4/0 /2025

S 6160 ryptology Lectu
Some Discrete Mathemat

Maria Francis

ugust 11, 2025

Sets

set M is a well-defined collection of distinct ob

$x \in M$ means x is a member of M otherwise $x \notin$

$N \subset M$ implies N is a subset of M, i.e. all the el
are also elements of M.

Examples – $\emptyset = \{\}$, $\{0\}$, $\{0, 1\}$, $\{\ \ , B, C, \ldots, Z\}$

Sets

$\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq$.

Cardinality/Size – of a finire set X is the number
elements and is denoted by $|X|$.

Size can have different notions – size of an intege
number of bits needed to represent it, size of a b
its length.

Representation of sets can be done either by expl
out elements or by intervals or implicitly using fo

- $M = \{x \in \mathbb{Z} : x^4 < 50\}$ describes the set of inte which $x^4 < 50$ holds, or equivalently, $M = \{ \quad 2,$
- $M = \{0, 1\}^{12}$ – elements in M can either be wr form $b_1, b_2, \ldots, b_{12}$ or as a vector $(b_1, b_2, \ldots,$

Set Operations

$M \cup N$ - union of two sets

$M \cap N$ - intersection of sets

$M \setminus N$ - set difference

$M \times N$ - Cartesian product of two sets

M^n - n-ary product

$\mathcal{P}(M)$ - power set

Intuition on size of numbers – wha practical and what is not!

- One can easily store one terabyte – $10^{12} \approx 2^{43}$ b
- On the other other hand, a large amount of resou required to store one extra million bytes, $\approx 2^{63}$ bi bits are impossible.
- Same with computing steps : $< 2^{40}$ steps/CPU c easily doable.
- 2^{60} requires a lot of computing power and signific of time.
- $> 2^{100}$ infeasible, for example : it is impossible to different keys with non quantum computers.

Functions

function/mapping/map $f : X \to Y$ consists of
domain X and codomain Y and a rule which assi
image $y = f(x) \in Y$ for every input element $x \in$
The set of all $f(x)$ is a subset of Y and called ra
ny $x \in X$ with $f(x) = y$ is called a preimage of
Let $B \subset Y$ then we say that $f\ ^{1}(B) = \{x \in X :$
the preimage/inverse image of B under f.

Note : mathematical function is not the same
algorithm, which is a step by step and possibly re
procedure which produces an output from a given
There can be different algorithms or no known al
for a given mathematical function.

lso algorithms can use random data making sur
different behaviour on different runs of the same

mathematical function is different from the fun
programming languages.

Different types of function

Every set has an identity function, $id_X : X \to X$
$x \in X$ to itself.

Functions can be composed if the range of the fi
lies within the domain of the second function.

Let $f : X \to Y$ and $g : Y \to Z$ be functions, the
a composite function

$$g \circ f : X \to Z \text{ defined as } g \circ f(x) = g($$

Let $f : X \to Y$ be a function, then f is injective
one-one/1-1 if different elements of the domain
different elements of the range, i.e.

$$\forall x_1, x_2 \in X, \text{ with } x_1 \neq x_2 \text{ we have } f(x_1)$$

Different types of function

Equivalently, $\forall x_1, x_2 \in X \; f(x_1) = f(x_2) \Rightarrow x_1 =$

function $f : X \to Y$ is surjective/onto if every
the codomain Y is contained in the image/range
$im(f) = Y$.

f is bijective if it is both injective and surjective.

Bijective functions posses an inverse map $f^{\ 1} :$
$f^{\ 1} \circ f = id_X$ and $f \circ f^{\ 1} = id_Y$.

Such functions are called invertible.

Bijective maps play an important role in ciphering
encryption map E must have a corresponding de
D s.t. $D \circ E = id_{\mathcal{M}}$, the identity map on the pla

Encryption map has to be injective and decryptio
be surjective. Ciphering is often a composition o
operations, each of them not necessarily bijective
compositions are ultimately bijective.

