CS306: Introduction to IT Security

Fall 2018

Lecture 4: Message authentication

Instructor: **Nikos Triandopoulos**

September 18, 2018

STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY®

Last week

- Symmetric encryption
  - introduction to modern cryptography
  - computational security
  - symmetric encryption in practice
  - modes of operation

Today

- Symmetric encryption
  - big picture, pseudorandomness, modes of operations (part II) & final remarks
- Message authentication
  - MACs
- Hash functions
  - properties & design framework
  - applications to MACs & other applications
  - generic attacks

4.0 Announcements

CS306: Announcements

- HW 1 is out
  - covers basic security concepts, symmetric encryption & (im)perfect secrecy
    - **cryptanalysis** of OTP w/ key reuse
  - due in **2 weeks** (generous timeline)
  - please
    - read instructions & syllabus, esp. w.r.t. **late-submission & collaboration policies**
    - **start early**!
- upcoming Lab 3
  - covers block ciphers & modes of operations
  - **practical assignment**

CS306: Tentative Syllabus

Week	Date	Topics	Reading	Assignment
1	Aug 28	Introduction	Ch. 1	-
2	Sep 4	Symmetric encryption	Ch. 2 & 12	Lab 1
3	Sep 11	Symmetric encryption II	Ch. 2 & 12	Lab 2, HW 1
4	**Sep 18**	**Message authentication**	**Ch. 2 & 12**	**Lab 3, HW1**
5	Sep 25	Public-key crypto II		
6	Oct 2	Access control & authentication		
–	Oct 9	**No class** (Monday schedule)		
7	Oct 16	**Midterm** (closed books)	All materials covered	

CS306: Tentative Syllabus (continued)

Week	Date	Topics	Reading	Assignment
8	Oct 23	Software & Web security		
9	Oct 30	Network security		
10	Nov 6	Database security		
11	Nov 13	Cloud security		
12	Nov 20	Privacy		
13	Nov 27	Economics		
14	Dec 4	Legal & ethical issues		
15	Dec 11 (or later)	**Final** (closed books)	All materials covered*	

* w/ focus on what covered after midterm

CS306: Course outcomes

- **Terms**
  - describe common security terms and concepts
- **Cryptography**
  - state basics/fundamentals about secret and public key cryptography concepts
- **Attack & Defense**
  - acquire basic understanding for attack techniques and defense mechanisms
- Impact
  - acquire an understanding for the broader impact of security and its integral connection to other fields in computer science (such as software engineering, databases, operating systems) as well as other disciplines including STEM, economics, and law
- Ethics
  - acquire an understanding for ethical issues in cyber-security

Discussion: Lab 2

- ASCII explanation
  - relevant to HW 1, problem 4
- Block ciphers
  - brute-force attack
  - modes of operation
- Another definition of perfect secrecy
  - 3rd equivalent view
  - more useful in certain aspects

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

Discussion: HW 1

- Problem 1
  - I really want to (try to) learn your names; ideally, your correct name… ☺
- Problem 2
  - basic CIA properties, protocol design & analysis
  - consider two different attackers: **Eve & Mallory**
- Problem 3
  - substitution codes, classical Vs OTP; consider **examples** we discussed in class
- Problem 4
  - attack against OTP
  - consider **the role of SP characters** in leaked XOR of pairs of messages

Questions?

4.1 Symmetric encryption: Big picture, pseudorandomness, modes of operations (part II) & final remarks

Recall: 3 equivalent definitions of perfect secrecy

1) a posteriori = a priori

For every $\mathcal{D}_{\mathcal{M}}$, m $\in \mathcal{M}$ and c $\in C$, for which Pr [C = c] > 0, it holds that

Pr[M = m | C = c] = **Pr[M = m]**

2) C is independent of M

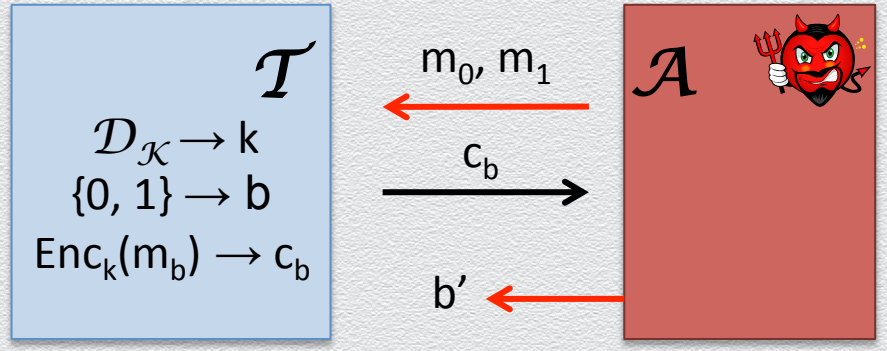
For every m, m' $\in \mathcal{M}$ and c $\in C$, it holds that

Pr[Enc_K(m) = c] = Pr[Enc_K(m') = c]

