Cryptography - Part 4

Mar. 20, 2025

Recap question:

March 20, 2025

Bob has come up with two pseudo-random number generators to give random numbers between 1 and 6. When he tests the first one, it gives the output

1,1,1,1,1,1,1,1,1,1,1,1,…

When he tests the second one, it gives the output

1,2,3,1,2,3,1,2,3,1,2,3,1,…

Are these good random number generators? Why/why not?

Recap question:

March 20, 2025

Both the number generators show very strong patterns so they do not seem random at all! It is easy to predict what the next generated “random” number will be, so they are both very bad random number generators.

Cryptography - Part 4

March 20, 2025

By the end of this lecture, you will be able to:

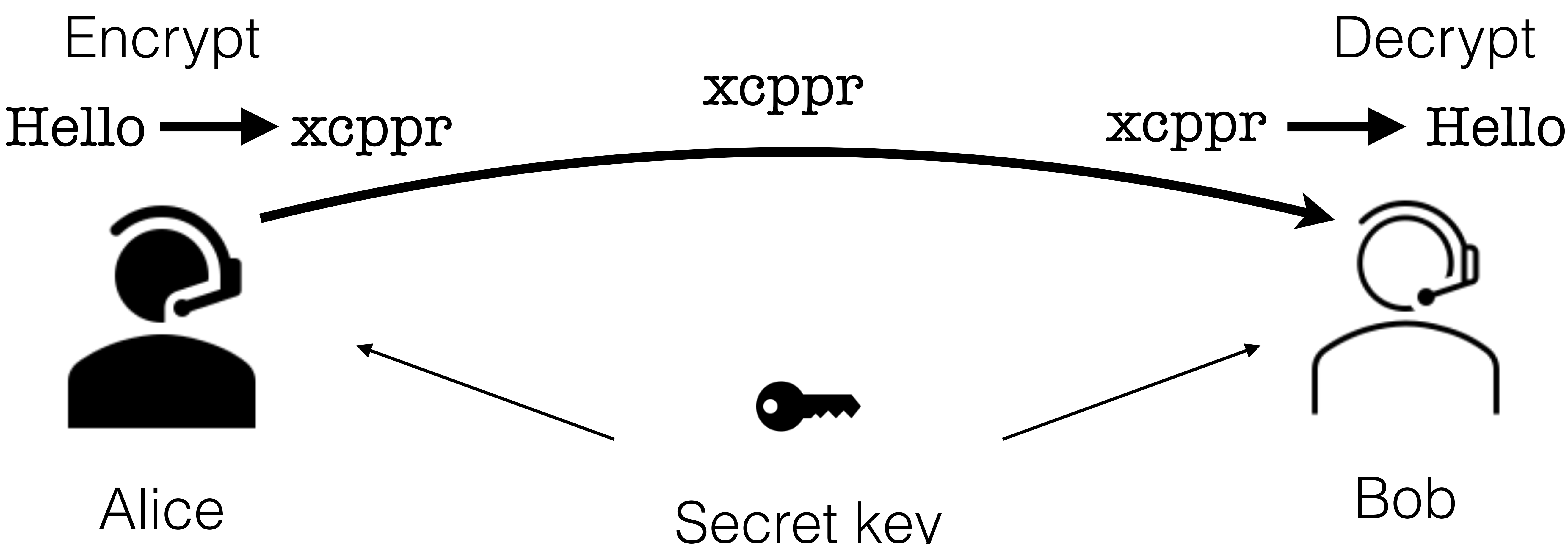
1. Define symmetric-key encryption and asymmetric-key encryption
2. Compute with modular arithmetic
3. Perform RSA-encryption

Symmetric-key encryption

Symmetric-key encryption is a type of encryption that uses the same key to encrypt and decrypt data. Both the sender and the recipient have identical copies of the key, which they keep secret and do not share with anyone.