Different types of function

Let $f : X \to Y$ be a map between finite sets. Then,

- If f is injective then $|X| \leq |Y|$.
- If f is surjective then $|X| \geq |Y|$
- If f is bijective then $|X| = |Y|$

If the function is from the same set to itself we call it
permutation.

Different types of function

Let $f : X \to Y$ be a map between finite sets. Then,

- If f is injective then $|X| \leq |Y|$.
- If f is surjective then $|X| \geq |Y|$
- If f is bijective then $|X| = |Y|$

Note that all are necessary and not sufficient.

If the function is from the same set to itself we call it permutation.

Pigeonhole principle

If $|X| > |Y|$ then f is not injective.
Suppose X is a set of some pigeons and Y a set of ho
are more pigeons than holes, then if you are trying to
all the pigeons in X to some hole (which is what a fu
then at least one hole has > 1 pigeon.

Relations

binary relation/relation between X and Y is a
$R \subset X \times Y$. I.e. It is a set of pairs (x, y), $x \in X$

function $f : X \to Y$ defines a relation between
$R = \{(x, y) \in X \times Y : y = f(x)\}$.
R contains all tuples $(x, f(x))$ with $x \in X$.

Equivalence relation on a set X are special relatio
induce a partition on X.

Relations

Let R be a relation on X, then R is called an equ
relation if it satisfies the following conditions:

R is reflexive, i.e. $(x, x) \in R$, $\forall x \in X$
R is symmetric, i.e if $(x, y) \in R$ then $(y, x) \in R$
R is transitive i.e. if $(x, y) \in R$ and $(y, z) \in R$ t

If R is an equivalence relation and $(x, y) \in R$ the
called equivalent and we write $x \sim y\}$.

For $x \in X$, the subset $\overline{x} = \{y \in X : x \sim y\} \subseteq X$
equivalence class of x and all elements in $\overline{x}$ are re
of that class.

The equivalence classes are disjoint and their uni

The set of equivalence classes is the quotient set

Equivalence Classes

The elements of $X/\sim$ are sets, but sometimes w
same x to represent both an element of X and o
Correct notation is $[x]$.

We define an equivalence relation R_2 on $X = \mathbb{Z}$:

$$R_2 = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} : (x \quad y) \in 2\mathbb{Z}$$

i.e. pairs $(x, y) \in R_2$ have the property that their
$(x \quad y)$ is even, i.e. divisible by 2.

$(2, 4), (3, 1), (\ \ 1, 5)$ are all elements of R_2 but
$(0, 1), (\ \ 3, 4) \notin R_2$

Exercise – R_2 is an equivalence relation.

Equivalence Classes

The equivalence class of $x \in \mathbb{Z}$ is $[x]$ or $\overline{x}$ is $\{\ldots, (x \quad 4), (x \quad 2), x, (x+2), (x+4), \ldots\}$.

There are only two different equivalence classes:

$$\overline{0} = \{\ldots, \quad 4, \quad 2, 0, 2, 4, \ldots\}$$
$$\overline{1} = \{\ldots, \quad 3, \quad 1, 1, 3, 5, , \ldots\}$$

I.e. $\quad \overline{2} = \overline{0} = \overline{2}$ and $\quad \overline{1} = \overline{1} = \overline{3}$.

$\mathbb{Z}/\sim$ only has two elements, which is often deno
$\mathbb{Z}/\langle 2 \rangle$, $GF(2)$ or $\mathbb{F}_2$, field of residue classes modu
be "identified" with the binary set $\{0, 1\}$.

Residue Classes modulo

Can we generalize to residue classes modulo n. L
$n \geq 2$.

Consider R_n on $X = \mathbb{Z}$,

$$R_n = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} : x \quad y \in n\mathbb{Z}\}$$

It is an equivalence relation. There are n differen
classes and thus $\mathbb{Z}/\sim$ has n elements.

We call this set the residue classes modulo n or i
n and denote it as $\mathbb{Z}_n$ or $\mathbb{Z}/(n)$.

Each residue class has a standard representative i
$\{0, 1, \ldots, n \quad 1\}$ and elements in the same resid
called congruent modulo n.