3) indistinguishability

For every $\mathcal{A}$, it holds that

Pr[b' = b] = **1/2**



ciphertext looks **completely random**

Recall: 1 security relaxation

- **perfect** security: M, $Enc_K(M)$ are independent, **unconditionally**
  - no extra information is leaked to any attacker
- **computational** security: M, $Enc_K(M)$ are independent, for all **practical** purposes
  - no extra information is leaked but a tiny amount
    - e.g., with prob. 2^{-128} or much less than the likelihood of being hit by lighting
  - to bounded attackers
    - e.g., who cannot count to 2^{128} or invest work of more than one century
- attacker's best strategy remains **ineffective**
  - random guess on secret key; or
  - exhaustive search over key space (brute-force attack)

Computational indistinguishability

Relax the definition of perfect secrecy that is based on indistinguishability

- require that target messages m_0, m_1 are chosen by a **PPT** attacker
- require that no such attacker can distinguish $Enc_k(m_0)$ from $Enc_k(m_1)$

non-negligibly better than guessing

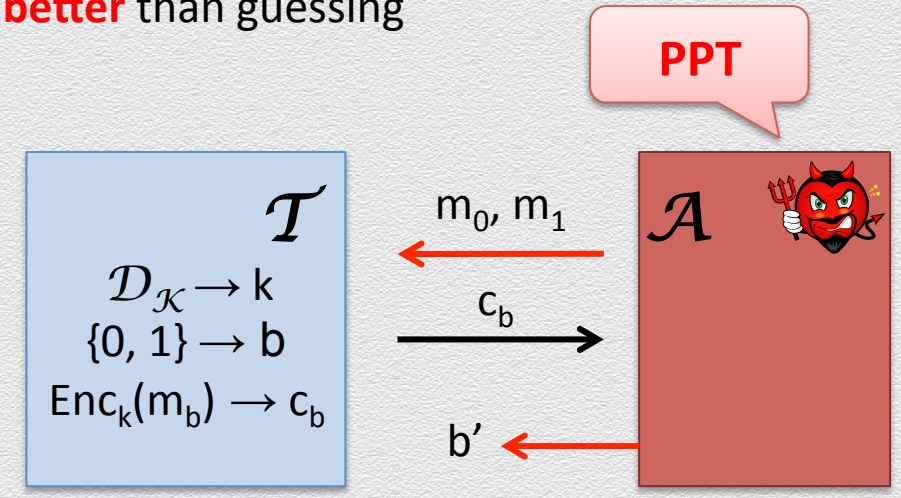
3) indistinguishability

For every $\mathcal{A}$, it holds that

Pr[b' = b] = 1/2 + negligible

PPT

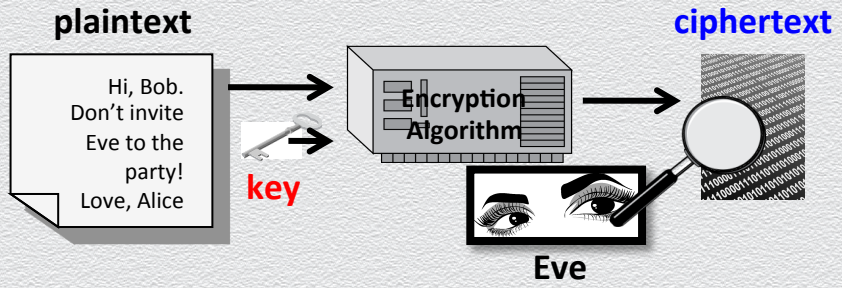
something that can be safely ignored



Recall: 2 security properties against eavesdropping

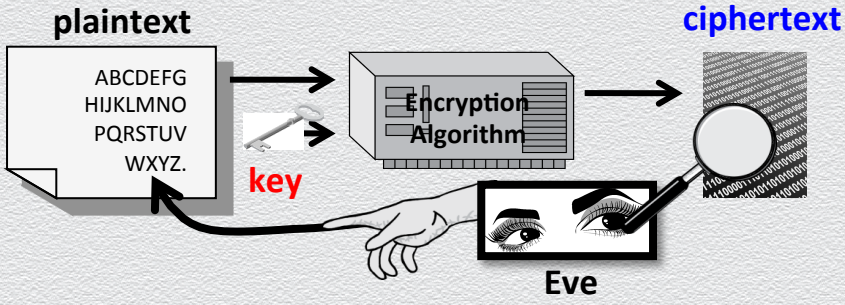
EAV security

- protects against ciphertext-only attacks



CPA security

- protects against chosen plaintext attacks



Computational CPA-security

CPA security implies probabilistic encryption – why?