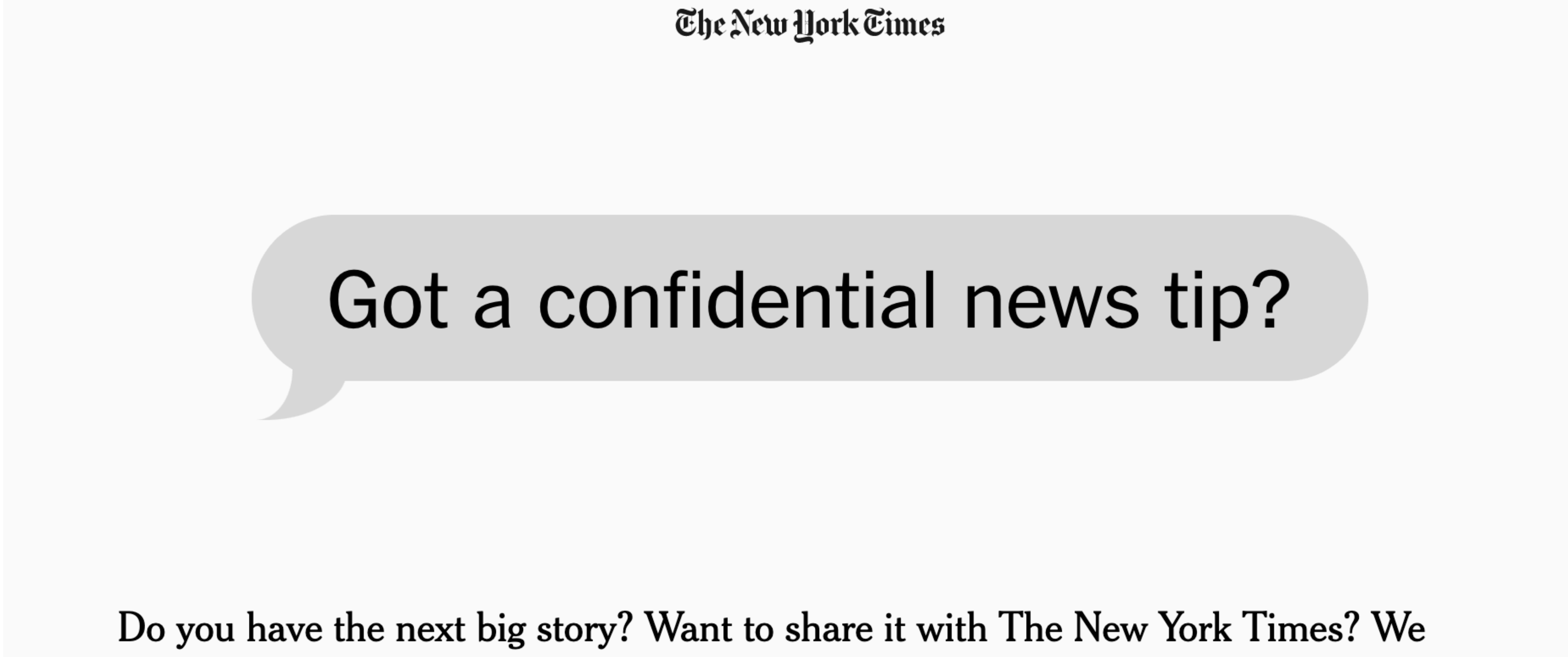
Symmetric-key encryption

Symmetric-key encryption is a type of encryption that uses the same key to encrypt and decrypt data. Both the sender and the recipient have identical copies of the key, which they keep secret and do not share with anyone.

