
T	0101 0100
U	0101 0101
V	0101 0110
W	0101 0111
X	0101 1000
Y	0101 1001
Z	0101 1010

?	0011 1111
!	0010 0001
'	0010 1100

a	0110 0001
b	0110 0010
c	0110 0011
d	0110 0100
e	0110 0101
f	0110 0110
g	0110 0111
h	0110 1000
I	0110 1001
j	0110 1010
k	0110 1011
l	0110 1100

"	0010 0010
(	0010 1000
)	0010 1001

m	0110 1101
n	0110 1110
o	0110 1111
p	0111 0000
q	0111 0001
r	0111 0010
s	0111 0011
t	0111 0100
u	0111 0101
v	0111 0110
w	0111 0111
x	0111 1000
y	0111 1001
z	0111 1010

Example (coding)

Text message: Phy114 quiz today

Message translated into binary:

01010000 01101000 01111001 00110001 00110001 00110100 00100000 01110001 01110101 01101001 01111010 00100000 01110100 01101111 01100100 01100001 01111001

Simplest Algorithm : Add binary code of a number to all letters except space. When adding the same numbers, take the result to be 0, otherwise take it to be 1. For adding two numbers $\boldsymbol{a}$ and $\boldsymbol{b}$, this operation is $(\boldsymbol{a}+\boldsymbol{b})(\boldsymbol{mod}\ 2)$

Key: 7

Contd ……

Original message in binary:

01010000 01101000 01111001 00110001 00110001 00110100
00100000 01110001 01110101 01101001 01111010 00100000
01110100 01101111 01100100 01100001 01111001

Key: 00110111

Final coded message:

01100111 01011111 01001110 00000110 00000110 00000011
00100000 01000111 01000010 01011110 01001101 00100000
01000011 01011000 01010011 01010110 01001110

Text Translation: g_N GB^M CXSVN

Example (decoding)

Coded message:

01100011 01011111 01010110 01010001 01011100 00100000 01001110
01011000 01000010 00100000 01010001 01011000 01000101 00100000
01000100 01010110 01011010 01011000 01000100 01010110 01100100

Text translation: c_VQ\ NXB QXE DVZXDVd

Key: 7 (00110111)

Deciphered code:

01010100 01101000 01100001 01101110 01101011 00100000 01111001
01101111 01110101 00100000 01100110 01101111 01110010 00100000
01110011 01100001 01101101 01101111 01110011 01100001 01110011

Text translation:

Thank you for samosas

Problem of key distribution

- It is clear that the message sender **Anubha** (called Alice in books), a name given by Prof. Manindra Agrawal (MA), CSE, IITK and receiver **Braj** (called Bob in literature), the name given by MA, must agree on a key that they use to encrypt and decrypt a message. In the examples discussed, the key was number 7.
- To exchange the key, either **Anu** (we use this for Anubha) and **Braj** meet personally or they communicate through a messenger or a communication channel, such as a phone or an e-mail.
- If **Anu** and **Braj** meet personally to agree on a key, it is safe. However, if they have to communicate through some channel, there is a possibility that an eavesdropper **Ela** (Eve in literature), again a name suggested by MA, will intercept the key and therefore be able to decipher the message herself.
- So a challenge in cryptography is: **How to distribute the key securely?**

How to distribute a key securely

Reference: *The Code book ……….* by Simon Singh

Idea: Use hard-to-invert functions to generate a key. Functions based on modular mathematics are such functions. These are known as **one-way-functions**.

Algorithm:

1. Anu and Braj agree on two numbers Y and P with $Y < P$. These numbers are exchanged openly.
2. Anu chooses a number A and Braj chooses a number B. A is known only to Anu and B only to Braj.
3. Anu calculates $\alpha = Y^A (mod\ P)$ and Braj calculates $\beta = Y^B (mod\ P)$.
4. Anu and Braj exchange α and β openly. Even if Ela knows these numbers, she cannot figure out A and B easily as is self-evident.
5. Anu calculates $\beta^A (mod\ P)$ and Braj calculates $\alpha^B (mod\ P)$. They both get the same number (its proof is simple) and this number is used as the key.

An example

$\boldsymbol{Y = 5} \qquad \boldsymbol{P = 13}$

Anu	Braj
$A = 2$	$B = 3$
$\alpha = Y^A (mod\ P)$ $\alpha = 25(mod\ 13) = 12$	$\beta = Y^B (mod\ P)$ $\beta = 125(mod\ 13) = 8$

Key $= \beta^A (mod\ P)$
Key $= 64(mod\ 13) = 12$
	Key$= \alpha^B (mod\ P)$
	Key$= 1728(mod\ 13) = 12$

Binary key = 0011000100110010

RSA (Rivest-Shamir-Adleman) encryption

Using modular mathematics, Rivest, Shamir and Adleman developed a code, known as RSA, which is asymmetric in that Anu and Braj both have their own encryption and decryption keys which are different. While encryption key is public knowledge, decryption key is not and is specific to each person. An advantage of this code is that now Anu and Braj do not have to exchange their keys to decode the messages. If a message is to be sent to Anu, it is coded using her encryption code and can be decoded by her using her decryption keys.