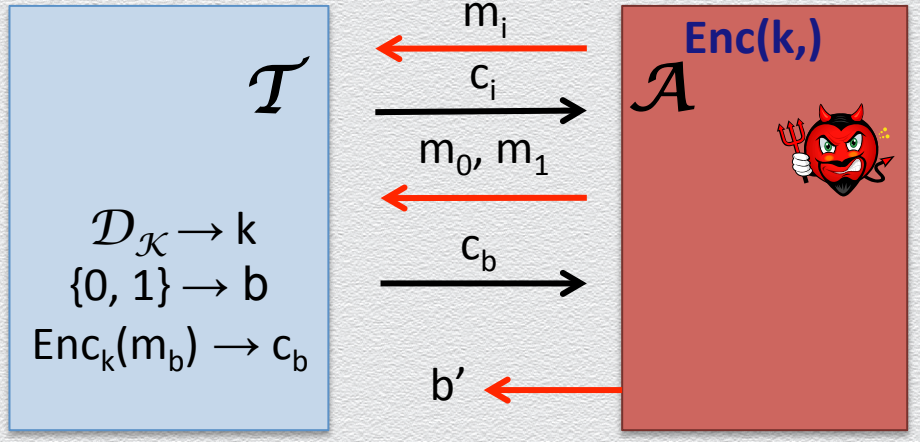
Strengthen the definition of computational EAV-security

- allow attacker to have access to an **encryption "box"**
- allow the attacker to select m_0, m_1 **after** using this "box" (as many time as desired)

3) indistinguishability

For every $\mathcal{A}$, it holds that

Pr[b' = b] = 1/2 + negligible



Recall: Perfect secrecy & randomness

In a perfectly secret cipher, the ciphertext **doesn't depend** on the message

- the ciphertext appears to be **truly random**
- the uniform key-selection distribution **is imposed also onto** produced ciphertexts
  - e.g., c = k XOR m (for uniform k and any distribution over m)
- role of randomness in encryption is **integral**

When security is computational, randomness is **relaxed** to "pseudorandomness"

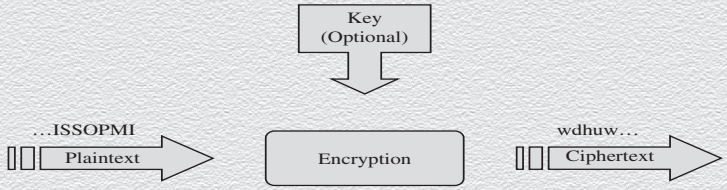
- the ciphertext appears to be "**pseudorandom**"
- it **cannot be efficiently distinguished** from truly random

Symmetric encryption & pseudorandomness

Stream cipher

Using a **short** key, it produces a **longer** pseudorandom ciphertext

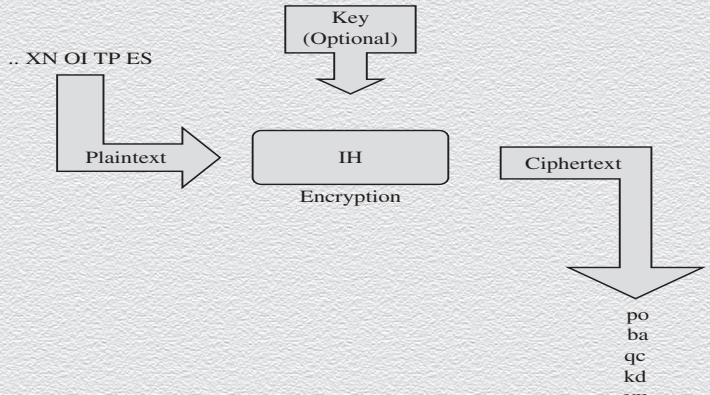
- abstract crypto primitive of **pseudorandom generator (PRG)**



Block cipher

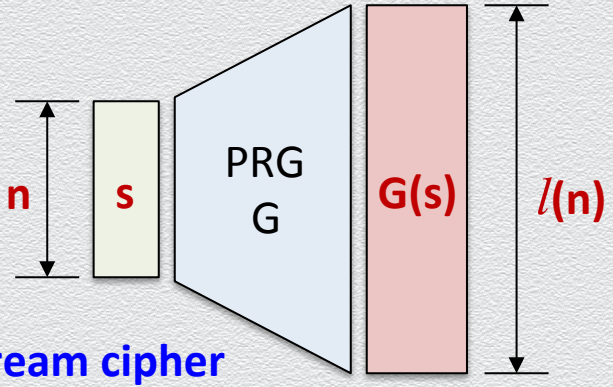
Using a (**shorter**) key, it produces a pseudorandom (**longer) input-specific** block

- abstract crypto primitive of **pseudorandom function (PRF)**



Pseudorandom generators (PRGs)

Deterministic algorithm G that
on input a **seed** $s \in \{0,1\}^n$, outputs $G(s) \in \{0,1\}^{l(n)}$