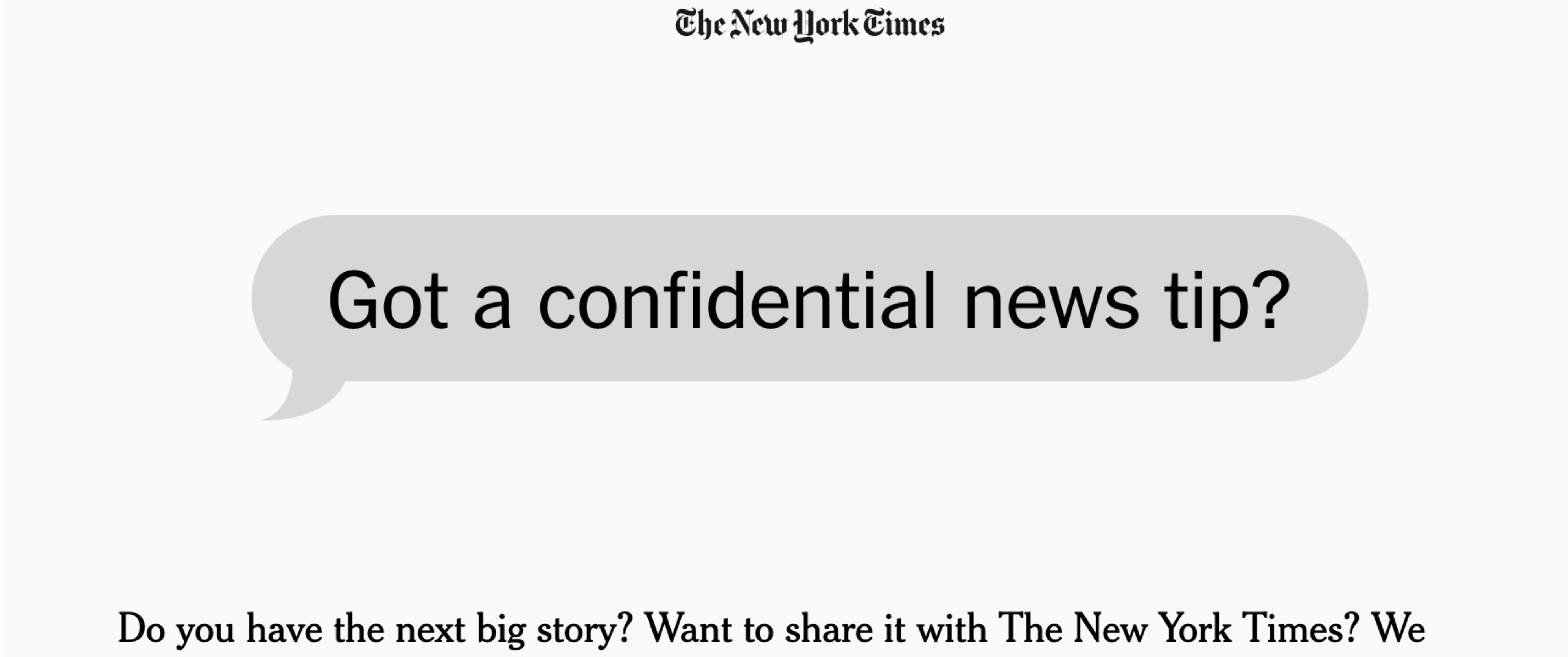


Examples

1. Substitution ciphers
2. One-time pads
3. Hagelin/Enigma machines
4. Banking: encrypting credit card information or other personally identifiable information required for transactions.
5. Data storage: encrypting data stored on a device when that data is not being transferred, for instance, Microsoft Azure uses symmetric encryption to encrypt and decrypt large quantities of data quickly.

