2 /07/2025

S 6160 ryptology Lectu
Introduction & Logistic

Maria Francis

July 28 2025

2 /07/2025

Welcome to Cryptology

Introductory course on Cryptography and Cryptar

It is a theoretical course with a few practical assi

We will cover the basics of modern cryptography
functions, provable security, reductionist argumen
key cryptography and public key cryptography (P

Prerequisites

No prerequisites other than discrete mathematics
required.

But it is good to know basics of probability (e.g:
variable, discrete probability distributions) and co
complexity (e.g: NP, NP completeness, polynomi
reductions)

The number theory required will be covered in cl
a good basic resource: `http://www.hyperellip`
`tanja/teaching/crypto 13/nt.pdf`

2 /07/2025

Syllabus

Classic cryptosystems, perfect secrecy, one-way functi
random generators, private and public key cryptograph
resistant hashing, PKI, digital signatures.

Introduction to Cryptanalysis, ttacks on block cipher
search, time-space tradeoffs, differential & linear crypt
meet in the middle.

References

Introduction to Modern ryptography - J. K
Lindell. new textbook with a very modern tak

Graduate Course in pplied Cryptography - D.
Shoup. More practical engineering!

Other References

Lecture notes by

- Shafi Goldwasser and Mihir Bellare (collected in document but slightly old),
- Rafael Pass and bhi Shelat (collected into a si document), and many many more.

Other References

- Lecture notes by
  - Shafi Goldwasser and Mihir Bellare (collected in
    document but slightly old),
  - Rafael Pass and bhi Shelat (collected into a si
    document), and many many more.
- Explore and utilize all resources -
  - lecture notes from reputed universities and profe
  - video lectures (from reputed sources like univers
    Simons Institute, Institute for dvanced Study, e

2 /07/2025

Classes

We meet during B slot, Mon (10 : 00   10 : 55),
(9 : 00   9 : 55) and Thurs (11 : 00   11 : 55).

T s for this course: Reisha   li, Supreet Shukla,
and Kanchan Bisht

Evaluation!!

- Exams - 50 (Previously announced, 2 of them)
- Programming ssignments - 30
- Tutorial Exams – 20

Evaluation!!

Exams - 50 (Previously announced, 2 of them)

Programming ssignments - 30

Tutorial Exams – 20

Plagiarism or any form of cheating will be an aut
will be reported to the dept.

ssignments

- bout 3 4 programming assignments
- With all written exams, assignments have becom
important - the only way for you to see crypto co
practice!

ssignments

bout 3 4 programming assignments

With all written exams, assignments have becom
important - the only way for you to see crypto c
practice!

Do not copy or cheat! Plagiarism or any form of
be an automatic F and will be reported to the de

Tutorials

We will have tutorials every week on Wednesday.

2 /07/2025

Tutorials

We will have tutorials every week on Wednesday.
will be posted on Fridays, ≈ 2 3 questions.

Tutorials

We will have tutorials every week on Wednesday. will be posted on Fridays, $\approx 2 \quad 3$ questions.

During the tutorial sessions, the T s will work ou questions for you.

2 /07/2025

Tutorials

- We will have tutorials every week on Wednesday.
will be posted on Fridays, $\approx 2 \quad 3$ questions.
- During the tutorial sessions, the T s will work ou
questions for you.
- But they will not spend a lot of time so it is criti
think about these problems on your own before th
only use these sessions to make sure your thinkin
and clear specific doubts!

Tutorials

- We will have tutorials every week on Wednesday. will be posted on Fridays, $\approx 2 \quad 3$ questions.
- During the tutorial sessions, the T s will work ou questions for you.
- But they will not spend a lot of time so it is criti think about these problems on your own before th only use these sessions to make sure your thinkin and clear specific doubts!
- The last 20 min will be a small quiz with one que on the questions you have done so far.

2 /07/2025

Tutorials

- We will have tutorials every week on Wednesday.
will be posted on Fridays, $\approx 2 \quad 3$ questions.
- During the tutorial sessions, the T s will work ou
questions for you.
- But they will not spend a lot of time so it is criti
think about these problems on your own before th
only use these sessions to make sure your thinkin
and clear specific doubts!
- The last 20 min will be a small quiz with one que
on the questions you have done so far.
- If you have applied the technique but have not fu
answer then you get 8 marks. If you have done it
then you get 10 marks. Else 0.

2 /07/2025

Tutorials

We will have tutorials every week on Wednesday. will be posted on Fridays, $\approx 2 \quad 3$ questions.

During the tutorial sessions, the T s will work ou questions for you.

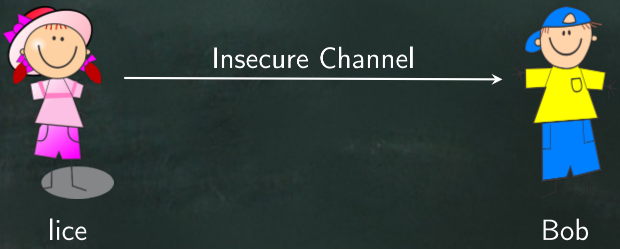
But they will not spend a lot of time so it is criti think about these problems on your own before th only use these sessions to make sure your thinkin and clear specific doubts!

The last 20 min will be a small quiz with one que on the questions you have done so far.