a.k.a. **stream cipher**

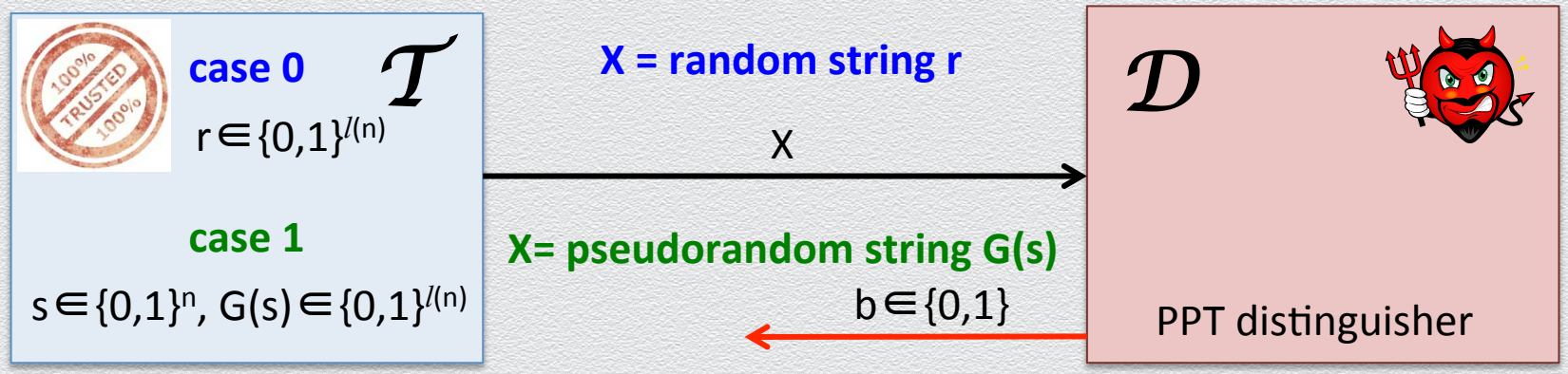
G is a PRG if:

- **expansion**
  - for polynomial *l*, it holds that for any n, *l*(n) > n
  - models the process of **extracting** randomness from a short random string
- **pseudorandomness**
  - **no efficient statistical test can tell apart** G(s) from a **truly random** string r

PRG: Security

b = 0 when $\mathcal{D}$ thinks that its input X is random

b = 1 when $\mathcal{D}$ thinks that its input X is pseudorandom



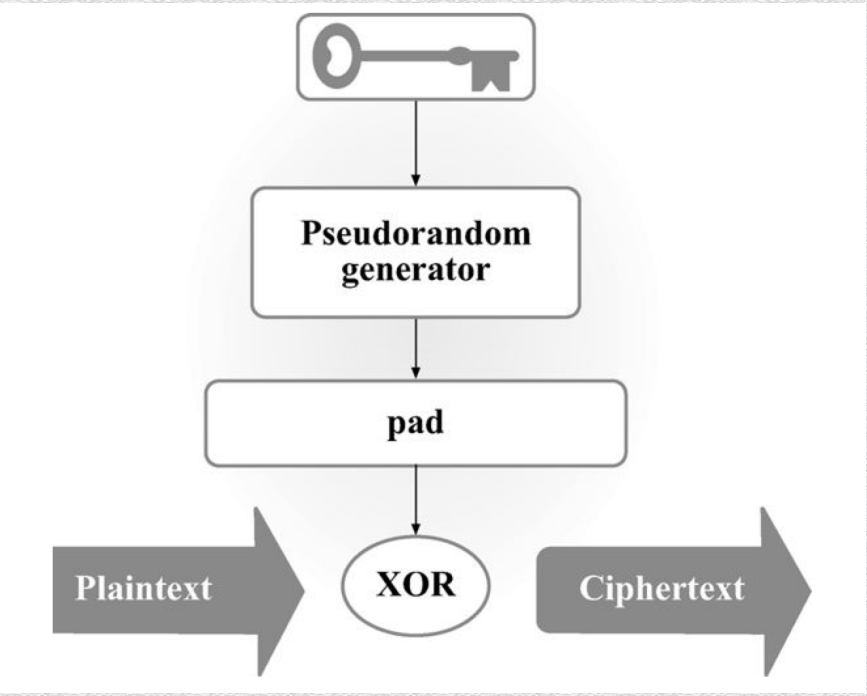
PRG G is **secure** if:

For all but negligible probability, any $\mathcal{D}$ **behaves the same** no matter what its input is!

I.e., no efficient test can meaningfully tell apart pseudorandom from truly random