Fields

field is a set with two binary operations – called add
multiplication – which satisfy field axioms:

- ssociativity of addition and multiplication
- Commutativity of addition and multiplication
- dditivity and multiplicative identity exist
- dditive inverse exists
- Multiplicative inverse exists for all non zero eleme
- Distributivity

commutative ring where $0 \neq 1$ and all non zero elem
invertible under multiplication.

Residue classes modulo 2 form a field. Exercise – Do r
modulo n, for any $n \in \mathbb{N}$ form a field?

Exercises in this class

Show that R_2 is an equivalence relation.

Do residue classes modulo n, for any $n \in \mathbb{N}$ form
yes, why? If no, then what conditions on n make

$a \equiv c \bmod n$ iff $[a] = [c]$.

ny two congruence classes mod n are either equ

There are exactly n congruence classes modulo n
$[0], [1], \ldots, [n \quad 1]$.

Prove that for $\mathbb{Z}/n\mathbb{Z}$, the binary operations $+$ mo
modulo n are well-defined.

Modular Exponentiation

Lets work in $\mathbb{Z}/100\mathbb{Z}$.

$$
\begin{aligned}
2^2 &= 4\\
2^4 &= (2^2)^2 \equiv 4^2 \equiv 16 \bmod 100\\
2\ \ &= (2^4)^2 \equiv 16^2 \equiv 56 \bmod 100\\
2^{16} &= (2\ \ )^2 \equiv 56^2 \equiv 36 \bmod 100\\
2^{32} &= (2^{16})^2 \equiv 36^2 \equiv \ \ 4 \bmod 100\\
2^{64} &= (2^{32})^2 \equiv (\ \ 4)^2 \equiv 16 \bmod 100
\end{aligned}
$$

Now $100 = 64 + 32 + 4$ so
$2^{100} = 2^{64} \cdot 2^{32} \cdot 2^4 \equiv 16 \cdots (\ \ 4) \cdot 16 \equiv 76 \bmod$
$2^{100} \equiv 76 \bmod 100$, and this requires only $6 + 3$
multiplications instead of 100.

It is possible to calculate $a^N \bmod n$ using only c
multiplications for some constant c. This means
to calculate a^N even if N is very large and has th
digits.

pplication – Diffie Hellman Key Exchange

- How does lice and Bob establish a secret key in place without meeting beforehand?
- We know that given a and N it is easy calculate
- But given a^N mod n and a it is very hard to find of the discrete logarithm problem.
- Why call it logarithm? Because over $\mathbb{R}$ if $a^N = b$ $N = log_a(b)$.
- But this is easy to do over $\mathbb{R}$ but over natural nu mod n it is not easy to find "logs".

pplication – Diffie Hellman Key Exchange

If $2^N \equiv 3 \bmod 11$, find N. We have to try out al
and $N = 8$.

What if N and n are 10^{100} then its a hopeless tas

The Key Exchange algorithm:

- lice and Bob publicly choose a large prime p a
- lice secretly chooses a number s and sends a^s
- Bob secretly chooses a number t and sends a^t n
- lice computes secretly $(a^t)^s \bmod p$ and Bob se
  computes $(a^s)^t \bmod p$.
- The shared key is $a^{st} \bmod p$,

4/0 /2025

pplication – Diffie Hellman Key Exchange

lice and Bob should not reveal s, t, k to anyone
Eve can see a^s and a^t but cannot find s or t to c
Note that there is no information theoretic securi
given ample time Eve can also find s, t. We have
is very large so that the running time is trillions o
effectively k is safe.

Inverses in $\mathbb{Z}$ $\mathbb{Z}$

Let $a \in \mathbb{Z}/n\mathbb{Z}$. solution $x \in \mathbb{Z}/n\mathbb{Z}$ of the equat

$$ax \equiv 1 \text{ mod } n$$

is called inverse of a mod n and denoted as a ^{1}.

a is invertible in $\mathbb{Z}_n$ iff $gcd(a, n) = 1$.

a is invertible implies $\exists x \in \mathbb{Z}_n$ s.t. $ax \equiv 1$ mod
This is true if and only if $\exists y \in \mathbb{Z}$ s.t. $ax + ny =$
But the above equation is solvable for x and y if
– Exercise! (Hint : Use Extended Euclidean lg

Exercise – Let p be a prime number, then every
element $\mathbb{Z}_p$ is invertible.

