Cryptography 1

Luisa Siniscalchi

(These slides are taken from the lectures of Prof. Jonathan Katz)

Defining Security of Encryption Scheme

If you don't understand what you want to achieve, how can you possibly know when (or if) you have achieved it? (Book: Introduction to Modern Cryptography 2nd ed. CRC Press 2015)

Defining security of encryption scheme

On the need of formal definitions

- What does it mean for a scheme to be secure?
  - What do we want the adversary **to not** be able to achieve?
  - What are the capabilities of the adversary?

Defining security of encryption scheme

On the need of formal definitions

- What does it mean for a scheme to be secure?
  - What do we want the adversary to not be able to achieve?
  - What are the capabilities of the adversary?

Formal definitions help because...

- Definitions enable meaningful analysis, evaluation, and comparison of schemes.

Formal Definition Encryption Scheme ...

Private-key encryption

- $\mathbb{K}$ (key space): set of all possible keys
- $\mathbb{M}$ (message space): set of all possible messages
- $\mathbb{C}$ (ciphertext space): set of all possible ciphertexts

Private-key encryption

A private-key encryption scheme is defined by a message space $\mathbb{M}$, (key space $\mathbb{K}$) and algorithms $\boldsymbol{e}$ and $\boldsymbol{d}$:

- KeyGen (key-generation algorithm): outputs $\boldsymbol{k} \in \mathbb{K}$. Usually: $\boldsymbol{k} \in \mathbb{K}$ uniformly random. (This algorithm is sometimes left implicit in the book)

Private-key encryption

A private-key encryption scheme is defined by a message space $\mathbb{M}$, (key space $\mathbb{K}$) and algorithms $(\mathsf{KeyGen}, \mathsf{e}, \mathsf{d})$:

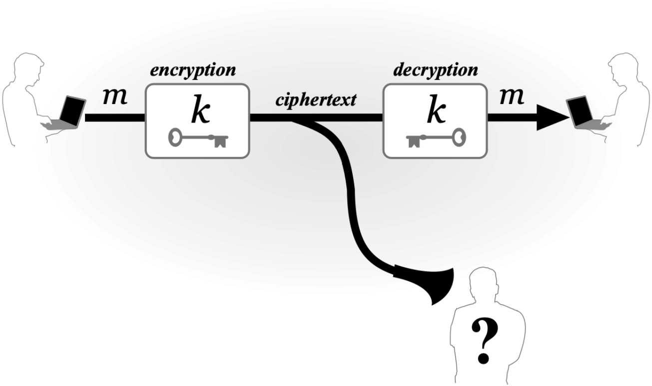
- KeyGen (key-generation algorithm): outputs $\boldsymbol{k} \in \mathbb{K}$. Usually: $\boldsymbol{k} \in \mathbb{K}$ uniformly random. (This algorithm is sometimes left implicit in the book)
- e (encryption algorithm): takes as input key $\boldsymbol{k}$ and message $\boldsymbol{m} \in \mathbb{M}$; outputs ciphertext $\boldsymbol{c} \leftarrow \mathsf{e}_{\boldsymbol{k}}(\boldsymbol{m})$

Private-key encryption

A private-key encryption scheme is defined by a message space $\mathbb{M}$, (key space $\mathbb{K}$) and algorithms $(\mathsf{KeyGen}, \mathsf{e}, \mathsf{d})$:

- KeyGen (key-generation algorithm): outputs $\boldsymbol{k} \in \mathbb{K}$. Usually: $\boldsymbol{k} \in \mathbb{K}$ uniformly random. (This algorithm is sometimes left implicit in the book)
- e (encryption algorithm): takes as input key $\boldsymbol{k}$ and message $\boldsymbol{m} \in \mathbb{M}$; outputs ciphertext $\boldsymbol{c} \leftarrow \mathsf{e}_{\boldsymbol{k}}(\boldsymbol{m})$
- d (decryption algorithm): takes as input key $\boldsymbol{k}$ and ciphertext $\boldsymbol{c}$; outputs $\boldsymbol{m}$ or "error": $\boldsymbol{m} = \mathsf{d}_{\boldsymbol{k}}(\boldsymbol{c})$

Private-key encryption



Introduction to Modern Cryptography 2nd ed. CRC Press 2015

What are the capabilities of the adversary $\boldsymbol{\mathcal{A}}$?