Generic PRG-based symmetric encryption

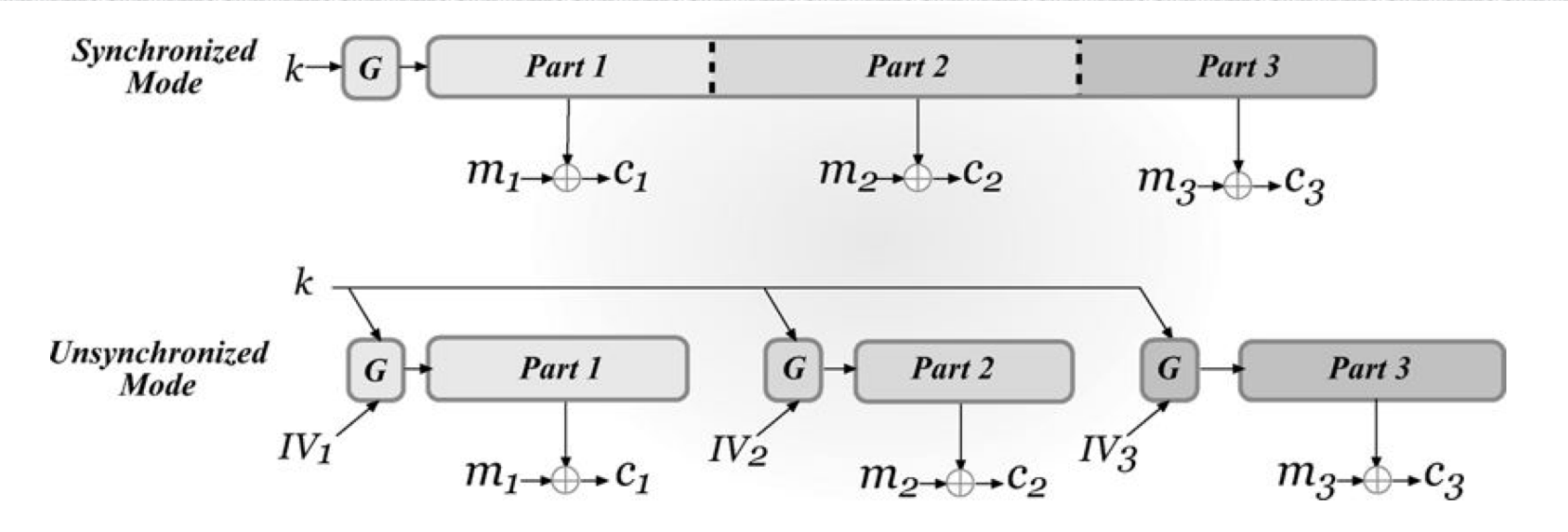
- Either fixed-length or arbitrary-length encryption



encryption scheme is EAV-secure
as long as the underlying PRG is secure

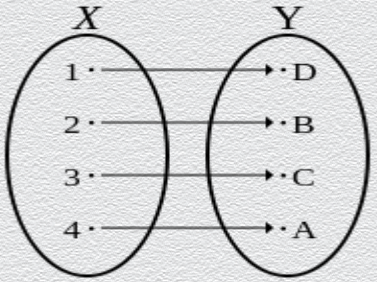
Stream ciphers: Modes of operation

on-the-fly computation of new pseudorandom bits, no IV needed, EAV-secure



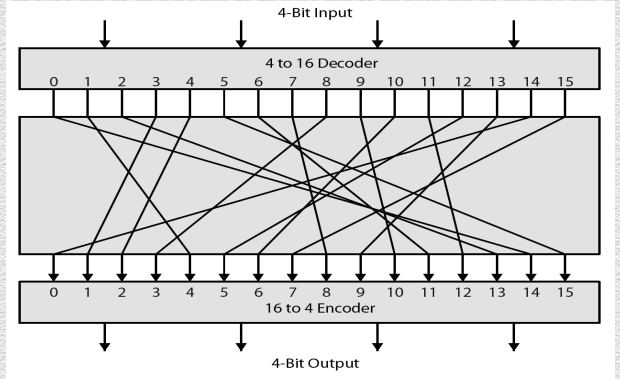
random IV used for every new message is sent along with ciphertext, CPA-secure

Recall: Ideal block ciphers



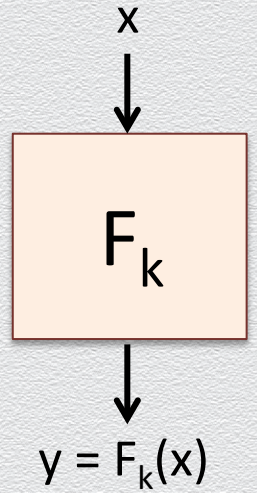
A **random** mapping of n-bit inputs to n-bit outputs

- there are ~2^(n2^n) possible such mappings
- of them, (2^n)! are permutations
- none of the above can be implemented in practice



Instead, a keyed function $F_k : \{0,1\}^n \rightarrow \{0,1\}^n$

- indexed by a k-bit key k
- there are only 2^n such keyed functions
- for this approximation to be secure, a random key should select a "random-enough" mapping, or a **pseudorandom function**



Pseudorandom functions (PRFs)