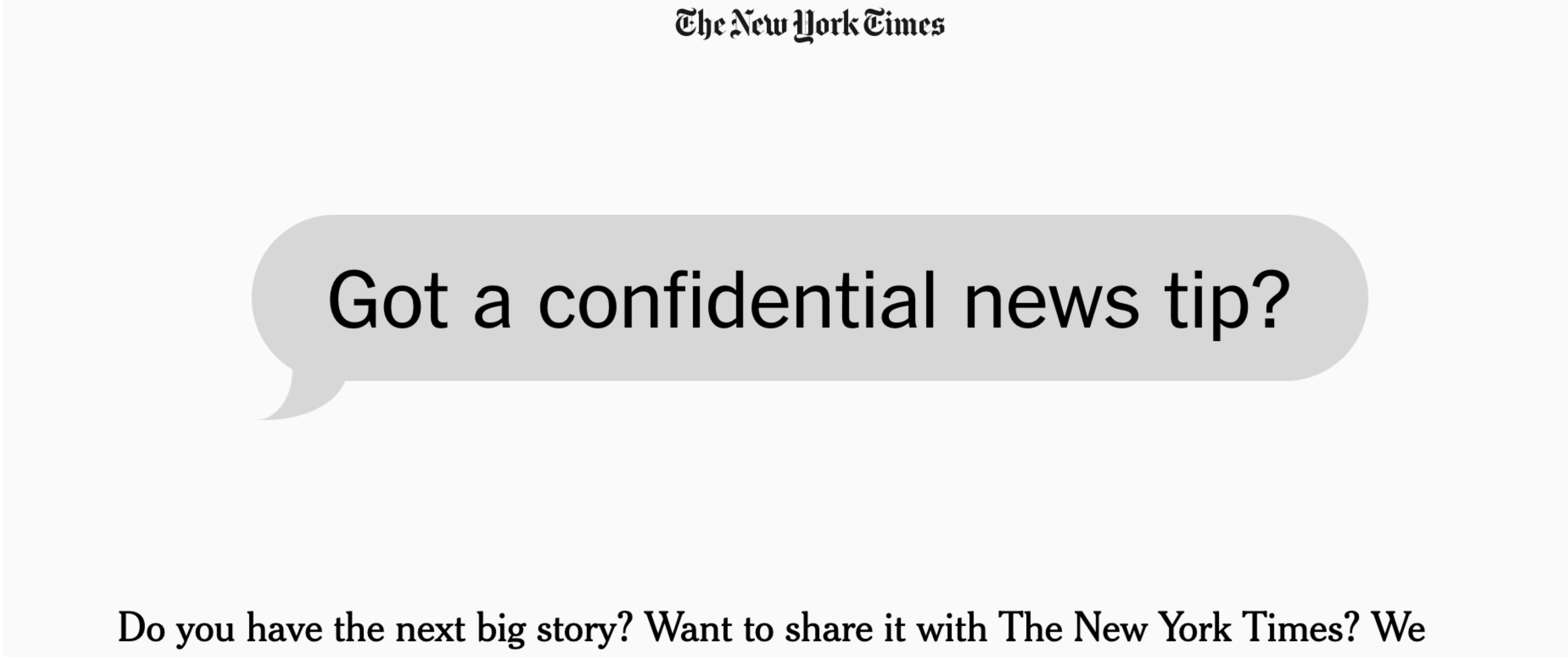


Example 1: Imagine that you are a journalist who wants tips from whistleblowers whom you have never met and who live far away.



Example 1: Imagine that you are a journalist who wants tips from whistleblowers whom you have never met and who live far away.

Example 2: WhatsApp (and other messenger apps) use encryption to secure messages between phones on different continents.



Example 1: Imagine that you are a journalist who wants tips from whistleblowers whom you have never met and who live far away.

Example 2: WhatsApp (and other messenger apps) use encryption to secure messages between phones on different continents.

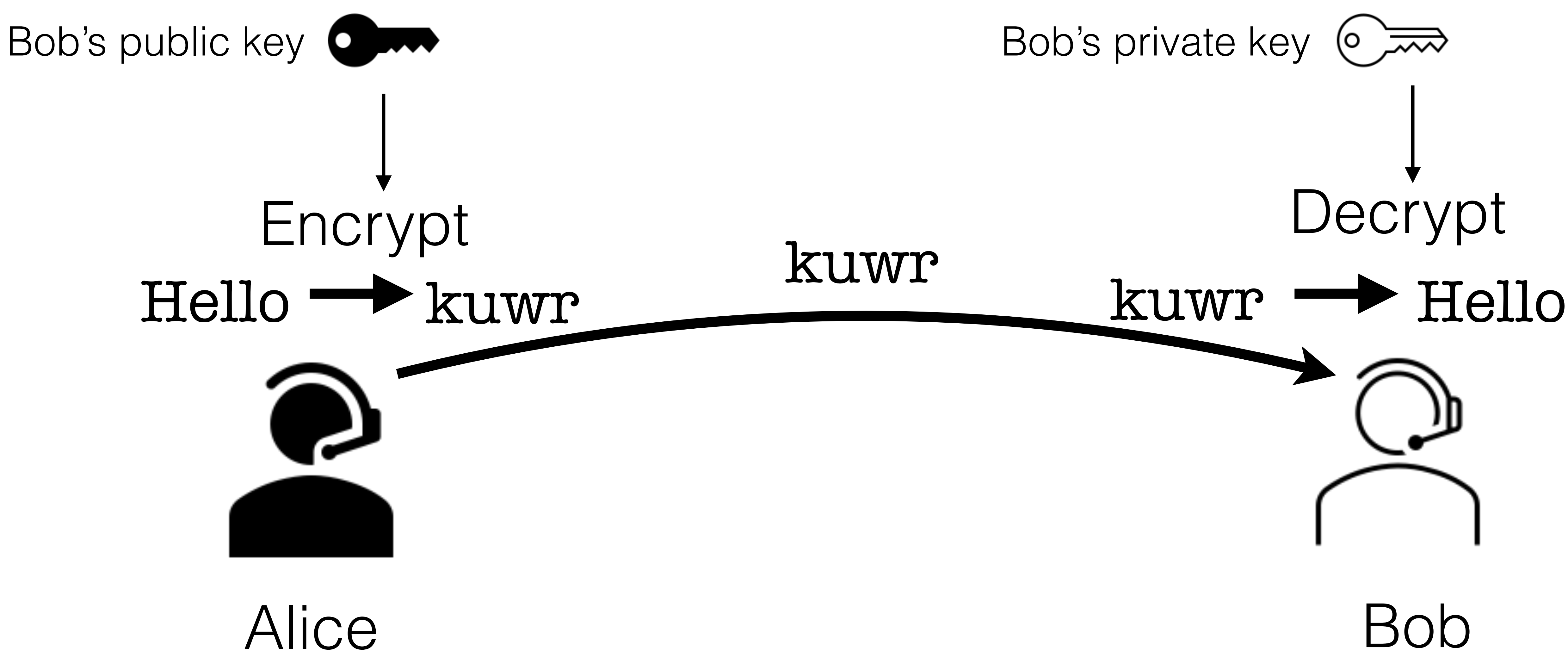
Could these scenarios use any of the encryption schemes we have talked about so far?

