Introduction to Cryptography

Prof. Dr.-Ing. Sebastian Schlesinger

Berlin School for Economics and Law

September 4, 2023

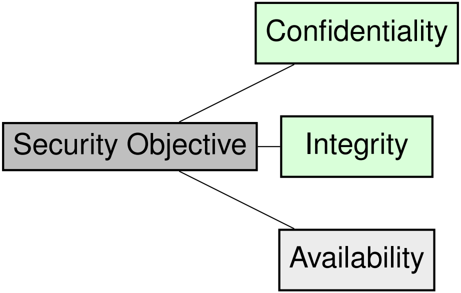
Learning objectives of this presentation

- Understanding of number theoretic basics for cryptography
- Understanding of key cryptographic concepts and algorithms

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
   - Basic numbertheoretic structures
   - Divisibility Relation and Factor Rings
   - Some important grouptheoretic results
   - Some important ringtheoretic results
4. Cryptographic Algorithms
   - Crypto Scheme
   - Symmetric Ciphers for Encryption
   - Asymmetric Ciphers for Encryption and Authentication
   - Key Exchange
   - Hash Functions for Ensuring Integrity
   - Selected Algorithms for Authentication
5. Summary

Protection Goals of Information Security



Cryptography Target

No Cryptography Target

- **Confidentiality**: only entities who are eligible are able to access data
- **Integrity**: data not maliciously altered
- **Availability**: access possible if required

Outline: Some Techniques to ensure Protection Goals for Cryptography

- **Encryption** aims at securing **confidentiality**.
- **Hash Functions** aim at ensuring **integrity** of data.
- **Digital Signatures** are used for **authentication** of an entity.

Authentication refers to the process of proving that some fact or some data is genuine. In other words: to bind a virtual entity to a real entity. In contrast, **Authorization** (out of scope of cryptography) controls the permissions to access information by entities.

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
   - Basic numbertheoretic structures
   - Divisibility Relation and Factor Rings
   - Some important grouptheoretic results
   - Some important ringtheoretic results
4. Cryptographic Algorithms
   - Crypto Scheme
   - Symmetric Ciphers for Encryption
   - Asymmetric Ciphers for Encryption and Authentication
   - Key Exchange
   - Hash Functions for Ensuring Integrity
   - Selected Algorithms for Authentication
5. Summary

Predicate Logic Notation

A formula of **predicate logic** is defined as follows:

- simple statements A, or with parameter $A(x)$ are formulas

Predicate Logic Notation

A formula of **predicate logic** is defined as follows:

- simple statements A, or with parameter $A(x)$ are formulas
- $\neg A$ meaning statement A does not hold
- $A \vee B$ meaning statement A *or* statement B hold (non exclusive or)
- $A \wedge B$ meaning statement A *and* statement B hold
- $A \Rightarrow B$ meaning that if A holds, B must hold

Predicate Logic Notation

A formula of **predicate logic** is defined as follows:

- simple statements A, or with parameter $A(x)$ are formulas
- $\neg A$ meaning statement A does not hold
- $A \vee B$ meaning statement A *or* statement B hold (non exclusive or)
- $A \wedge B$ meaning statement A *and* statement B hold
- $A \Rightarrow B$ meaning that if A holds, B must hold
- $\forall x \in M : A(x)$ meaning for all elements x from a supporting set M the statement $A(x)$ holds

Predicate Logic Notation

A formula of **predicate logic** is defined as follows:

- simple statements A, or with parameter $A(x)$ are formulas
- $\neg A$ meaning statement A does not hold
- $A \vee B$ meaning statement A *or* statement B hold (non exclusive or)
- $A \wedge B$ meaning statement A *and* statement B hold
- $A \Rightarrow B$ meaning that if A holds, B must hold
- $\forall x \in M : A(x)$ meaning for all elements x from a supporting set M the statement $A(x)$ holds
- $\exists x \in M : A(x)$ meaning there exists an element x from a supporting set M such that the statement $A(x)$ holds

Questions

- Consider the statement ”when it rains, the street is wet”. Does this imply ”The street is wet, hence it rains.”?
- Can you conclude (based on the statement) that if the street is not wet, it doesn’t rain?
- Formalize the statement ”there exists one and only one x with $A(x)$” as formula in predicate logic.

Sets

Sets are collections of arbitrary items of our reasoning (informal defition of naive set theory). We denote them in the following ways:

- $M = \{0, 1, 2, ...\}$, i.e., by explicitly enumerating the elements
- $M = \{x \in N | A(x)\}$, i.e., by defining a predicate $A(x)$, which specifies the condition under which the elements are to be taken from the supporting set N.

Relations and Functions

For two sets M, N, we define the **Cartesian Product** as the set of pairs, denoted by

$$M \times N = \{(x, y) | x \in M, y \in N\}$$

Relations and Functions

For two sets M, N, we define the **Cartesian Product** as the set of pairs, denoted by

$$M \times N = \{(x, y) | x \in M, y \in N\}$$

A **relation** R between two sets M, N is a subset of the Cartesian Product, i.e., $R \subseteq M \times N$. For two elements x, y in relation R, we denote either $(x, y) \in R$ or in the infix notation: xRy. An example is the canonical $\leq$ relation, typically denoted in infix style.