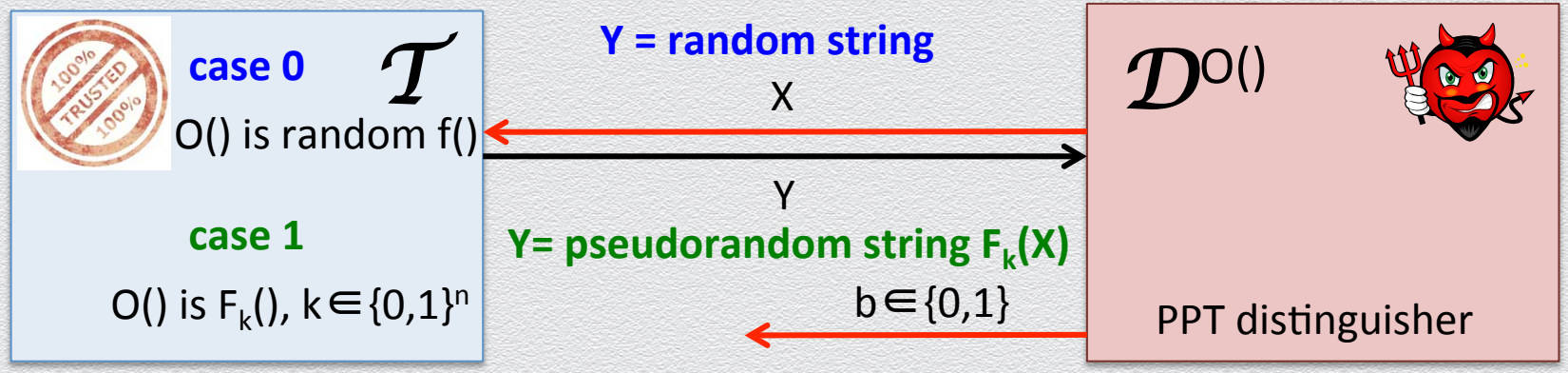
Generalize the concept of a PRG

- keyed functions producing pseudorandom bits that **depend** on a specific input
- no statistical test can tell them apart from truly **random** function

Security

b = 0 when $\mathcal{D}$ thinks that it talks to f()

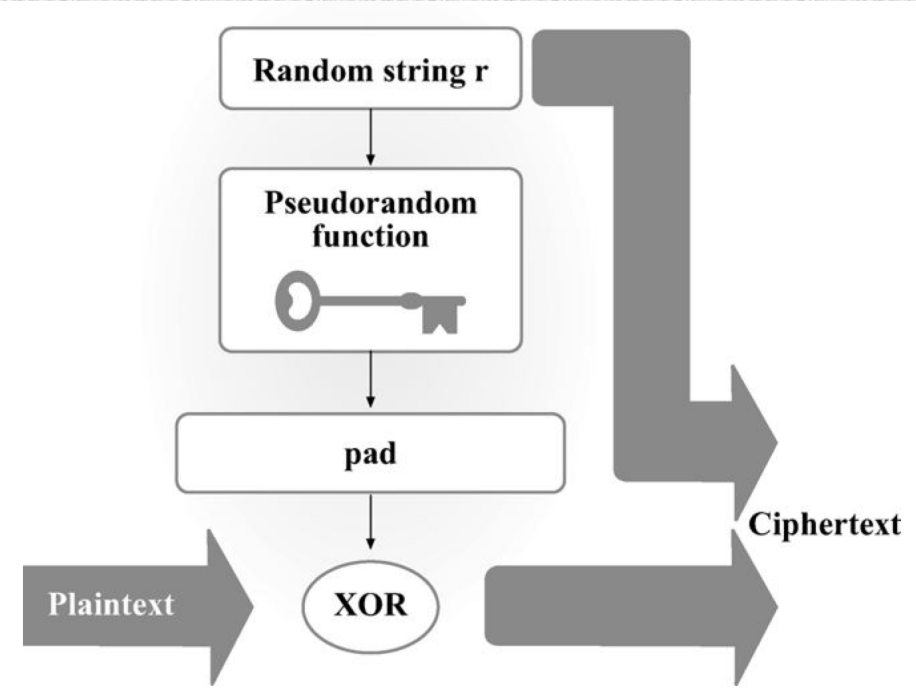
b = 1 when $\mathcal{D}$ thinks that it talks to $F_k()$



$F_k()$ is **secure** if any efficient test can only negligibly often tell apart pseudorandom from random!