Problems with symmetric-key encryption:

1. How does the journalist give the key to the whistleblower? She cannot send it over an insecure channel, because if the key is intercepted, the encrypted messages can be read.
2. The journalist will need to keep track of one key per whistleblower. If she would use the same key for everyone, the different whistleblowers could read each other’s messages.

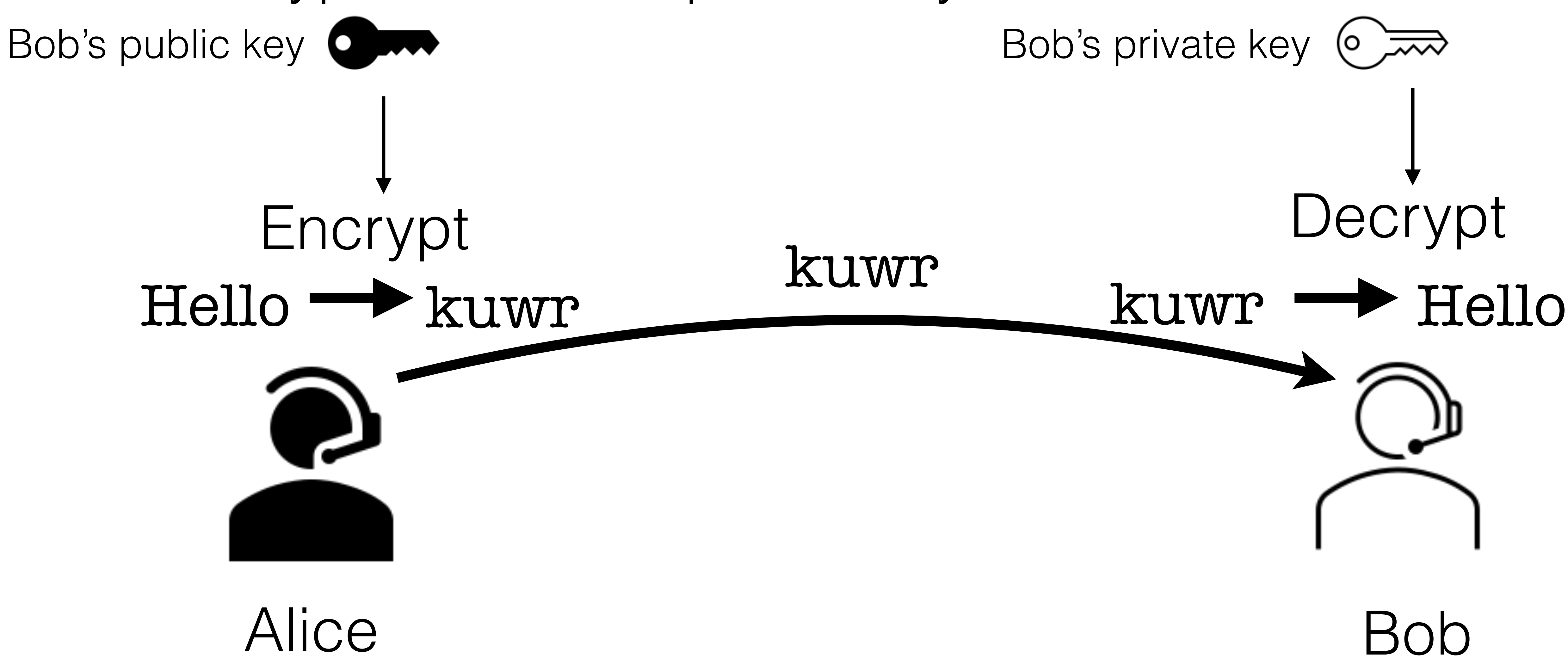
Asymmetric-key encryption

With **asymmetric-key encryption**, the encryption procedure can be made public without compromising security: knowing how to encrypt does not enable you to decrypt for these public key systems.



Asymmetric-key encryption

1. Bob displays his public key on his website.
2. Anyone wishing to send him a message encrypts the message with the public key.
3. Bob keeps his private key secret. Using the private key is the only way to decrypt a message that is encrypted with the public key.