- Ciphertext-only attack ($\boldsymbol{\mathcal{A}}$ has access only at ciphertext, specifically:)
  - One ciphertext
  - Many ciphertexts
- Known-plaintext attack ($\boldsymbol{\mathcal{A}}$ has access to pairs of known plaintexts and their corresponding ciphertexts)
- Chosen-plaintext attack ($\boldsymbol{\mathcal{A}}$ has the ability to choose plaintexts and to view their corresponding ciphertexts)
- Chosen-ciphertext attack ($\boldsymbol{\mathcal{A}}$ has the ability to obtain the decryption of ciphertexts of its choice)

Define secure encryption

- What does it mean for encryption scheme $(\mathsf{KeyGen}, \mathsf{e}, \mathsf{d})$ to be **secure**?
- When $\mathcal{A}$ has access only to one ciphertext.

Define secure encryption. Attempt 1

$\mathcal{A}$ does not learn the key

- Consider the scheme $\mathsf{e}_{\boldsymbol{k}}(\boldsymbol{m}) = \boldsymbol{m}$

Define secure encryption. Attempt 2

$\mathcal{A}$ does not learn the plaintext from the ciphertext

- What if the adversary learns only a part of the plaintext?
- What if the adversary is able to learn some partial information about the plaintext? (e.g. is the salary $> \mathbf{30.000}$ DKK)

Define secure encryption. Attempt 3:

Perfect Secrecy

Regardless of any **prior information**, the adversary has about the plaintext, the ciphertext should leak **no additional information** about the plaintext

Perfect Secrecy

Let us start with recalling some probability elements...

Probability Review

Random variable (*RV*)

Variable that takes on (discrete) values with certain probabilities

Probability distribution (*PD*)

A *PD* for a *RV* specifies the probabilities with which the variable takes on each possible value

- Each probability must be between **0** and **1**
- The probabilities must sum to **1**

Probability Review

Event

A particular occurrence in some experiments:

- $\mathbf{Pr}[\boldsymbol{E}]$: probability of event $\boldsymbol{E}$

Conditional probability

Probability that one event occurs, given that some other event occurred:

- $\mathbf{Pr}[\boldsymbol{A}|\boldsymbol{B}] = \mathbf{Pr}[\boldsymbol{A} \textbf{ and } \boldsymbol{B}]/\mathbf{Pr}[\boldsymbol{B}] \equiv \mathbf{Pr}[\boldsymbol{AB}]/\mathbf{Pr}[\boldsymbol{B}]$

Independence

Two *RV* $\boldsymbol{X}, \boldsymbol{Y}$ are **independent** if:

- $\forall\, \boldsymbol{x}, \boldsymbol{y} :\ \mathbf{Pr}[\boldsymbol{X} = \boldsymbol{x} | \boldsymbol{Y} = \boldsymbol{y}] = \mathbf{Pr}[\boldsymbol{X} = \boldsymbol{x}]$

Probability Review

Law of total probability

Let $\boldsymbol{E_1 \dots E_n}$ are a partition of all possibilities. Then $\forall \boldsymbol{A}$:

$$\begin{aligned} \mathbf{Pr}[A] &= \sum_{i} \mathbf{Pr}[AE_i] \\ &= \sum_{i} \mathbf{Pr}[A|E_i]\ \mathbf{Pr}[E_i] \end{aligned}$$

Note

$\mathbf{Pr}[A|B] = \mathbf{Pr}[AB]/\mathbf{Pr}[B] \implies \mathbf{Pr}[AB] = \mathbf{Pr}[A|B]\mathbf{Pr}[B]$

Probability Distributions

The random variable $\boldsymbol{M}$

- $\boldsymbol{M}$ is the *RV* denoting the value of the message
- $\boldsymbol{M}$ ranges over $\mathbb{M}$; context dependent
- Reflects the likelihood of different messages being sent, given the adversary's **prior knowledge**

Example

$$\mathbf{Pr}[M = \textbf{attack today}] = \mathbf{0.7}$$
$$\mathbf{Pr}[M = \textbf{don't attack}] = \mathbf{0.3}$$

Probability Distributions

The random variable $\boldsymbol{K}$

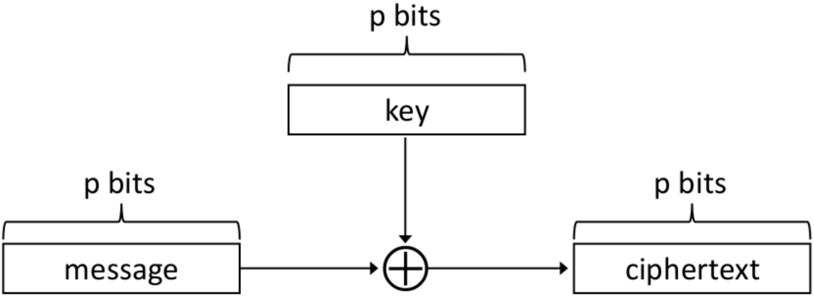
- $\boldsymbol{K}$ is the *RV* denoting the key
- $\boldsymbol{K}$ ranges over $\mathbb{K}$
- Fix some encryption scheme $(\mathsf{KeyGen}, \mathsf{e}, \mathsf{d})$
- KeyGen defines a probability distribution for $\boldsymbol{K}$:

$$\mathbf{Pr}[\boldsymbol{K} = \boldsymbol{k}] = \mathbf{Pr}[\mathsf{KeyGen}\ \textbf{outputs key}\ \boldsymbol{k}]$$