Relations and Functions

For two sets M, N, we define the **Cartesian Product** as the set of pairs, denoted by

$$M \times N = \{(x, y) | x \in M, y \in N\}$$

A **relation** R between two sets M, N is a subset of the Cartesian Product, i.e., $R \subseteq M \times N$. For two elements x, y in relation R, we denote either $(x, y) \in R$ or in the infix notation: xRy. An example is the canonical $\leq$ relation, typically denoted in infix style.
A **function** f from M to N, denoted $f : M \to N$ is a relation for which each $x \in M$ has exactly one $y \in N$ such that $(x, y) \in f$, which is then denoted by $y = f(x)$ or $x \mapsto y$.

Questions

What are examples for relations in a real-world scenario?

Attributes of Relations

Definition

Let $R \subseteq M \times M$ be a relation on the same set M. We call R

- **reflexive** if $\forall x \in M : (x, x) \in R$

Attributes of Relations

Definition

Let $R \subseteq M \times M$ be a relation on the same set M. We call R

- **reflexive** if $\forall x \in M : (x,x) \in R$
- **symmetric** if $\forall x,y \in M : (x,y) \in R \Rightarrow (y,x) \in R$

Attributes of Relations

Definition

Let $R \subseteq M \times M$ be a relation on the same set M. We call R

- **reflexive** if $\forall x \in M : (x, x) \in R$
- **symmetric** if $\forall x, y \in M : (x, y) \in R \Rightarrow (y, x) \in R$
- **antisymmetric** if $\forall x, y \in M : (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y$

Attributes of Relations

Definition

Let $R \subseteq M \times M$ be a relation on the same set M. We call R

- **reflexive** if $\forall x \in M : (x, x) \in R$
- **symmetric** if $\forall x, y \in M : (x, y) \in R \Rightarrow (y, x) \in R$
- **antisymmetric** if $\forall x, y \in M : (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y$
- **transitive** if $\forall x, y, z \in M : (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$.

Attributes of Relations

Definition

Let $R \subseteq M \times M$ be a relation on the same set M. We call R

- **reflexive** if $\forall x \in M : (x, x) \in R$
- **symmetric** if $\forall x, y \in M : (x, y) \in R \Rightarrow (y, x) \in R$
- **antisymmetric** if $\forall x, y \in M : (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y$
- **transitive** if $\forall x, y, z \in M : (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$.

A reflexive, antisymmetric and transitive relation is called an **order**. A reflexive, symmetric and transitive relation is called an **equivalence relation**.

Composition of Relations

Definition

For relations $R \subseteq M \times N, S \subseteq N \times P$, we define the **composition** of the relation

$$R \circ S = \{(x, z) \in M \times P | \exists y \in N : (x, y) \in R \wedge (y, z) \in S\}$$

Attributes of Functions

Definition

Let $f : M \to N$ be a function. We call f

- **injective** if $\forall x, y \in M : f(y) = f(x) \Rightarrow x = y$

Attributes of Functions

Definition

Let $f : M \to N$ be a function. We call f

- **injective** if $\forall x, y \in M : f(y) = f(x) \Rightarrow x = y$
- **surjective** if $\forall y \in N : \exists x \in M : y = f(x)$

Attributes of Functions

Definition

Let $f : M \to N$ be a function. We call f

- **injective** if $\forall x, y \in M : f(y) = f(x) \Rightarrow x = y$
- **surjective** if $\forall y \in N : \exists x \in M : y = f(x)$
- **bijective** if f is injective and surjective.

Inverse Image of a function

For $f: M \to N$, we denote $Im(f) = \{y \in N | \exists x \in M : y = f(x)$, the image of f. So, $f: M \to Im(f)$ is surjective.
For $A \subset Im(f)$, we denote $f^{-1}(A) = \{x \in M | \exists y \in A : y = f(x)\}$ the inverse image of f of A.

Composition of Functions

Definition

For functions $f : M \to N$, $g : N \to P$ with $Im(f) = N$, we define the **composition** of the functions

$$g \circ f : M \to P : x \mapsto g(f(x))$$

Questions

- Make yourself familiar with what the attributes of relations actually mean.
- What are examples for orders and equivalence relations?

Equivalence Relations and Quotient Sets

For an equivalence relation $\sim\subseteq M \times M$, we denote for an element $x \in M$ a set

$$[x] = \{y \in M | x \sim y\}$$

as **equivalence class** of x.

The element x in $[x]$ is called the **representative** of the equivalence class. Note that in an equivalence relation, this selection if arbitrary.

We denote

$$M/ \sim = \{[x] | x \in M\}$$

the set of equivalence classes as **quotient set** modulo $\sim$.

