Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 4, part 1

One-time Pad

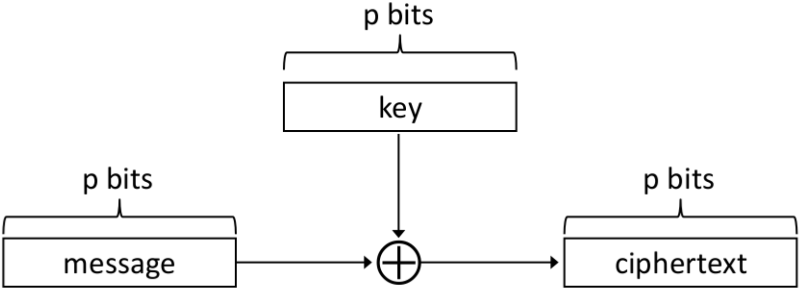
One-time Pad (OTP)

- Patented in **1917** by Vernam
- Invented (at least) **35** years earlier
- Proven **perfectly secret** by Shannon (**1949**)

One-time Pad

- Let $\mathcal{M} = \{0,1\}^n$
- Gen: choose a uniform key $k \in \{0,1\}^n$
- $\mathsf{Enc}_k(m) = k \oplus m$
- $\mathsf{Dec}_k(c) = k \oplus c$
- $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = m$

One-time Pad



Perfect Secrecy of One-time Pad

Theorem

The One-time Pad satisfies perfect secrecy.

Intuition

- Any observed ciphertext can correspond to any message
  - (This is necessary, but not sufficient, for perfect secrecy)
- Having observed a ciphertext, the attacker cannot conclude for certain which message was sent

Perfect Secrecy of One-time Pad

Proof.

- Fix arbitrary distribution over $\boldsymbol{\mathcal{M}} = \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$, and choose arbitrary $\boldsymbol{m}, \boldsymbol{c} \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$
- Check if

$$\mathbf{Pr}[\boldsymbol{M} = \boldsymbol{m} | \boldsymbol{C} = \boldsymbol{c}] = \mathbf{Pr}[\boldsymbol{M} = \boldsymbol{m}]$$

Perfect Secrecy of One-time Pad

Proof.

- Recall (Bayes' theorem)

$$\mathbf{Pr}[\boldsymbol{M = m | C = c}] = \frac{\mathbf{Pr}[\boldsymbol{C = c | M = m}]\ \mathbf{Pr}[\boldsymbol{M = m}]}{\mathbf{Pr}[\boldsymbol{C = c}]}$$

- We can see that $\forall \boldsymbol{c, m}$

$$\mathbf{Pr}[\boldsymbol{C = c | M = m}] = \mathbf{Pr}[\boldsymbol{M \oplus K = c | M = m}] =$$
$$= \mathbf{Pr}[\boldsymbol{m \oplus K = c}] = \mathbf{Pr}[\boldsymbol{K = c \oplus m}] = \boldsymbol{2^{-n}}$$

Perfect Secrecy of One-time Pad

Proof.

By law of total probability:

$$
\begin{aligned}
&\mathbf{Pr}[C = c] = \\
&= \sum_{m'} \mathbf{Pr}[C = c | M = m'] \, \mathbf{Pr}[M = m'] \\
&= \sum_{m'} \mathbf{Pr}[K = m' \oplus c | M = m'] \, \mathbf{Pr}[M = m'] \\
&= \sum_{m'} 2^{-n} \, \mathbf{Pr}[M = m'] \\
&= 2^{-n} \sum_{m'} \mathbf{Pr}[M = m'] = 2^{-n}
\end{aligned}
$$

Perfect Secrecy of One-time Pad

Proof.

$$\begin{aligned}
&\mathbf{Pr}[M = m | C = c] = \\
&= \frac{\mathbf{Pr}[C = c | M = m]\,\mathbf{Pr}[M = m]}{\mathbf{Pr}[C = c]} \\
&= \frac{2^{-n}\,\mathbf{Pr}[M = m]}{2^{-n}} \\
&= \mathbf{Pr}[M = m]
\end{aligned}$$

□

One-time Pad and Brute-force Attacks

The same ciphertext		Decrypted with this key...		...gives this plaintext
SMAIJIZJSIFPSTWFI	→	STHIHYZQRRBPIOWNP	→	ATTACKATBREAKFAST
	→	BIHRFIGIODRYOGIRV	→	RETREATBEFORENOON
	→	MYARVOMGKVDHBRLBQ	→	GOAROUNDINCIRCLES
	→	ATAVGOGQORURAAOUX	→	STANDUTTERLYSTILL
	→	AENCQMLCSTQRAFJZQ	→	SINGTWOHAPPYSONGS
	→	AFMOQIHYEOCPAEINQ	→	SHOUTASLOUDASPOSS
	→	IIWTQUGJHXHXQMDLW	→	KEEPTOTALLYSCHTUM
	→	SBPUPPKPZTRXALVUE	→	ALLOUTPUTPOSSIBLE

- OTP resists even a brute-force attack
- Decrypt a ciphertext with every key returns every possible plaintext (incl. every ASCII/English string)
- No way of telling the correct plaintext

Image credit: https://nakedsecurity.sophos.com

One-time Pad

- The One-time Pad achieves perfect secrecy!
- Resists even a brute-force attack
- One-time Pad has historically been used in the real world
- e.g. *red phone* between Washington and Moscow
- Not currently used! Why?