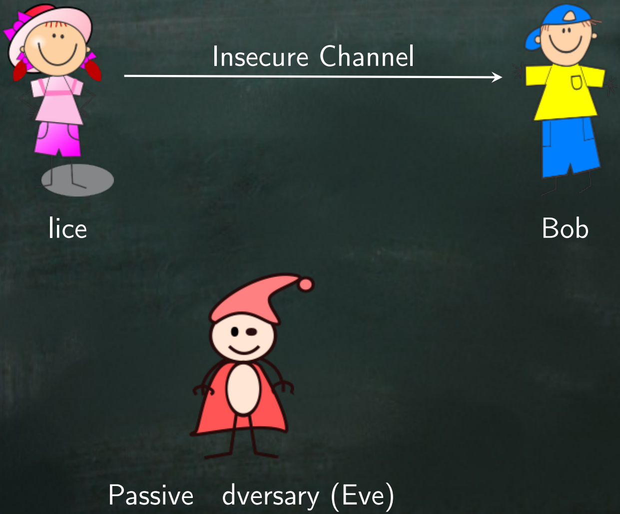
If you have applied the technique but have not fu answer then you get 8 marks. If you have done it then you get 10 marks. Else 0.

We will eliminate the lowest 3 tutorial marks in t

The need for cryptography

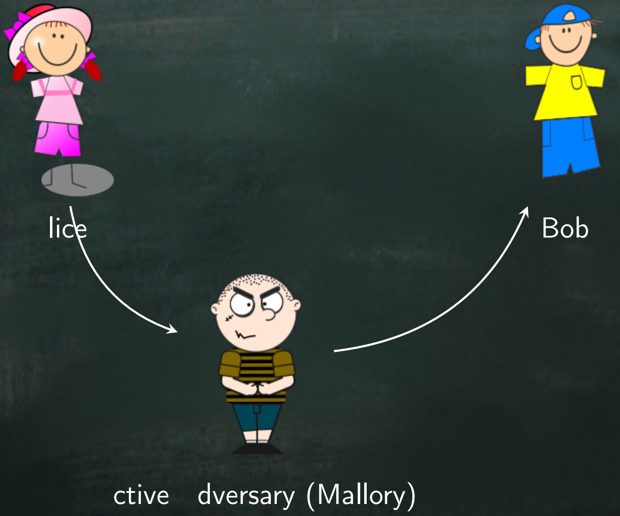


The need for cryptography



2 /07/2025

The need for cryptography



2 /07/2025

The need for cryptography

- Communicating or computing over a channel whe adversaries.
- There are information systems(PCs, cellphones, computers, TMs, cars, smart grids, etc.)
- Our aim is to control access to the information – looking to see if we can control who sees and mo information.
- Some examples of such rules :
  - Only XX can read the contents of the file
  - The contents of this file has not been changed a
  - The recipient of this email can authenticate the

Our ims and the Tools at Hand

What we aim to achieve:

- Data Confidentiality
- Data Integrity
- uthentication –
- Non-repudiation – the sender cannot claim that not send it

How do we achieve them? Tools such as:

- Encryption
- Hash Functions
- Digital Signatures
- Zero knowledge Proofs

What about the dversary?

- Could be anybody – insider/outsider
- ssume he knows system design including implen details
- Resources – powerful computers, ability to interc ability to collude with some participants
- Generally generous assumptions about adversary' we assume a computationally bounded adversary.

2 /07/2025

Simple Solution

lice

Key k
Encrytion lgorithm *Enc*
Decryption lgorithm *Dec*

ll previously agreed upon

Simple Solution

$$c = Enc_k(m)$$

lice

Bob

Retrieving the mes
$m = Dec_k(c$

2 /07/2025

Simple Solution

$c = Enc_k(m)$

lice

Bob

Retrieving the mes

$m = Dec_k(c)$

Questions: Which of these are secret here?
How are the secrets agreed upon? Parties may not ev
before?

Formalizing the Solution

Symmetric key encryption (SKE) scheme consists of:

- $\mathcal{M}$: a set of possible plaintexts
- $\mathcal{C}$: a set of possible ciphertexts
- $\mathcal{K}$: a set of possible keys.
- Gen (called the key generation algorithm) is a ra
  algorithm that returns a key k such that $k \in \mathcal{K}$.
  - family of encryption functions, $Enc_k : \mathcal{M} \to \mathcal{C}$
  - family of decryption functions, $Dec_k : \mathcal{C} \to \mathcal{M}$

such that $Dec_k(Enc_k(m)) = m$ for all $m \in \mathcal{M}$ an

2 /07/2025

Timeline of Cryptography as a fiel

0

ncient times to 1900s:
Classical
Ciphers

1900s :
Mechanical
Ciphers

M
-
Pu

Classical Ciphers

The first idea that comes up when you need secret co

Popular upto 1900s :

- Shift/Ceaser cipher
- Substitution cipher
- Vigenère cipher, etc

S	E	N	D	R	E	I	N	F	O	R	C	E	M	E	N
V	I	G	E	N	E	R	E	V	I	G	E	N	E	R	E
N	M	T	H	E	I	Z	R	A	W	X	G	R	Q	V	R

2 /07/2025

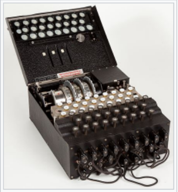
Mechanical Ciphers - Enigma, Pur

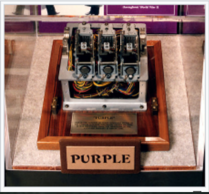
Motor devices

Electromechanical devices

Cryptography performed by (typically, rotor) mac

lan Turing and others at Bletchley Park, Willian
and others in the US – all helped break these ci





2 /07/2025

Classical Cryptosystems

s time proceeds the cryptosystems get more sop
but they are all broken!

2 /07/2025

Classical Cryptosystems

s time proceeds the cryptosystems get more sop
but they are all broken!

The main idea was security by obscurity.

Gen, *Enc*, *Dec* and the generated key k were sec

Less information we give to the adversary, the ha
break the scheme, right??

Classical Cryptosystems

s time proceeds the cryptosystems get more sop
but they are all broken!

The main idea was security by obscurity.

Gen, *Enc*, *Dec* and the generated key k were sec

Less information we give to the adversary, the ha
break the scheme, right??