This makes $\mathbb{Z}_p$ similar to real numbers, it is also

We will discuss Euler Totient function, Fermat's
theorem, CRT later in the course.

Complexity Classes

conservative approach to computing devices, a
efficient computations with the complexity class

In cryptography we take a more liberal approach
computing devices to be randomized.

Complexity Class P: language L is recognizable
polynomial time if there exists a deterministic Tu
M and a polynomial $p(\cdot)$ s.t.

on input a string x, machine M halts after at m
and
$M(x) = 1$ if and only if (iff) $x \in L$.

P is the class of languages/problems that can be
deterministic polynomial time.

Complexity Classes

NP : complexity class NP associated with compu
problems with solutions that once given can be te
validity.

lso sometimes defined as class of languages tha
recognized by a non-deterministic polytime TM.

More formally, language L is in NP if there exi
relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ and a polynomial
can be recognized in deterministic polynomial tim
iff $\exists y$ s.t. $|y| \leq p(|x|)$ and $(x, y) \in R$.

Such a y is called a witness for membership of x

Thus NP consists of the set of languages for whi
short proofs of membership that can be efficientl

It is widely believed $P \neq NP$.

Complexity Classes

NP-completeness: language is NP-complete if and every language in NP is polynomially reducib

What does polynomial reduction mean? langua polynomially reducible to a language L' if there e polynomial time computable function f s.t. $x \in$ $f(x) \in L'$.

Note : This implies L' is at least as hard as L, i.e. difficult than L.

If you have a way of solving for L' then we can u $f(x)$ and solve for $x \in L$ or not.

Example : Satisfiability of formula, graph colourir hamiltonicity.

In the above definition if the problem does not ha NP then we get NP-hard. So in some sense, NP- the hardest problem in NP.

Randomized lgorithms

We present a simple randomized algorithm – algo
use random bits in logic or procedure – for decidi
not a given undirected graph is connected.

Interesting because it uses significant less space t
standard BFS or DFS based deterministic algorit

There may or may not be error in the outputs of
algorithms (Las Vegas algorithms give correct ou
Montecarlo algorithms have a probability of error
Miller Rabin primality test.)

Whether or not a graph is connected is easily red
testing connectivity between any pair of vertices.

Space complexity is reduced because only two ve
stored.

Time complexity is high since we need to invoke
vertex tester $C(n, 2)$ times, where n is the numbe

Randomized lgorithms

On input $G = (V, E)$ and two vertices s, t we ta
walk of length $\mathcal{O}(|V||E|)$ starting at s and test a
whether or not t is encountered.

If t is encountered then algorithm will ccept, els

By random walk, we mean that at each step we
select one of the edges incident at the current ve
traverse the edge to the other end point.

Clearly, if s is not connected to t in G the probal
algorithm will accept is 0.

The harder part is to prove that if s is connected
algorithm will accept with probability $\geq 2/3$.

Either way the algorithm will error with probabilit
can be brought down further by running the algo
times.

Randomized lgorithms

- Outcome of internal coin tosses as a transition of
  by $(<\mathit{state}>, <\mathit{symbol}>, <\mathit{direction}>)$ inste
  $(<\mathit{state}>, <\mathit{symbol}>)$ with each direction hav
  probability.
- Or view the outcome of internal coin tosses as ar
  input.
- Probabilistic Polynomial Time Turing Machine m
  probabilistic machine that always – independent o
  outcome of its internal coin tosses – halts after a
  in the length of the input – number of steps.
- It also follows that the coin tosses for a PPT TM
  by a polynomial in its input length.

ssociating Efficient Computation
BPP

- We want to consider only efficient randomized al which the running time is bounded by a polynom length of the input.
- This is captured by the complexity BPP, bounde polynomial time.
- Formally, we say that L is recognized by the PPT
  - $\forall x \in L$ it hold that $Pr[M(x) = 1] \geq 2/3$ and
  - $\forall x \notin L$ it holds that $Pr[M(x) = 0] \geq 2/3$.
- BPP is the class of languages that can be recogn probabilistic polynomial time TM.
- Is BPP = P? Open question!
- No relation between BPP and NP is known!