Generic PRF-based symmetric encryption

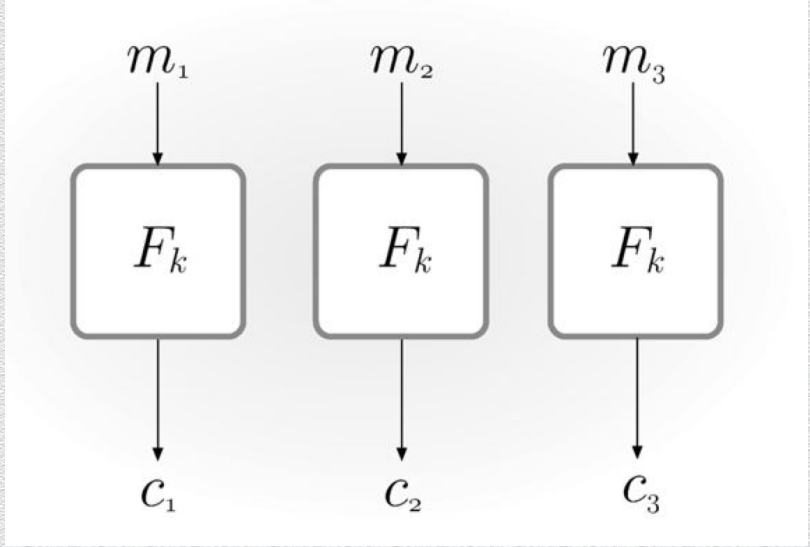
- Fixed-length encryption



encryption scheme is CPA-secure
as long as the underlying PRF is secure

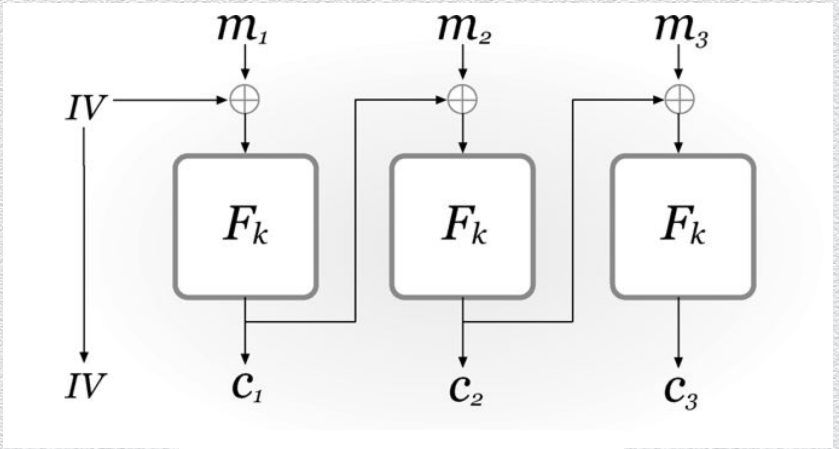
Block ciphers: Modes of operations (I)

- ECB - electronic code book
  - insecure, of only historic value
  - deterministic, thus not CPA-secure
  - actually, not even EAV-secure

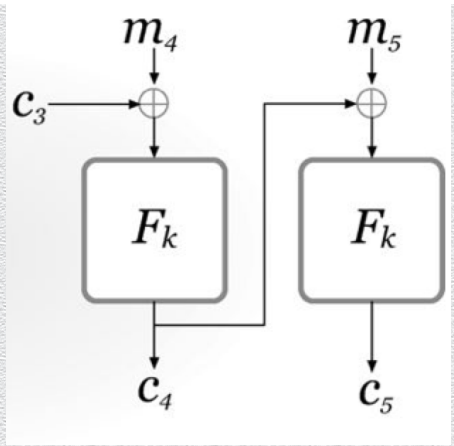


Block ciphers: Modes of operations (II)

- CBC – cipher block chaining
  - CPA-secure if F_k a permutation
  - uniform IV
    - otherwise security breaks

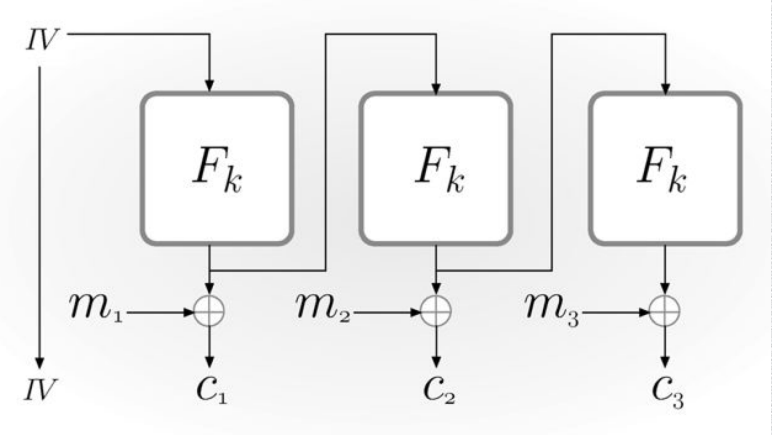


- Chained CBC
  - use last block ciphertext of current message as IV of next message
  - saves bandwidth but not CPA-secure



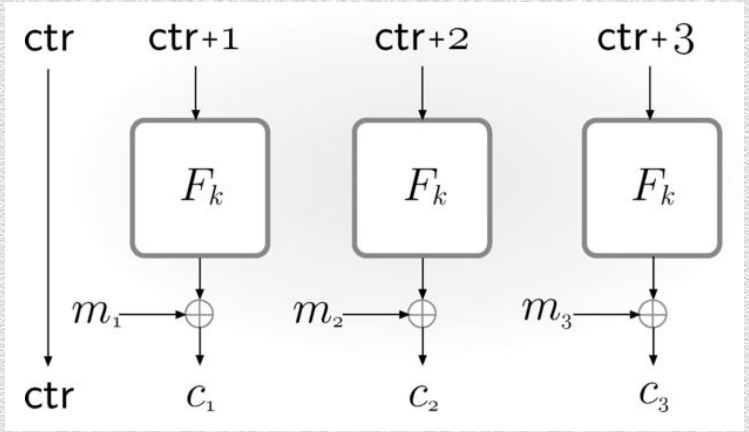
Block ciphers: Modes of operations (III)

- OFB – output feedback
  - uniform IV
  - no need message length to be multiple of n
  - resembles synchronized stream-cipher mode
  - CPA-secure if F_k is PRF



Block ciphers: Modes of operations (IV)

- CTR – counter mode
  - uniform ctr
  - no need message length to be multiple of n
  - resembles synchronized stream-cipher mode
  - CPA-secure if F_k is PRF
  - no need for F_k to be invertible
  - parallelizable



Notes on modes of operation

- block length matters
  - if small, IV or ctr can be “recycled”
- IV are often misused
  - e.g., reused or not selected uniformly at random
  - in this case, CBC is a better option than OFB/CTR

Discussion: Brute-force attacks against block ciphers

Brute-force attack amounts to checking all possible 2^k keys

- due to confusion & diffusion, the key cannot be extracted even if a valid plaintext/ciphertext pair is captured
- thus, as expected, **the longer the key size the stronger the security**

DES was designed to use the relatively short key size of 56 bits

- strengthening was attempted via **repeated** encryptions using different keys
  - thus, **seemingly** using longer keys
  - e.g., double DES encrypts **twice** as $E(k_2, E(k_1, m))$, yet it is **not effective**!
  - why?