Not really! – Kerchoff's principle (1884)

2 /07/2025

Kerchoff's principle

The only thing that should be private is the key
Dec should be assumed to be public.

Kerchoff's principle

The only thing that should be private is the key
Dec should be assumed to be public.

Conservative approach – guarantees that security
even if everything but the key is known to the ad

First step to formally defining the security of enc
schemes.

Kerchoff's principle

The only thing that should be private is the key
Dec should be assumed to be public.

Conservative approach – guarantees that security
even if everything but the key is known to the ad

First step to formally defining the security of enc
schemes.

Immediate consequence – all of the algorithms (
Dec) cannot be deterministic.

If so, then Eve would be able to compute everyth
and Bob could compute and would thus be able t
anything that Bob can decrypt.

Kerchoff's principle

The only thing that should be private is the key *
Dec should be assumed to be public.

Conservative approach – guarantees that security
even if everything but the key is known to the ad

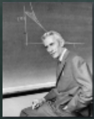
First step to formally defining the security of enc
schemes.

Immediate consequence – all of the algorithms (
Dec) cannot be deterministic.

If so, then Eve would be able to compute everyth
and Bob could compute and would thus be able t
anything that Bob can decrypt.

To prevent this we require Gen to be randomized

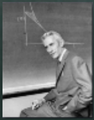
Perfect Secrecy (C. Shannon (1916



Can we perfect secrecy (information theoretic sec

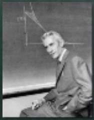
2 /07/2025

Perfect Secrecy (C. Shannon (1916



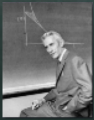
Can we perfect secrecy (information theoretic sec
adversary has infinite computational resources an
able to break it?

Perfect Secrecy (C. Shannon (1916



Can we perfect secrecy (information theoretic sec
adversary has infinite computational resources an
able to break it?
Yes!

Perfect Secrecy (C. Shannon (1916

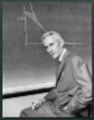


Can we perfect secrecy (information theoretic sec
adversary has infinite computational resources an
able to break it?

Yes!

- Key is perfectly random
- Key is at least as long as the message
- Key is never re-used!

Perfect Secrecy (C. Shannon (1916



Can we perfect secrecy (information theoretic sec
adversary has infinite computational resources an
able to break it?

Yes!

- Key is perfectly random
- Key is at least as long as the message
- Key is never re-used!

One Time Pad/ Vernam Cipher - unbreakable!

2 /07/2025

One Time Pad - n Overview

$Enc_{k=k_1 k_2 \quad k_n}(m \;\; m_2 \quad m_n) = c \;\; c_2 \quad c_n,$ where $c_i =$

One Time Pad - n Overview

$Enc_{k=k_1 k_2 \quad k_n}(m \; m_2 \quad m_n) = c \; c_2 \quad c_n,$ where $c_i =$

Perfectly secret if and only if key is random, is as

message and is never re-used! - Shannon's theore

One Time Pad - n Overview

$Enc_{k=k_1k_2\ \ k_n}(m\ \ m_2\quad m_n) = c\ \ c_2\quad c_n,$ where $c_i =$

Perfectly secret if and only if key is random, is as
message and is never re-used! - Shannon's theore

This is not practical!

2 /07/2025

One Time Pad - n Overview

$Enc_{k=k_1 k_2 \;\; k_n}(m \; m_2 \quad m_n) = c \; c_2 \quad c_n,$ where $c_i =$

Perfectly secret if and only if key is random, is as

message and is never re-used! - Shannon's theore

This is not practical! Why?

Computationally Bounded dversa

re we done then? We cannot achieve a good ci

Computationally Bounded dversa

re we done then? We cannot achieve a good ci

Key thing is assumption!

Computationally Bounded dversa

re we done then? We cannot achieve a good ci

Key thing is assumption! Eve has unbounded cor
power - too strong!

Computationally Bounded dversa

re we done then? We cannot achieve a good ci

Key thing is assumption! Eve has unbounded co
power - too strong!

So we make a reasonable assumption – Eve has
probabilistic polynomial time (PPT) computing

Computationally Bounded dversa

re we done then? We cannot achieve a good ci

Key thing is assumption! Eve has unbounded co power - too strong!

So we make a reasonable assumption – Eve has probabilistic polynomial time (PPT) computing p fair, we restrict lice and Bob to PPT as well.

Computational Complexity Theor

systematic study of what computationally boun
can and cannot do

Started with lan Turing but formal study with C
ward '82), Karp (Turing ward '85) and Blum (
'95)

Notions that are critical in this area – polynomia
reductions, NP-completeness

Computational Complexity Theory

systematic study of what computationally bour
can and cannot do

Started with lan Turing but formal study with C
ward '82), Karp (Turing ward '85) and Blum (
'95)

Notions that are critical in this area – polynomial
reductions, NP-completeness

Many ideas are still "conjectured" or believed!

Computational Complexity Theory

systematic study of what computationally bour
can and cannot do

Started with lan Turing but formal study with C
ward '82), Karp (Turing ward '85) and Blum (
'95)

Notions that are critical in this area – polynomial
reductions, NP-completeness

Many ideas are still "conjectured" or believed!

The foundation of modern cryptography.

Big-O Notation

efinition

Let $f, g : \quad \to \mathbb{R}$ be two functions on .

Big-O Notation

efinition

Let $f, g : \ \to \mathbb{R}$ be two functions on . Then we say
asymptotic upper bound for f if there exists a real nu
and an integer $n_0 \in$ such that

Big-O Notation

efinition

Let $f, g : \quad \to \mathbb{R}$ be two functions on . Then we say
asymptotic upper bound for f if there exists a real nu
and an integer $n_0 \in$ such that

$$|f(n)| \leq \quad |g(n)| \text{ for all } n \geq n_0,$$

and one writes $f = \mathcal{O}(g)$ or $f \in \mathcal{O}(g)$.