What have we learnt? and the future!

- In modern cryptography the algorithm is known publicly. It is the key that is a secret and is used to encode a message.
- A communication is safe if the key is formed by using one-way-functions.
- In using one-way-functions, Anu and Braj exchange some information publicly and that information is used to form a common key.
- Need to exchange the information every time two persons have to communicate secretly is avoided by having asymmetric codes in which each person has their own encryption and decryption codes.
- **A warning:** with extremely fast parallel computers, based on principles of quantum mechanics and therefore known as **quantum computers**, the codes described so far will be broken swiftly by inverting the one-way-functions efficiently. The way to beat that will be having a **quantum encryption key** that we now discuss.

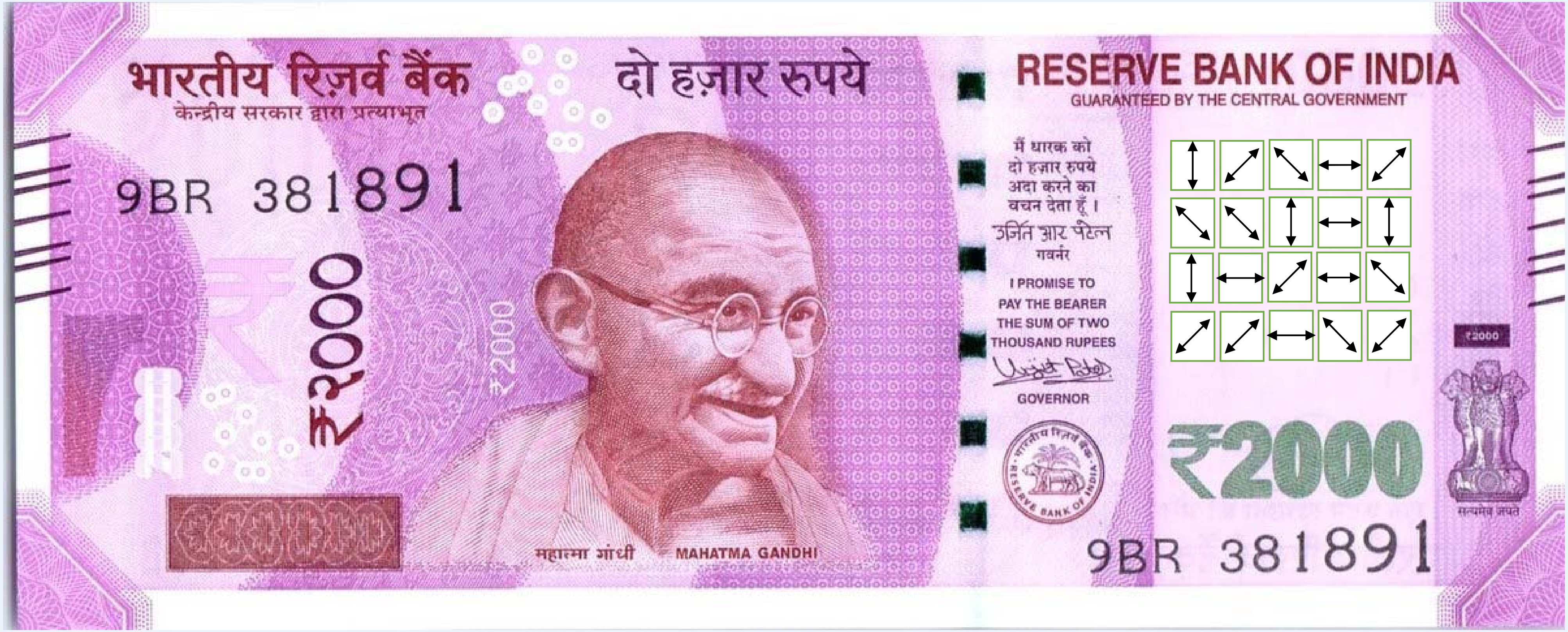
Background – quantum currency notes

- Idea developed by Stephen Wiesner (Stephen Wiesner – Wikipedia).
- In addition to a serial number, a currency note is to have 20 light traps each with one polaroid filter.
- Use polaroid filters with the following four orientations in different sequences in each note

  ↕ ↔ ⤢ ⤡

- Bank keeps a list of serial numbers and the sequence of polarizers.
- These currency notes are **impossible** to counterfeit.

A quantum 2000 rupees note



How counterfeiting quantum money is impossible!

A person who wants to make a fake currency note should copy both its serial number and the polarization sequence.

Serial number can be duplicated without any problem.