One-time Pad

Limitations of OTP

1. The key is as long as the message
2. A key must be used only once
    - Only secure if each key is used to encrypt a single message
    - (Trivially broken by a known-plaintext attack)

$\Longrightarrow$ Parties must share keys of (total) length equal to the (total) length of all the messages they might ever send

Using the Same Key Twice?

- Say

$$\boldsymbol{c_1 = k \oplus m_1}$$
$$\boldsymbol{c_2 = k \oplus m_2}$$

- Attacker can compute

$$\boldsymbol{c_1 \oplus c_2 = (k \oplus m_1) \oplus (k \oplus m_2) = m_1 \oplus m_2}$$

- This leaks information about $\boldsymbol{m_1, m_2}$

Using the Same Key Twice?

$\boldsymbol{m_1} \oplus \boldsymbol{m_2}$ leaks information about $\boldsymbol{m_1}, \boldsymbol{m_2}$

Is this significant?

- $\boldsymbol{m_1} \oplus \boldsymbol{m_2}$ reveals where $\boldsymbol{m_1}, \boldsymbol{m_2}$ differ
- No longer perfectly secret!
- Exploiting characteristics of ASCII...

ASCII table (recall)

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

https://hubpages.com/technology/What-Are-ASCII-Codes

Using the Same Key Twice: recall ASCII

Observatoins

- Letters begin with `0x4`, `0x5`, `0x6` or `0x7`
  - $\implies$ **letters all begin with 01...**
- ASCII code for the space character `0x20` = **00100000**
  - $\implies$ **the space character begins with 00...**
- `XOR` of two letters gives **00**...
- `XOR` of letter and space gives **01**...
- **Easy to identify `XOR` of letter and space!**

Using the Same Key Twice

- The last byte of $\boldsymbol{c_1} \oplus \boldsymbol{c_2}$ starts with **01**
- Therefore

$$\boldsymbol{c_1} \oplus \boldsymbol{c_2} = \boldsymbol{m_1} \oplus \boldsymbol{m_2} = \boldsymbol{x} \oplus \mathbf{00100000}$$
$$\boldsymbol{x} = \boldsymbol{c_1} \oplus \boldsymbol{c_2} \oplus \mathbf{00100000}$$

- e.g. let $\boldsymbol{c_1} \oplus \boldsymbol{c_2} = \mathbf{01010000}$

$$\boldsymbol{x} = \mathbf{01010000} \oplus \mathbf{00100000}$$
$$\boldsymbol{x} = \mathbf{01110000} = \texttt{0x70} = \text{"}\mathbf{p}\text{"}$$

- Attacker learns one plaintext character: $\boldsymbol{m_1} = \mathbf{p}$ or $\boldsymbol{m_2} = \mathbf{p}$

One-time Pad

Drawbacks

- Key as long the message
- Only secure if each key is used to encrypt once
- Trivially broken by a known-plaintext attack

Note

These limitations are inherent for schemes achieving perfect secrecy

Optimality of the One-time Pad

Theorem

If $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$ *is perfectly secret, then* $|\mathcal{K}| \geq |\mathcal{M}|$.

Intuition

- Given any ciphertext, try decrypting under every possible key in $\mathcal{K}$
- This gives a list of up to $|\mathcal{K}|$ possible messages
- If $|\mathcal{K}| < |\mathcal{M}| \implies$ some message is not on the list

Optimality of the One-time Pad

Proof.

- Assume $|\mathcal{K}| < |\mathcal{M}|$
- Need to show that there is a distribution on $\mathcal{M}$, a message m, and a ciphertext c such that

$$\mathbf{Pr}[M = m | C = c] \neq \mathbf{Pr}[M = m]$$

Optimality of the One-time Pad

Proof.

- Take the uniform distribution on $\mathcal{M}$
- Take any ciphertext c
- Consider the set $M(c) = \{\mathsf{Dec}_k(c)\}_{k \in \mathcal{K}}$
  - the set of messages that could yield the ciphertext c
- $|M(c)| \leq |K| < |M| \implies \exists m$ s.t. $m \notin M(c)$:

$$\mathbf{Pr}[M = m | C = c] = 0 \neq \mathbf{Pr}[M = m]$$

□

Summary

- We defined the notion of **perfect secrecy** (PS)
- We proved that the One-time Pad achieves PS
- We proved that the One-time Pad is optimal (in the key length)
  - i.e. we cannot improve the key length
- Are we done? What about the limitations of OTP?
- Address OTP's limitations by relaxing the definition
  - But in a meaningful way...
- (next slides)

End

References: From Section 2.2 until the end of Chapter 2.