Big-O Notation

We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Big-O Notation

We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

2 /07/2025

Big-O Notation

We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

$$f(n) = 2n^3 + n^2 + 7n + 2.$$

2 /07/2025

Big-O Notation

We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

- $f(n) = 2n^3 + n^2 + 7n + 2$. Since $n^2 \leq n^3$, $n \leq$
  for $n \geq 1$ one has $f(n) \leq (2 + 1 + 7 + 2)n^3$.

Big-O Notation

We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

$f(n) = 2n^3 + n^2 + 7n + 2$. Since $n^2 \leq n^3$, $n \leq$
for $n \geq 1$ one has $f(n) \leq (2 + 1 + 7 + 2)n^3$. Se
$n_0 = 1$. Thus $f = \mathcal{O}(n^3)$.

Big-O Notation

We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

$f(n) = 2n^3 + n^2 + 7n + 2$. Since $n^2 \leq n^3$, $n \leq$
for $n \geq 1$ one has $f(n) \leq (2 + 1 + 7 + 2)n^3$. Se
$n_0 = 1$. Thus $f = \mathcal{O}(n^3)$.

Let $f(n) = 100 + \frac{20}{n}$. Set $= 101$ and $n_0 = 1$
$\frac{20}{n} \leq 1$ for $n \geq 19$ we have $f = \mathcal{O}(1)$.

Big-O Notation

We say that f has linear, quadratic, cubic or poly growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$ $f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

$f(n) = 2n^3 + n^2 + 7n + 2$. Since $n^2 \leq n^3$, $n \leq$ for $n \geq 1$ one has $f(n) \leq (2 + 1 + 7 + 2)n^3$. Se $n_0 = 1$. Thus $f = \mathcal{O}(n^3)$.

Let $f(n) = 100 + \frac{20}{n}$. Set $= 101$ and $n_0 = 1$ $\frac{20}{n} \leq 1$ for $n \geq 19$ we have $f = \mathcal{O}(1)$. That is asymptotically bounded by a constant.

Big-O Notation

We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

- $f(n) = 2n^3 + n^2 + 7n + 2$. Since $n^2 \leq n^3$, $n \leq$
  for $n \geq 1$ one has $f(n) \leq (2 + 1 + 7 + 2)n^3$. Se
  $n_0 = 1$. Thus $f = \mathcal{O}(n^3)$.
- Let $f(n) = 100 + \frac{20}{n}$. Set $= 101$ and $n_0 = 1$
  $\frac{20}{n} \leq 1$ for $n \geq 19$ we have $f = \mathcal{O}(1)$. That is
  asymptotically bounded by a constant.
- Let $f(n) = 5\sqrt{2^{n \quad 3} + n^2 \quad 2n}$ then $f = \mathcal{O}(2^{n \quad 2}$

2 /07/2025

Big-O Notation

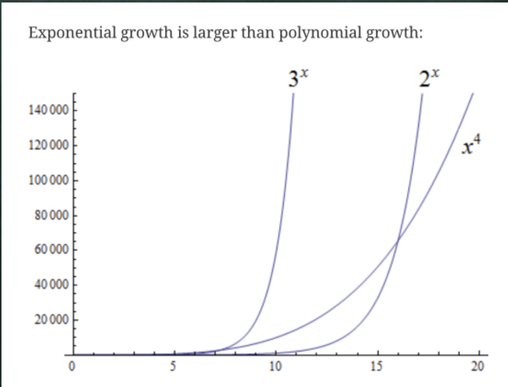
We say that f has linear, quadratic, cubic or poly
growth in n if $f = \mathcal{O}(n)$, $f = \mathcal{O}(n^2)$, $f = \mathcal{O}(n^3)$
$f = \mathcal{O}(n^k)$, for some $k \in$, respectively.

Examples:

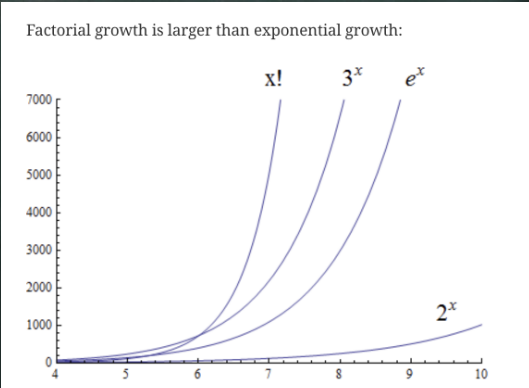
- $f(n) = 2n^3 + n^2 + 7n + 2$. Since $n^2 \leq n^3$, $n \leq$
  for $n \geq 1$ one has $f(n) \leq (2+1+7+2)n^3$. Se
  $n_0 = 1$. Thus $f = \mathcal{O}(n^3)$.
- Let $f(n) = 100 + \frac{20}{n}$. Set $\quad = 101$ and $n_0 = 1$
  $\frac{20}{n} \leq 1$ for $n \geq 19$ we have $f = \mathcal{O}(1)$. That is
  asymptotically bounded by a constant.
- Let $f(n) = 5\sqrt{2^{n \quad 3} + n^2 \quad 2n}$ then $f = \mathcal{O}(2^{n \quad 2}$
  exponentially in n.