Size of integer

Relation between positive integer n and it size $size(n) = \lfloor log_2 n \rfloor + 1$.

The size of an integer is the number of bits that
represent it.

Let $n = 10^{24}$, $size(n) = \lfloor 24 log_2 10 \rfloor + 1 = 80$ bits

Unary representation will need 10^{24} bits to repres
number, not efficient.

Consider the algorithm that finds the prime diviso
integer n. We can use an algorithm that does div
odd positive integers $\leq \sqrt{n}$.

The number of divisors is at most $\lfloor \frac{1}{2}\sqrt{n} \rfloor$, and si
$\sqrt{n} = 2^{\frac{log_2 n}{2}}$, the worst case running time is $\mathcal{O}(2^{\frac{si}{}}$
subexponential.

Note that $\frac{1}{2}$ in the exponent can not be removed
notation.

Discrete Probability

We consider discrete probability spaces here. Let
countable set, $\mathcal{S} = \mathcal{P}(\ )$, the powerset of and
$Pr : \mathcal{S} \to [0, 1]$, a function with the following pro

- $Pr[\ ] = 1$.
- If $_1, \ _2, \ _3, \ldots$ are pairwise disjoint subsets of

$$Pr[\cup_i \ _i] = \sum_i Pr[\ _i].$$

The triple $(\ , \mathcal{S}, Pr)$ is called a discrete probabili

is called the sample space, Pr the discrete prol
distribution on .

The subsets $\ \subset \ $ are said to be events and $Pr[$
probability of .

Uniform Distribution

- The sets/events in the second condition are eithe number or countably infinite, since all probabilitie negative and bounded by 1, the series converges.
- Uniform Distribution: Pr is a uniform distribution singleton events have equal probability, i.e. $Pr[\{\omega$ all $\omega \in$.
- Examples include a fair coin or fair dice. Random generators should produce uniformly distributed r
- In cryptographic algorithms we assume a key is c uniformly at random so that all possible keys hav probability.

Independent Events

Let , B and $_1, _2, \dots, _n$ be events in a proba

and B are said to be independent if the joint
equals the product of probabilities,

$$Pr[\ \cap B] = Pr[\ ]Pr[B].$$

$_1, _2, \dots, _n$ are mutually independent if for e
subset of indices, $\mathcal{I} \subset \{1, \dots, n\}$ one has

$$Pr[\cap_{i\ \mathcal{I}}\ _i] = \prod_{i\ \mathcal{I}} Pr[\ _i].$$

Note that mutual independence of n events is str
pairwise independence of all pairs.

Mutually independent events Vs P
independent

Let X_1 and X_2 be two binary random variables (w
RVs more formally soon) that are given by tossin
coins so that X_1 and X_2 are independent.

Let $X_3 = X_1 \oplus X_2$.

Then X_1, X_2, X_3 are pairwise independent and ea
has a uniform distribution, but they are not mutu
independent.

$Pr[X_1 = 1 \wedge X_2 = 1 \wedge X_3 = 1] = 0$, since X_3 mus
$X_1 = X_2 = 1$.

But $Pr[X_1 = 1] \cdot Pr[X_2 = 1] \cdots Pr[X_3 = 1] = \frac{1}{2}^3$

Conditional Probability

Let and B be two events in a probability space
that $Pr[\ ] \neq 0$. Then the conditional probability
given is defined by

$$Pr[B|\ ] = \frac{Pr[\ \cap B]}{Pr[\ ]}.$$

Note: $Pr[B|\ ] = Pr[B]$ iff and B are independ

Random Variables

The sample space of a probability space is ofte
a standard space and in particular to real number
map is called a random variable.