4.2 Message authentication

Recall: Integrity

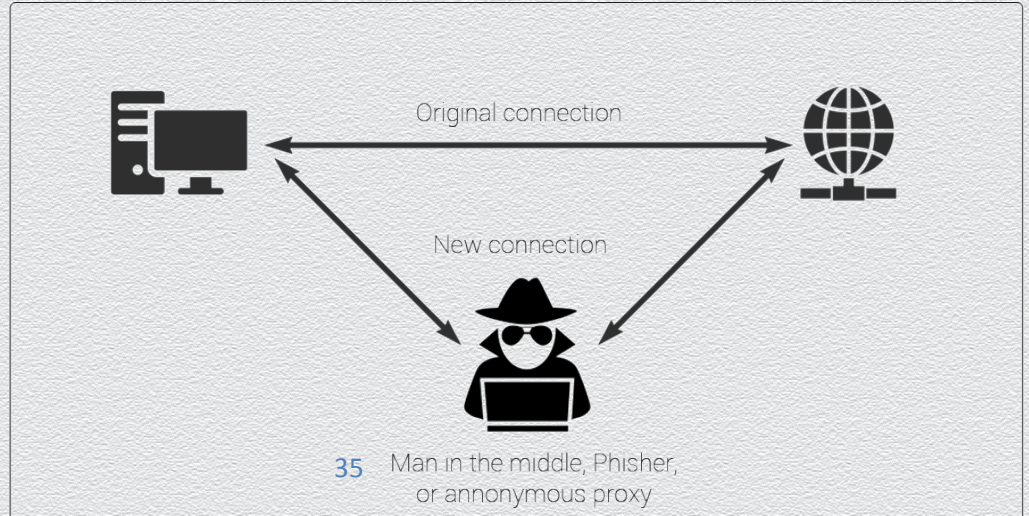
Fundamental security property

- **an asset is modified only by authorized parties**
- "I" in the CIA triad

> *"computer security seeks to prevent **unauthorized** viewing (confidentiality) or **modification (integrity)** of **data** while preserving access (availability)"*

Alteration

- main threat against integrity of **in-transit** data
- e.g., MITM attack



Security problems studied by modern cryptography

- Classical cryptography: **message encryption**
  - early crypto schemes tried to provide **secrecy / confidentiality**
- Modern cryptography: **wide variety** of security problems
  - today we need to study a large set of **security properties** beyond secrecy
- The sibling of message encryption: **message authentication**
  - **another cornerstone** of any secure system aiming to provide **authenticity & integrity**

Message authentication: Motivation

Information has **value**, but only when it is **correct**

- random, incorrect, inaccurate or maliciously altered data is **useless** or **harmful**
  - **message authentication = message integrity + authenticity**
    - while in transit (or at rest), no message should be **modified** by an outsider
    - no outsider can **impersonate** the stated message sender (or owner)

- it is often necessary / worth to protect critical / valuable data
  - **message encryption**
    - while in transit (or at rest), no message should be **leaked** to an outsider

Example 1

Secure electronic banking

- a bank receives an electronic request to transfer $1,000 from Alice to Bob

Concerns

- who ordered the transfer, Alice or an attacker (e.g., Bob)?
- is the amount the intended one or was maliciously modified while in transit?
  - adversarial Vs. random message-transmission errors
    - standard error-correction is **not sufficient** to address this concern

Example 2

Web browser cookies

- a user is performing an online purchase at Amazon
- a “cookie” contains session-related info, as client-server HTTP traffic is stateless
  - stored at the client, included in messages sent to server
  - contains client-specific info that affects the transaction
    - e.g., the user’s shopping cart along with a discount due to a coupon

Concern

- was such state maliciously altered by the client (possibly harming the server)?

Integrity of communications / computations

Highly important

- any unprotected system cannot be assumed to be trustworthy w.r.t.
  - origin/source of information (due to impersonation attacks, phishing, etc.)
  - contents of information (due to man-in-the-middle attacks, email spam, etc.)
  - overall system functionality

Prevention Vs. detection

- unless system is "closed," adversarial tampering with its integrity **cannot be avoided**!
- goal: identify system components that are not trustworthy
  - **detect tampering** or **prevent undetected tampering**
    - e.g., avoid "consuming" falsified information

Encryption does not imply authentication

A common misconception

> "since ciphertext c hides message m, Mallory cannot meaningfully modify m via c"

Why is this incorrect?

- all encryption schemes (seen so far) are based on one-time pad, i.e., masking via XOR
- consider flipping a single bit of ciphertext c; what happens to plaintext m?
  - such property of one-time pad does not contradict the secrecy definitions

Generally, secrecy and integrity are distinct properties

- encrypted traffic generally provides **no integrity** guarantees