2 /07/2025

Growth of Functions



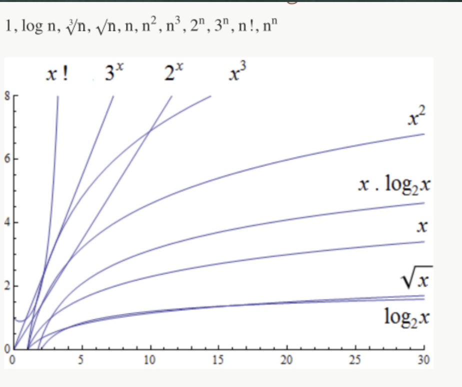
Growth of Functions



Pic courtesy – `https://ggc-discrete-math.github.io/growth_functi`

2 /07/2025

Growth of Functions



Efficient Vs Not so efficient algorit

We have a notation now to asses the running tim
algorithms in the worst case.

Efficient Vs Not so efficient algorit

We have a notation now to asses the running tim
algorithms in the worst case.

lgorithms with polynomial running time in term
parameter are considered efficient or fast.

2 /07/2025

Efficient Vs Not so efficient algorit

We have a notation now to asses the running tim
algorithms in the worst case.

lgorithms with polynomial running time in term
parameter are considered efficient or fast.

Many standard algorithms including adding or mu
numbers are polynomial.

Efficient Vs Not so efficient algorit

We have a notation now to asses the running tim
algorithms in the worst case.

lgorithms with polynomial running time in term
parameter are considered efficient or fast.

Many standard algorithms including adding or mu
numbers are polynomial.

n algorithm that loops over every instance of a
elements is exponential in n.

2 /07/2025

Efficient Vs Not so efficient algorit

We have a notation now to asses the running tim
algorithms in the worst case.

lgorithms with polynomial running time in term
parameter are considered efficient or fast.

Many standard algorithms including adding or mu
numbers are polynomial.

n algorithm that loops over every instance of a
elements is exponential in n.

problem is said to be hard if no efficient, i.e. n
polynomial-time algorithm exists to solve such a

Efficient Vs Not so efficient algorit

We have a notation now to asses the running tim
algorithms in the worst case.

lgorithms with polynomial running time in term
parameter are considered efficient or fast.

Many standard algorithms including adding or mu
numbers are polynomial.

n algorithm that loops over every instance of a
elements is exponential in n.

problem is said to be hard if no efficient, i.e. n
polynomial-time algorithm exists to solve such a

That is every algorithm solving the problem will h
super-polynomial running time *in the worst case*

P vs NP

Problems for which there is a polynomial time alg
are said to belong to the complexity class P.

P vs NP

Problems for which there is a polynomial time alg
are said to belong to the complexity class P.

decision problem has only two possible answers

P vs NP

Problems for which there is a polynomial time alg
are said to belong to the complexity class P.

decision problem has only two possible answers
The class NP (nondeterministic polynomial) is th
decision problems which can be *verified* in polyno

P vs NP

Problems for which there is a polynomial time alg
are said to belong to the complexity class P.

decision problem has only two possible answers
The class NP (nondeterministic polynomial) is th
decision problems which can be *verified* in polyno

Checking the correctness of the proof looks like i
solving a problem.

P vs NP

Problems for which there is a polynomial time alg
are said to belong to the complexity class P.

decision problem has only two possible answers
The class NP (nondeterministic polynomial) is th
decision problems which can be *verified* in polyno

Checking the correctness of the proof looks like is
solving a problem. It is conjectured that NP is st
than P, though it remains unresolved.

2 /07/2025

P vs NP

Problems for which there is a polynomial time alg
are said to belong to the complexity class P.

decision problem has only two possible answers
The class NP (nondeterministic polynomial) is th
decision problems which can be *verified* in polyno

Checking the correctness of the proof looks like is
solving a problem. It is conjectured that NP is st
than P, though it remains unresolved.

We assume that our adversaries are efficient adve

P vs NP

Problems for which there is a polynomial time alg
are said to belong to the complexity class P.

decision problem has only two possible answers
The class NP (nondeterministic polynomial) is th
decision problems which can be *verified* in polyno

Checking the correctness of the proof looks like is
solving a problem. It is conjectured that NP is st
than P, though it remains unresolved.

We assume that our adversaries are efficient adve

ctually we assume they are probabilistic polynon
adversaries. More on that later!

Crypto Vs Computational Comple
World

In computer science, we are interested in worst ca
of an algorithm that solves a certain problem.

Crypto Vs Computational Comple
World

In computer science, we are interested in worst ca
of an algorithm that solves a certain problem.

For a cryptosystem is insecure if attacks are ineff
certain bad cases but efficient in most other case

2 /07/2025

Crypto Vs Computational Comple World

In computer science, we are interested in worst ca of an algorithm that solves a certain problem.

For a cryptosystem is insecure if attacks are ineff certain bad cases but efficient in most other case

So it requires average complexity of algorithms to that breaking the scheme will be hard for most o i.e. for randomly picked instances.

2 /07/2025

nother ssumption

Eve should only have negligible chance to guess t
Eve cannot break the system with significant prol
after poly no of chances.

nother ssumption

Eve should only have negligible chance to guess t
Eve cannot break the system with significant prol
after poly no of chances.

efinition (negl( ))

function $v(k)$ is called negligible, negl(k), if:

$$(\forall c > 0)(\exists k\ )(\forall k \geq k\ )[v(k) < \frac{1}{k^c}]$$

2 /07/2025

nother ssumption

Eve should only have negligible chance to guess t
Eve cannot break the system with significant prol
after poly no of chances.

efinition (negl( ))

function $v(k)$ is called negligible, $\text{negl}(k)$, if:

$$(\forall c > 0)(\exists k\ )(\forall k \geq k\ )[v(k) < \frac{1}{k^c}]$$

n gl function is one that is asymptotically smaller
inverse poly. function.

nother ssumption

Eve should only have negligible chance to guess t
Eve cannot break the system with significant prol
after poly no of chances.

efinition (negl( ))

function $v(k)$ is called negligible, negl(k), if:

$$(\forall c > 0)(\exists k\ )(\forall k \geq k\ )[v(k) < \frac{1}{k^c}]$$

n gl function is one that is asymptotically smaller
inverse poly. function.