To measure polarization, let the person uses a filter with ↕ polarization.

The person will then read polarization to be ↕ even with ⤢ and ⤡ boxes 50% of the times.

Similarly if ⤢ filter is used , polarization in boxes ↕ and ↔ will be read **incorrectly** half the time.

This makes it impossible for a person to get the sequence of polarization correctly.

Quantum key distribution

The idea of using photons with four different polarizations was extended and developed further by Bennett and Brassard to generate and distribute an encoding key that (i) is secure, and (ii) lets the two persons generating the key know that an eavesdropper is making attempts to listen to their conversation.

Using polarization for generating a key and sharing it BB84 (Bennett and Brassard 1984) protocol

- Anu sends series of bits (0 and 1) chosen at random to Braj.
- Anu, however, uses two different polarization schemes to represent the bits.
- In + scheme, polarization state ↕ represents 1 and ⟷ is 0.
- In x scheme ⤢ is 1 and ⤡ represents 0.
- An example of sequence sent by Anu and the corresponding schemes and polarization is as follows

1	0	0	1	1	1	0	0	1	0	1	1	0	1	1	0
x	+	+	x	+	+	+	x	x	x	+	+	x	+	+	x
⤢	⟷	⟷	⤢	↕	↕	⟷	⤡	⤢	⤡	↕	↕	⤡	↕	↕	⤡

- Braj now detects the series of photons sent by Anu using his set of polarizer filters (+ or ×) and gets readings as

	1	0	0	1	1	1	0	0	1	0	1	1	0	1	1	0
Anu	x	+	+	x	+	+	+	x	x	x	+	+	x	+	+	x
	⤢	↔	↔	⤢	↕	↕	↔	⤡	⤢	⤡	↕	↕	⤡	↕	↕	⤡
Braj	+	+	x	x	+	x	+	+	x	x	+	x	x	+	+	x
	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0

- Anu and Braj compare their schemes openly and reject bits corresponding to those that do not match. The remaining bits make the key

- 0 - 1 1 - 0 - 1 0 1 - 0 1 1 0

Key: 01101010110

What is the result of the exercise carried out by Anu and Braj?

- Suppose Anu and Braj had $N = 2000$ random polarizers $+$ and $\times$ to start their exercise.
- On the average $\frac{N}{2} = 1000$ will match. Thus, after they finish their exercise, they have a string of 1000 qubits $(0 \text{ and } 1)$ and the corresponding scheme (polarizer filters) of generating them.
- These qubits can form the key as concluded in the previous slide.
- However, to be absolutely safe, Anu and Braj would now like to know if Ela was intercepting their messages.
- For this, Anu publicly tells Braj the qubits $(0 \text{ and } 1)$ used for $\frac{N}{4} = 500$ filters.

Why Ela cannot get the correct key?

- Ela also uses polarization filters (+ or ×) put in a sequence randomly like Braj and detects polarization of photons sent by Anu.
- After Ela listens to the conversation between Anu and Braj comparing their sequence of filters, she also keeps the bits corresponding to filters which match for Anu and Braj, whether her filter matched with theirs or not.
- However, if her filter did not match, her reading of polarization will be correct only half the time because the corresponding photon's polarization can collapse to any one of the state with 50% probability. **This is how laws of quantum mechanics make the key safe**.
- For example, if Anu and Braj use + polarizers and Ela an × polarizer then for photon polarization ↕ or ⟷, probability of it being read as ⤢ or ⤡ is 0.5 for each case. So Ela is equally likely to mark the bit as 0 or 1. Thus on the average, she will get only half of these right and her key will not be correct.

How is presence of Ela detected?
And what do Anu and Braj do if Ela is listening?

- After having made their key, Anu tells Braj a small fraction of bits she had sent. We took these to be $\frac{N}{4} = 500$.
- If Ela was eavesdropping while key was being established, 50% of these bits (250) on the average would have been altered. Of these $50\%\,(125)$ will be read incorrectly by Braj. As a consequence $25\%\,(125)$ readings for Braj will be wrong. This would mean that Ela was eavesdropping.
- At this stage, Anu and Braj can start the whole key setting operation again to get a new key or use some other means to communicate.

Summary of BB-84 protocol

Adapted from: *The Code Book ……..* by Simon Singh

Another reference: Quantum Computing for Everyone by Chris Bernhardt

1. Anu sends Braj a series of photons.
2. Anu tells Braj on which occasions their polarizing schemes matched. Result of what was sent or measured is not exchanged.
3. They discard the polarization schemes that are not compatible. The remaining bits can make up the key.
4. Anu and Braj check the integrity of their key by testing the correctness of a few of their bits that Anu tell Braj openly.
5. If the verification process is satisfactory, they can use the key to encrypt a message. If errors occur during the verification, they know that Ela was eavesdropping and they should start the process of establishing the key again.