Symmetric-key encryption

Asymmetric-key encryption

Symmetric-key encryption

Asymmetric-key encryption

Question: how do we implement this mathematically?

ASCII Table
(American standard code for information interchange)

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	∆	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	“	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	”	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	‘	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	’	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	⁄	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	˚
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	ﬁ	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	ﬂ	255	ˇ

Modular arithmetic

Mathematical notation to make it easier to talk about divisors and remainders when performing integer division.

Example: Integer division of 9 by 4.

$$9 = 2 \cdot 4 + 1$$

Modular arithmetic

Mathematical notation to make it easier to talk about divisors and remainders when performing integer division.

Example: Integer division of 9 by 4.

$$9 = 2 \cdot 4 + 1$$

dividend

divisor

Modular arithmetic

Mathematical notation to make it easier to talk about divisors and remainders when performing integer division.

Example: Integer division of 9 by 4.

$$9 = 2 \cdot 4 + 1$$

dividend

quotient

divisor

remainder

Modular arithmetic

Mathematical notation to make it easier to talk about divisors and remainders when performing integer division.

Example: Integer division of 9 by 4.

$$9 = 2 \cdot 4 + 1$$

We introduce new symbols and write this as

$$9 \equiv 1 \pmod 4$$

Modular arithmetic

Mathematical notation to make it easier to talk about divisors and remainders when performing integer division.

Example: Integer division of 9 by 4.

$$9 = 2 \cdot 4 + 1$$

We introduce new symbols and write this as

$$9 \equiv 1 \pmod 4$$

dividend

remainder

divisor

and the quotient is not written out

Modular arithmetic

Mathematical notation to make it easier to talk about divisors and remainders when performing integer division.

Example: Integer division of 9 by 4.

$$9 = 2 \cdot 4 + 1$$

We introduce new symbols and write this as

$$9 \equiv 1 \pmod 4$$

"9 is equivalent to 1 modulo 4"

Modular arithmetic

Mathematical notation to make it easier to talk about divisors and remainders when performing integer division.

Example: Integer division of 9 by 4.

$$9 = 2 \cdot 4 + 1$$

We introduce new symbols and write this as

$$9 \equiv 1 \pmod{4}$$

"9 and 1 differ by a multiple of 4"

Definition: the number a is equivalent to the number b modulo n, or in shorthand notation

$$a \equiv b \pmod{n}$$

if a differs from b by a multiple of n. This can be written in either order, so we also write

$$b \equiv a \pmod{n}$$

Definition: the number a is equivalent to the number b modulo n, or in shorthand notation

$$a \equiv b \pmod{n}$$

if a differs from b by a multiple of n. This can be written in either order, so we also write

$$b \equiv a \pmod{n}$$

Example:

What is $35 \pmod{10}$?

Definition: the number a is equivalent to the number b modulo n, or in shorthand notation

$$a \equiv b \pmod{n}$$

if a differs from b by a multiple of n. This can be written in either order, so we also write

$$b \equiv a \pmod{n}$$

Example:

What is $35 \pmod{10}$? The remainder when dividing 35 by 10 is 5 so

$$35 \pmod{10} \equiv 5$$

Definition: the number a is equivalent to the number b modulo n, or in shorthand notation

$$a \equiv b \pmod{n}$$

if a differs from b by a multiple of n. This can be written in either order, so we also write

$$b \equiv a \pmod{n}$$

Example:

What is $35 \pmod{10}$? The remainder when dividing 35 by 10 is 5 so

$$35 \pmod{10} \equiv 5$$

We can also write $5 \pmod{10} \equiv 35$

Modular arithmetic

Examples:

1. Integer division of 6 by 2.

$$6 = 3 \cdot 2 + 0$$

so

$$6 \equiv 0 \pmod{2}$$

Modular arithmetic

Examples:

1. Integer division of 6 by 2.

$$6 = 3 \cdot 2 + 0$$

so

$$6 \equiv 0 \pmod 2$$

2. Integer division of 14 by 6

$$14 = 2 \cdot 6 + 2$$

so

$$2 \equiv 14 \pmod 6$$

RSA Algorithm
(1977, by Rivest, Shamir, Adleman)

Let x be a message that has been converted to numbers. Given the pair of values n and r (the public key), the encrypted message y is given by

$$y \equiv x^r \pmod{n}$$

Given the value of s (the private key), we can retrieve x by

$$x \equiv y^s \pmod{n}$$

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

Example: We have the keys $n = 15, r = 3, s = 3$ and encrypt $x = 2$ by computing:

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

Example: We have the keys $n = 15, r = 3, s = 3$ and encrypt $x = 2$ by computing:

$$y \equiv 2^3 \pmod{15} \equiv 8 \pmod{15}$$

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

Example: We have the keys $n = 15, r = 3, s = 3$ and encrypt $x = 2$ by computing:

$$y \equiv 2^3 \pmod{15} \equiv 8 \pmod{15}$$

so the encrypted message is $y = 8$.