$$\forall c > 0,\ lim_{k \to \infty} f(k)k^c = 0$$

Exercises/Tutorials

- Is $2^{n} = \mathcal{O}(2^n)$? Is $2^{2n} = \mathcal{O}(2^n)$?
- Show that $2^{\;k}$, $k^{\;\mathit{logk}}$ are $\mathrm{n\;gl}$ but $1/k^{\;000}$ is
- Let $\mathrm{n\;gl}\;\;\mathrm{n\;gl}_2$ be two negligible functions. Prov
  sum $\mathrm{n\;gl} + \mathrm{n\;gl}_2$ is negligible.
- Negligible function stays negligible even after pol
  $\mathrm{poly}(k) \cdot \mathrm{n\;gl}(k) = \mathrm{n\;gl}(k)$.

Security Parameter

It is a measure of both the resource requirements
cryptographic protocol/algorithm as well as the p
the adversary breaking the system.

Security Parameter

It is a measure of both the resource requirements cryptographic protocol/algorithm as well as the p the adversary breaking the system.

Represented as k or n and security guarantees ar asymptotically as a function of k or n. It is also unary notation as 1^k or 1^n.

Security Parameter

- It is a measure of both the resource requirements cryptographic protocol/algorithm as well as the p the adversary breaking the system.
- Represented as k or n and security guarantees ar asymptotically as a function of k or n. It is also unary notation as 1^k or 1^n.
- One example is key size given in bits. For RS it of the modulus n in bits.

Security Parameter

It is a measure of both the resource requirements
cryptographic protocol/algorithm as well as the p
the adversary breaking the system.

Represented as k or n and security guarantees ar
asymptotically as a function of k or n. It is also
unary notation as 1^k or 1^n.

One example is key size given in bits. For RS it
of the modulus n in bits.

lice and Bob and the adversary are assumed to
polynomial in k and the adversary can break the
probability negligible in k.

2 /07/2025

Security Parameter

- It is a measure of both the resource requirements
cryptographic protocol/algorithm as well as the p
the adversary breaking the system.
- Represented as k or n and security guarantees ar
asymptotically as a function of k or n. It is also
unary notation as 1^k or 1^n.
- One example is key size given in bits. For RS it
of the modulus n in bits.
- lice and Bob and the adversary are assumed to
polynomial in k and the adversary can break the
probability negligible in k.
- Increasing k we get higher security but a degrade
What is the correct value of k?

2 /07/2025

What changes when we weaken assumptions?

We can have a cipher where the key is a pseudor
generated from a small random seed.

What changes when we weaken assumptions?

We can have a cipher where the key is a pseudor
generated from a small random seed.

$Pseudorandom(k) \oplus m = c$, where k is a fixed le
or 256 bits) uniformly random small seed.

What changes when we weaken assumptions?

We can have a cipher where the key is a pseudor
generated from a small random seed.

$Pseudorandom(k) \oplus m = c$, where k is a fixed le
or 256 bits) uniformly random small seed.

The small key is reused for all messages and encr
scheme is not perfectly secret.

What changes when we weaken assumptions?

We can have a cipher where the key is a pseudor
generated from a small random seed.

Pseudorandom$(k) \oplus m = c$, where k is a fixed le
or 256 bits) uniformly random small seed.

The small key is reused for all messages and encr
scheme is not perfectly secret. Why is this a viab

2 /07/2025

1970s - Public Key Revolution

Merkle, and independently Hellman and Diffie, in
notion of public-key cryptography.

In November 1976, Diffie and Hellman published
Directions in Cryptography, proclaiming *We are a*
a revolution in cryptography.

PKE - Diffie/Hellman Ideas

Every party has a public key PK and a privat

2 /07/2025

PKE - Diffie/Hellman Ideas

Every party has a public key PK and a privat

nybody who wants to send a message m to u
encrypt:

$$c = Enc_{PK}\ (m)$$

will use SK to decrypt:

$$m = Dec_{SK}\ (c)$$

PKE - Diffie/Hellman Ideas

Every party has a public key PK and a privat

nybody who wants to send a message m to u

encrypt:

$$c = Enc_{PK}(m)$$

will use SK to decrypt:

$$m = Dec_{SK}(c)$$

Should be efficient to compute PK , SK pairs.

PKE - Diffie/Hellman Ideas

Every party has a public key PK and a privat

nybody who wants to send a message m to u
encrypt:

$$c = Enc_{PK}(m)$$

will use SK to decrypt:

$$m = Dec_{SK}(c)$$

Should be efficient to compute PK, SK pairs.
Publishing PK does not give any information ab
computationally infeasible to get SK from PK .

2 /07/2025

PKE - Diffie/Hellman Ideas

Every party has a public key PK and a privat

nybody who wants to send a message m to u
encrypt:

$$c = Enc_{PK}(m)$$

will use SK to decrypt:

$$m = Dec_{SK}(c)$$

Should be efficient to compute PK, SK pairs.
Publishing PK does not give any information ab
computationally infeasible to get SK from PK .
Digital Signatures – sign with SK and verify wi

2 /07/2025

Public Key Encryption

c

lice

B

Encrypting
using Bob's public key:
$c = Enc_{PK_B}(m)$

Decry
using Bob's
$m = De$

Digital Signatures using Public Ke

(m, s)

Bob

Signing
using Bob's private key:
$s = Sign_{SK_B}(m)$

Ver
using Bob'
$Verify_P$

Public Key Encryption – RS

How to implement Diffie/Hellman ideas? – Rivest, Sh dleman in 1977 came with the RS algorithm.

Public Key Encryption – RS

How to implement Diffie/Hellman ideas? – Rivest, Sh
dleman in 1977 came with the RS algorithm.