Let $Pr : \ \rightarrow [0,1]$ be a discrete probability distr
function $X : \ \rightarrow \mathbb{R}$ is called a real random varia

For any $x \in \mathbb{R}$ one obtains an event $X^{\ 1}(x) \subset$
probability $Pr[X = x]$ is defined by $Pr[X^{\ 1}(x)]$.
$p_X : \mathbb{R} \rightarrow [0,1]$ is defined by $p_X(x) = Pr[X = x]$
probability mass function/pmf of X.

Cumulative distributive function $F : \mathbb{R} \rightarrow \mathbb{R}$ is de
$F(x) = Pr[X \leq x]$.

X induces a discrete probability distribution Pr_X
countable subset $X(\ ) \subset \mathbb{R}$, with the sample spa
being a subset of $\mathbb{R}$.

Random Variables

Let $X_1, X_2, \ldots, X_n$ be random variables on discre
space with pmf $p_{X_1}, p_{X_2}, \ldots, p_{x_n}$. The random va
called mutually independent if for any sequence x

$Pr[X = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_n = x_n] = p_{X_1}(x_1) \cdot p_X$

Let $\quad = \{0, 1\}$ be a space of plaintext and ciphe
the plaintexts X are uniformly distributed. Let
bit permutation. I.e. only positions are permuted
If we assume $Y = \quad (X)$ as the ciphering algorith
see that Y is also uniformly distributed.
re X and Y mutually independent? i.e. is
$Pr[X = m \wedge Y = c] = Pr[X = m] \cdot Pr[Y = c]$
$\forall x \in X, y \in Y$?
No! if m and c have a different number of 1s the
since it is impossible for a bit permutation. Thus
secure cipher.

Random Numbers

Random numbers/ random bits play an importan
cryptography.

Keys (or seeds for the keys) are chosen uniformly
and many cryptographic algorithms are probabilis
require random inputs.

random bit generator (RBG) is a mechanism/d
generates a sequence of random bits, s.t. the cor
sequence of binary random variables $X_1, X_2, X_3, .$
following properties:

- $PR[X_n = 0] = Pr[X_n = 1] = \frac{1}{2}, \forall n \in \mathbb{N}$ (uniform
- $X_1, X_2, \ldots, X_n$ are mutually independent for all
  subset of indices we consider the probability is t
  probabilities.

Some sequences which are NOT uniformly rando
$X_3 = X_1 \oplus X_2$. So obvious ways to stretch a give
cannot be used.

Random Numbers

Generated manually by coin tossing or die rolling
hardware that uses physical phenomena such as t
or electrical noise as inputs.

But they are slow, elaborate and/or costly.

There are fast all digital random bit generators o
processor chips which use thermal noise but whet
trust them or if they have a backdoor is disputed

The generation of random bits is basically equiva
generation of random integer numbers.

Birthday Paradox

Random numbers match surprisingly often, the b
paradox.

ssume there are 23 people in the a room. Then
probability that at least two of them have their b
same day of the year is $> 50\%$.

Intuitively we would have thought for that to hap
$\approx \lfloor \frac{365}{2} \rfloor$ people in the room.

Why is this so? Let p be the probability that no
occurs, i.e. all birthdays are different.

Claim : p decreases exponentially as n increases.

For $n = 2$, $p = \frac{364}{365} = 0.99726$. For $n = 3$, $p =$
$= 0.99179$.

$n = 9$ $p = .89065$, $n = 13, p = 0.79249$, $n = 17$,
$n = 20, p = 57899$.

For $n = 23$, $p = 0.493$. This implies 1 $p > 0.5$

Birthday Paradox

This is the formal statement: Let Pr be a unifor
on a set of cardinality n. If we draw
$k = \lceil \sqrt{2ln(2)n} \rceil \approx 1.2\sqrt{n}$ independent samples f
the probability of a collision is $\approx 50\%$.

Consider collision of binary random strings of len

Collisions are expected after around $\sqrt{2}$ values.

So if the strings are sufficiently long and uniform
then the collisions should almost never occur.

This is why we need at least 128 bit lengths for h