To decrypt, we compute

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

Example: We have the keys $n = 15, r = 3, s = 3$ and encrypt $x = 2$ by computing:

$$y \equiv 2^3 \pmod{15} \equiv 8 \pmod{15}$$

so the encrypted message is $y = 8$.

To decrypt, we compute

$$x \equiv 8^3 \pmod{15} \equiv 512 \pmod{15} \equiv 2 \pmod{15}$$

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

Example: We have the keys $n = 15, r = 3, s = 3$ and encrypt $x = 2$ by computing:

$$y \equiv 2^3 \pmod{15} \equiv 8 \pmod{15}$$

so the encrypted message is $y = 8$.

To decrypt, we compute

$$x \equiv 8^3 \pmod{15} \equiv 512 \pmod{15} \equiv 2 \pmod{15}$$

This gives $x = 2$ which is the message, so it works!

How to construct the keys r,n and s?

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

In contrast to what we have done before, the keys should not be chosen randomly! r,n and s are chosen by the following procedure:

How to construct the keys r,n and s?

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

In contrast to what we have done before, the keys should not be chosen randomly! r,n and s are chosen by the following procedure:

1. Choose two prime numbers p and q

How to construct the keys r,n and s?

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

In contrast to what we have done before, the keys should not be chosen randomly! r,n and s are chosen by the following procedure:

1. Choose two prime numbers p and q
2. Choose $n = p \times q$

How to construct the keys r,n and s?

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

In contrast to what we have done before, the keys should not be chosen randomly! r,n and s are chosen by the following procedure:

1. Choose two prime numbers p and q
2. Choose $n = p \times q$
3. Choose r as any number that has no prime factors in common with $(p-1) \times (q-1)$

How to construct the keys r,n and s?

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

In contrast to what we have done before, the keys should not be chosen randomly! r,n and s are chosen by the following procedure:

1. Choose two prime numbers p and q
2. Choose $n = p \times q$
3. Choose r as any number that has no prime factors in common with $(p-1) \times (q-1)$
4. Compute s from p and q by the Euclidean algorithm

Prime numbers

Prime numbers: a positive integer is a prime number if the only integers that divide it (with no remainder) is itself and 1.

Prime numbers

Prime numbers: a positive integer is a prime number if the only integers that divide it (with no remainder) is itself and 1.

Example: 2 is a prime number
3 is a prime number
4 is not a prime number
5 is a prime number
⋮

Prime factorization

Prime factorization: every positive integer has a unique factorization into prime numbers

Prime factorization

Prime factorization: every positive integer has a unique factorization into prime numbers

Examples: 77 = 7 x 11, 120 = 2 x 2 x 2 x 3 x 5

How to construct the keys r,n and s?

Encryption: $y \equiv x^r \pmod{n}$

Decryption: $x \equiv y^s \pmod{n}$

In contrast to what we have done before, the keys should not be chosen randomly! r,n and s are chosen by the following procedure:

1. Choose two prime numbers p and q
2. Choose $n = p \times q$
3. Choose r as any number that has no prime factors in common with $(p-1) \times (q-1)$
4. Compute s from p and q by the Euclidean algorithm

How to construct the keys r,n and s?

Example:

1. Choose $p = 3$ and $q = 5$

How to construct the keys r,n and s?

Example:

1. Choose $p = 3$ and $q = 5$
2. Choose $n = p \times q = 15$

How to construct the keys r,n and s?

Example:

1. Choose $p = 3$ and $q = 5$
2. Choose $n = p \times q = 15$
3. Choose r as any number that has no prime factors in common with $(p-1) \times (q-1) = 8$.

How to construct the keys r,n and s?

Example:

1. Choose $p = 3$ and $q = 5$
2. Choose $n = p \times q = 15$
3. Choose r as any number that has no prime factors in common with $(p-1) \times (q-1) = 8$.

   8 has the prime factorization $8 = 2 \times 2 \times 2$ so its only prime factor is 2. We can therefore choose e.g. $r = 3$

How to construct the keys r,n and s?