Note: In 1999, it was revealed that Ellis, Cocks and W
invented PKC in the British secret service, before thei
outside.

Public Key Encryption – RS

Idea : reduce the problem of finding the secret ke
known hard mathematical problem.

Public Key Encryption – RS

Idea : reduce the problem of finding the secret ke
known hard mathematical problem.

RS security relies (in part) on the conjectured h
factoring n, a product of two very large primes p.

2 /07/2025

Public Key Encryption – RS

- Idea : reduce the problem of finding the secret ke
known hard mathematical problem.
- RS security relies (in part) on the conjectured h
factoring n, a product of two very large primes p
- What we look for in any PKC are one-way functi
trapdoor:
  - Function f must be invertible so as to decrypt e
messages.
  - Efficient to encrypt
  - Difficult to invert so that Eve cannot compute m
  - Has a trapdoor: given some information (SK)
easy given $f(m)$.

PKC revolution

Huge implications - exponential rise in study and
cryptography.

2 /07/2025

PKC revolution

- Huge implications - exponential rise in study and cryptography.
- Wide scale deployment of crypto systems

PKC revolution

- Huge implications - exponential rise in study and cryptography.
- Wide scale deployment of crypto systems
- More Turing award winners in theoretical crypto & Micali (2012)

PKC revolution

- Huge implications - exponential rise in study and
cryptography.
- Wide scale deployment of crypto systems
- More Turing award winners in theoretical crypto
& Micali (2012)
- More interesting ideas like for e.g:
  - secret sharing : collaboration between distrustin
  - zero knowledge proofs (ZKPs) : revealing nothi
validity of the statement
  - fully homomorphic encryption : computation on
- I CR - Intl. ssn. for Crypto Research. Sponsors
crypto conferences – Crypto, Eurocrypt and sia

Principles of Modern Cryptograph

The aim is to move away from art of solving cod
science/mathematics for securing digital informat
adversarial attacks.

2 /07/2025

Principles of Modern Cryptograph

The aim is to move away from art of solving cod
science/mathematics for securing digital informat
adversarial attacks.

Principle 1 - Formal Definitions

Principles of Modern Cryptograph

The aim is to move away from art of solving cod
science/mathematics for securing digital informat
adversarial attacks.

Principle 1 - Formal Definitions

Clear description of threat model and security gu
before the design process begins.

2 /07/2025

Principles of Modern Cryptograph

The aim is to move away from art of solving cod
science/mathematics for securing digital informat
adversarial attacks.

Principle 1 - Formal Definitions

Clear description of threat model and security gu
before the design process begins.

You need to know what you have to achieve bef

2 /07/2025

Principles of Modern Cryptograph

The aim is to move away from art of solving cod
science/mathematics for securing digital informat
adversarial attacks.

Principle 1 - Formal Definitions

Clear description of threat model and security gu
before the design process begins.

You need to know what you have to achieve bef
You do not want to work with intuitive ideas.

2 /07/2025

Principles of Modern Cryptograph

The aim is to move away from art of solving cod
science/mathematics for securing digital informat
adversarial attacks.

Principle 1 - Formal Definitions

Clear description of threat model and security gu
before the design process begins.

- You need to know what you have to achieve bef
  You do not want to work with intuitive ideas.

It helps analyze and evaluate the scheme and son
security too!

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover
$Enc_k(m) = m$ is not secure.

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover
$Enc_k(m) = m$ is not secure.

t should be impossible for an attacker to recover
plaintext from the ciphertext.

2 /07/2025

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover
$Enc_k(m) = m$ is not secure.

t should be impossible for an attacker to recover
plaintext from the ciphertext. Not good enough!

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover
$Enc_k(m) = m$ is not secure.

t should be impossible for an attacker to recover
plaintext from the ciphertext. Not good enough!

t should be impossible for an attacker to recover
character of the plaintext from the ciphertext.

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover
$Enc_k(m) = m$ is not secure.

t should be impossible for an attacker to recover
plaintext from the ciphertext. Not good enough!

t should be impossible for an attacker to recover
character of the plaintext from the ciphertext. V
some relations?

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover
$Enc_k(m) = m$ is not secure.

t should be impossible for an attacker to recover
plaintext from the ciphertext. Not good enough!

t should be impossible for an attacker to recover
character of the plaintext from the ciphertext. V
some relations?

right answer: *regardless of any information an*
already has, a ciphertext should leak no addition
about the underlying plaintext.

Formal Definitions - n example

Example : What does one mean by secure encryp

t should be impossible for an attacker to recover
$Enc_k(m) = m$ is not secure.

t should be impossible for an attacker to recover
plaintext from the ciphertext. Not good enough!

t should be impossible for an attacker to recover
character of the plaintext from the ciphertext. V
some relations?

right answer: *regardless of any information an*
already has, a ciphertext should leak no additiona
about the underlying plaintext.

What is prior knowledge? What is leak?

Principle 2 - Precise ssumptions

Typically security is not proved unconditionally as
seen in this lecture so we rely on assumptions.

Principle 2 - Precise ssumptions

Typically security is not proved unconditionally as
seen in this lecture so we rely on assumptions.

They are not proven but conjectured to be true.

Principle 2 - Precise ssumptions

- Typically security is not proved unconditionally as seen in this lecture so we rely on assumptions.
- They are not proven but conjectured to be true.
- This means the assumptions should be examined be true.

Principle 2 - Precise ssumptions

Typically security is not proved unconditionally as
seen in this lecture so we rely on assumptions.

They are not proven but conjectured to be true.

This means the assumptions should be examined
be true.

If it is not refuted for many years our confidence
increased.