The random variable $\boldsymbol{C}$

- Fix some encryption scheme $(\mathsf{KeyGen}, \mathsf{e}, \mathsf{d})$, and some *PD* for $\boldsymbol{M}$
- Consider the following (randomized) experiment:
  - Generate a key $\boldsymbol{k}$ using KeyGen
  - Choose a message $\boldsymbol{m}$, according to the given *PD*
  - Compute $\boldsymbol{c} \leftarrow \mathsf{e}_{\boldsymbol{k}}(\boldsymbol{m})$
- **This defines a distribution on the ciphertext**
- Let $\boldsymbol{C}$ be a *RV* denoting the value of the ciphertext in this experiment

One-time Pad



Private-key encryption

A private-key encryption scheme is defined by a message space $\mathbb{M}$, (key space $\mathbb{K}$) and algorithms $(\mathsf{KeyGen}, \mathsf{e}, \mathsf{d})$:

- KeyGen (key-generation algorithm): outputs $\boldsymbol{k} \in \mathbb{K}$. Usually: $\boldsymbol{k} \in \mathbb{K}$ uniformly random. (This algorithm is sometimes left implicit in the book)
- e (encryption algorithm): takes as input key $\boldsymbol{k}$ and message $\boldsymbol{m} \in \mathbb{M}$; outputs ciphertext $\boldsymbol{c} \leftarrow \mathsf{e}_{\boldsymbol{k}}(\boldsymbol{m})$
- d (decryption algorithm): takes as input key $\boldsymbol{k}$ and ciphertext $\boldsymbol{c}$; outputs $\boldsymbol{m}$ or "error": $\boldsymbol{m} = \mathsf{d}_{\boldsymbol{k}}(\boldsymbol{c})$

One-time Pad

- Let $\mathbb{M} = \{0, 1\}^n$
- KeyGen: choose a uniform key $k \in \{0, 1\}^n$
- $\mathsf{e}_k(m) = k \oplus m$
- $\mathsf{d}_k(c) = k \oplus c$
- $\mathsf{d}_k(\mathsf{e}_k(m)) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = m$

Perfect Secrecy of One-time Pad

Theorem

The One-time Pad satisfies perfect secrecy.

Intuition

- Having observed a ciphertext, the attacker cannot conclude for certain which message was sent

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

Perfect Secrecy of One-time Pad

Proof.

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

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

Perfect Secrecy of One-time Pad

Proof.

- Recall (Bayes' theorem)

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

- We can see that $\forall c, m$

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

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.

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

$\square$

One-time Pad

- The One-time Pad achieves perfect secrecy!
- Resists even a brute-force attack
- 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?

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

Is this significant?

- $\boldsymbol{m_1 \oplus m_2}$ reveals where $\boldsymbol{m_1, m_2}$ 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`
  - $\Longrightarrow$ **letters all begin with 01...**
- ASCII code for the space character `0x20` = **00100000**
  - $\Longrightarrow$ **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!**

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

Perfect Secrecy (PS)

Is the notion too strong?

PS requires that absolutely **no information** about the plaintext is leaked, even to eavesdroppers with **unlimited computational power**

- Has some inherent drawbacks
- Seems **unnecessarily strong**

Computational Secrecy (CS)

A weaker, yet practical notion

- Still fine if a scheme **leaks information** with tiny probability to eavesdroppers with **bounded computational resources**
- i.e. we can **relax perfect secrecy** by
  1. Allowing security to "fail" with tiny probability
  2. Restricting attention to "efficient" attackers

Tiny probability of failure?

- Say security fails with probability $\mathbf{2^{-60}}$
- Should we be concerned about this?
- With probability $> \mathbf{2^{-60}}$, the sender and receiver will both be struck by lightning in the next year...
- Something that occurs with probability $\mathbf{2^{-60}}$/sec is expected to occur once every **100** billion years

Bounded attackers?

- Consider brute-force search of key space; assume one key can be tested per clock cycle
- Desktop computer $\approx \mathbf{2^{57}}$ keys/year
- Supercomputer $\approx \mathbf{2^{80}}$ keys/year
- Supercomputer since Big Bang $\approx \mathbf{2^{112}}$ keys
- Therefore restricting attention to attackers who can try $\mathbf{2^{112}}$ keys is fine!
- Modern key space: $\mathbf{2^{128}}$ keys or more...