Example:

1. Choose $p = 3$ and $q = 5$
2. Choose $n = p \times q = 15$
3. Choose r as any number that has no prime factors in common with $(p-1) \times (q-1) = 8$.

   8 has the prime factorization $8 = 2 \times 2 \times 2$ so its only prime factor is 2. We can therefore choose e.g. $r = 3$
4. Compute s from p and q by the Euclidean algorithm

Key generation in RSA requires two distinct primes p and q. To give secure encryption, they should be large primes (several hundred digits).

Key generation in RSA requires two distinct primes p and q. To give secure encryption, they should be large primes (several hundred digits).

Why does this matter?

Part of the public key is $n = p \times q$ which is the prime factorization of n. Anyone who knows n can try to compute this prime factorization. If they could do it, they would get their hands on p and q which lets them compute the secret key s, breaking the encryption.

Key generation in RSA requires two distinct primes p and q. To give secure encryption, they should be large primes (several hundred digits).

Why does this matter?

Part of the public key is $n = p \times q$ which is the prime factorization of n. Anyone who knows n can try to compute this prime factorization. If they could do it, they would get their hands on p and q which lets them compute the secret key s, breaking the encryption.

Choosing the primes large enough makes it computationally infeasible to find p and q from knowledge of n

RSA-250 challenge

In 2020, a team of mathematicians made the headlines (in the science section) because they factored a number with 250 digits. The factorization took 2700 CPU years (performed on many computers).

```
21403246502407449612644230728393335630
08614715144755017797754920881418023447
14013664334551909580467961099285187247
09145876873962619215573630474547705208
05119056493106687691590019759405693457
45223058932597669747168173806936489469
9871578494975937497937
                =
64135289477071580278790190170577389084
82501474294344720811685963202453234463
02386235987526683477087376619255856946
39798853367
                x
33372027594978156556226010605355114227
94076034476755466678452098702384172921
00370802574486732968818775657189862580
36932062711
```

In practical RSA schemes, n has 400 or more digits.

Some modular arithmetic acrobatics

If $\quad a \equiv b \pmod{n}$

$\quad c \equiv d \pmod{n}$

then $\quad a + c \equiv b + d \pmod{n}$

and $\quad ac \equiv bd \pmod{n}$

If we need to make multiple modular calculations, we can simplify them after each step, so that we won't need to multiply or add numbers bigger than $n-1$. This process of replacing a number with the remainder you get when you divide it by n is called **reduction modulo** n.

Examples:

$$321 \times 714 \equiv 4 \pmod 5$$

$$321 \times 714 \equiv 0 \pmod 7$$

$$321 \times 715 \equiv 6 \pmod 7$$

$$715^3 = 715 \times 715 \times 715 \equiv 1 \pmod 7$$

$$715^{984} \equiv 1 \pmod 7$$

$$321^3 \equiv 6 \pmod 7$$

$$321^{984} \equiv 6^{984} \equiv (-1)^{984} \equiv 1 \pmod 7$$

How about $320^{984} \pmod{7}$?

$$320^{984} \equiv 5^{984} \pmod{7}$$

But 5^{984} is still a large number

Start by writing 984 as the sum of powers of 2

$$\begin{aligned} 984 &= 512 + 256 + 128 + 64 + 16 + 8 \\ &= 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 2^3 \end{aligned}$$

$$5^{984} = 5^{512} \times 5^{256} \times 5^{128} \times 5^{64} \times 5^{16} \times 5^{8}$$

$$5^2 = 25 \equiv 4 \pmod 7$$

$$5^4 = 5^2 \times 5^2 \equiv 4 \times 4 \equiv 2 \pmod 7$$

$$5^8 = 5^4 \times 5^4 \equiv 2 \times 2 \equiv 4 \pmod 7$$

$$5^{16} = 5^8 \times 5^8 \equiv 4 \times 4 \equiv 2 \pmod 7$$

$$5^{32} \equiv 4 \pmod 7 \qquad 5^{64} \equiv 2 \pmod 7$$

$$5^{128} \equiv 4 \pmod 7 \qquad 5^{256} \equiv 2 \pmod 7$$

$$5^{512} \equiv 4 \pmod 7$$

$$
\begin{aligned}
5^{984} &\equiv 5^{512} \times 5^{256} \times 5^{128} \times 5^{64} \times 5^{16} \times 5^{8} \\
&\equiv 4 \times 2 \times 4 \times 2 \times 2 \times 4 \\
&\equiv 8^{3} \equiv 1 \pmod{7}
\end{aligned}
$$