Principle 2 - Precise ssumptions

- Typically security is not proved unconditionally as
seen in this lecture so we rely on assumptions.
- They are not proven but conjectured to be true.
- This means the assumptions should be examined
be true.
- If it is not refuted for many years our confidence
increased. This means precisely and simply statin
one will study it!

Principle 2 - Precise ssumptions

- Typically security is not proved unconditionally as
seen in this lecture so we rely on assumptions.
- They are not proven but conjectured to be true.
- This means the assumptions should be examined
be true.
- If it is not refuted for many years our confidence
increased. This means precisely and simply statin
one will study it!
- ssumptions give us a way of comparing two sch
on two different assumptions.

Principle 2 - Precise ssumptions

Typically security is not proved unconditionally as
seen in this lecture so we rely on assumptions.

They are not proven but conjectured to be true.

This means the assumptions should be examined
be true.

If it is not refuted for many years our confidence
increased. This means precisely and simply statin
one will study it!

ssumptions give us a way of comparing two sch
on two different assumptions.

If an assumption is broken (like factoring on a qu
computer) the schemes built on the assumption (
ECDS) will break!

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain
under certain assumptions.

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain
under certain assumptions. I.e. no adversary with
specified resources can break the scheme because
entail the underlying assumption to be false.

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain
under certain assumptions. I.e. no adversary with
specified resources can break the scheme becaus
entail the underlying assumption to be false.

Cryptography is still an art!

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain
under certain assumptions. I.e. no adversary with
specified resources can break the scheme because
entail the underlying assumption to be false.

Cryptography is still an art! Creativity needed for
new definitions, new schemes and proving their s

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain under certain assumptions. I.e. no adversary with specified resources can break the scheme because entail the underlying assumption to be false.

Cryptography is still an art! Creativity needed for new definitions, new schemes and proving their s for attacking deployed cryptosystems even if they secure,

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain
under certain assumptions. I.e. no adversary with
specified resources can break the scheme because
entail the underlying assumption to be false.

Cryptography is still an art! Creativity needed for
new definitions, new schemes and proving their se
for attacking deployed cryptosystems even if they
secure, i.e. there is a difference between provable
real-world security.

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain
under certain assumptions. I.e. no adversary with
specified resources can break the scheme because
entail the underlying assumption to be false.

Cryptography is still an art! Creativity needed for
new definitions, new schemes and proving their se
for attacking deployed cryptosystems even if they
secure, i.e. there is a difference between provable
real-world security.

Not a drawback. It means the definitions or assu
being broken!

2 /07/2025

Principle 2 - Proofs of Security

Rigorous proof shows a scheme satisfies a certain
under certain assumptions. I.e. no adversary with
specified resources can break the scheme because
entail the underlying assumption to be false.

Cryptography is still an art! Creativity needed for
new definitions, new schemes and proving their s
for attacking deployed cryptosystems even if they
secure, i.e. there is a difference between provable
real-world security.

Not a drawback. It means the definitions or assu
being broken!

Pegasus shows us that well-resourced targeted at
impossible to prevent! Read :
`https://blog.cryptographyengineering.co`
`20/a-case-against-security-nihilism/.`

The future of the field

Post Quantum Cryptography – In 1994, Peter Sh
fast factorization algorithm that runs in polynom
(hypothetical) quantum computer.

The future of the field

Post Quantum Cryptography – In 1994, Peter Sh
fast factorization algorithm that runs in polynom
(hypothetical) quantum computer. So now we ne
alternatives to RS .

The future of the field

- Post Quantum Cryptography – In 1994, Peter Sh
fast factorization algorithm that runs in polynom
(hypothetical) quantum computer. So now we ne
alternatives to RS .
- Security over Blockchains (using ZKPs)

The future of the field

- Post Quantum Cryptography – In 1994, Peter Sh fast factorization algorithm that runs in polynom (hypothetical) quantum computer. So now we ne alternatives to RS .
- Security over Blockchains (using ZKPs)
- Voting systems

The future of the field

- Post Quantum Cryptography – In 1994, Peter Sh
fast factorization algorithm that runs in polynom
(hypothetical) quantum computer. So now we ne
alternatives to RS .
- Security over Blockchains (using ZKPs)
- Voting systems
- ttacks on different cryptographic primitives like
functions.

2 /07/2025

Conclusion

Cryptography is not THE solution to security issu

Conclusion

Cryptography is not THE solution to security issu

Security involves a lot of other things like practic software and hardware engineering, policy, huma

Conclusion

- Cryptography is not THE solution to security issu
- Security involves a lot of other things like practic
software and hardware engineering, policy, human
- Cryptography is an essential component of any so
correctly used can help us greatly enhance secur

Conclusion

- Cryptography is not THE solution to security issu
- Security involves a lot of other things like practic software and hardware engineering, policy, humar
- Cryptography is an essential component of any sc correctly used can help us greatly enhance secur
- Lots of cryptographic tools are not used in practi

2 /07/2025

Conclusion

Cryptography is not THE solution to security issu

Security involves a lot of other things like practic software and hardware engineering, policy, humar

Cryptography is an essential component of any sc correctly used can help us greatly enhance secur

Lots of cryptographic tools are not used in practi

But all in all, cryptography is an interesting field abstract ideas from mathematics and builds secu using the tools of hardware and software enginee

2 /07/2025

Conclusion

Cryptography is not THE solution to security issu

Security involves a lot of other things like practic
software and hardware engineering, policy, human

Cryptography is an essential component of any s
correctly used can help us greatly enhance secur

Lots of cryptographic tools are not used in practi

But all in all, cryptography is an interesting field
abstract ideas from mathematics and builds secu
using the tools of hardware and software enginee

nd of course, there is cryptanalysis!

Question for Tutorial

If $f(n)$ is non-negligible and $g(n)$ is negligible, then $h(n) = f(n) \quad g(n)$ is non-negligible.