Partitions

A quotient set forms a **partition** on M. A partition of a set M is a family of sets P_i (i from an index set I), which fulfills the following conditions:

1. mutually exclusive: this means that sets in P_i are pairwise disjoint, i.e., $i \neq j \Rightarrow P_i \cap P_j = \emptyset$
2. collectively exhaustive: this means that the union of all P_i captures all elements of M, i.e., $\bigcup_{i \in I} P_i = M$.

This principle also works vice versa: For a partition, there exists always an equivalence relation in the described manner.

Partitions

A quotient set forms a **partition** on M. A partition of a set M is a family of sets P_i (i from an index set I), which fulfills the following conditions:

1. mutually exclusive: this means that sets in P_i are pairwise disjoint, i.e., $i \neq j \Rightarrow P_i \cap P_j = \emptyset$
2. collectively exhaustive: this means that the union of all P_i captures all elements of M, i.e., $\bigcup_{i \in I} P_i = M$.

This principle also works vice versa: For a partition, there exists always an equivalence relation in the described manner. For

$\sim \subseteq M \times M$, the projection $\pi : M \to M/\sim, x \mapsto [x]$ is surjective.

Commutative Diagram of Functions in Quotient Sets

If we have a *surjective* function $f : M \to N$ and an equivalence relation $\sim \subseteq M \times M$, we obtain the surjective projection $\pi : M \to M/\sim$.

Commutative Diagram of Functions in Quotient Sets

If we have a *surjective* function $f : M \to N$ and an equivalence relation $\sim \subseteq M \times M$, we obtain the surjective projection $\pi : M \to M/\sim$.

Then f induces a natural bijection $[f] : M/\sim \to N$ such that $f = \pi \circ f$

Commutative Diagram of Functions in Quotient Sets

If we have a *surjective* function $f : M \to N$ and an equivalence relation $\sim \subseteq M \times M$, we obtain the surjective projection $\pi : M \to M/\sim$.

Then f induces a natural bijection $[f] : M/\sim \to N$ such that $f = \pi \circ f$

$$\begin{tikzcd} M \arrow[rr, "f"] \arrow[dr, "\pi"'] & & N \\ & M/\sim \arrow[ur, "{[f]}"'] & \end{tikzcd}$$

We say, the following diagram *commutes*.
In a sense, we have *divided* the noisy details out of M to make f injective (and hence bijective) via $[f]$.

Questions

- What is $\mathbb{Z}/=$?
- Consider the relation that places all elements in a set M in relation with all other elements. Is that an equivalence relation? If so, what is the quotient set?

Exercise

Let $X = \{1, 2, 3, 4, 5, 6, 7, 8\}, Y = \{a, b, c\}$, and $f : X \to Y$ be defined by: $f(1) = a$, $f(2) = a$, $f(3) = c$, $f(4) = b$, $f(5) = a$, $f(6) = b$, $f(7) = c$, $f(8) = a$.

(i) Write down $f^{-1}(\{a\})$, $f^{-1}(\{b\})$, $f^{-1}(\{c\})$. Observe that they realize a partition of X.

(ii) Define $x_1 \sim x_2$ when $f(x_1) = f(x_2)$. Check that $\sim$ satisfies the conditions of an equivalence relation. What are the equivalence classes?

(iii) We have seen now that we have constructed an equivalence relation on X. Write down the projection function $\pi : X \to X/\sim$.

(iv) Consider the function $[f] : X/\sim \to Y$, defined by $[f]([x]) = f(x)$. Show that $[f]$ is a well defined function and that it is a bijection.

(v) Show that $f = [f] \circ \pi$.

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
    - Basic numbertheoretic structures
    - Divisibility Relation and Factor Rings
    - Some important grouptheoretic results
    - Some important ringtheoretic results
4. Cryptographic Algorithms
    - Crypto Scheme
    - Symmetric Ciphers for Encryption
    - Asymmetric Ciphers for Encryption and Authentication
    - Key Exchange
    - Hash Functions for Ensuring Integrity
    - Selected Algorithms for Authentication
5. Summary

Groups, Rings, and Fields

Groups

Let $G \neq \emptyset$, $+ : G \times G \to G$ a function. $(G, +)$ is said to be an (Abelian) group if and only if the following properties hold.

Groups, Rings, and Fields

Groups

Let $G \neq \emptyset$, $+ : G \times G \to G$ a function. $(G, +)$ is said to be an (Abelian) group if and only if the following properties hold.

- *Associativity:* $\forall x, y, z \in G : (x + y) + z = x + (y + z)$

Groups, Rings, and Fields

Groups

Let $G \neq \emptyset$, $+ : G \times G \to G$ a function. $(G, +)$ is said to be an (Abelian) group if and only if the following properties hold.

- *Associativity:* $\forall x, y, z \in G : (x + y) + z = x + (y + z)$
- *Commutativity:* $\forall x, y \in G : x + y = y + x$

Groups, Rings, and Fields

Groups

Let $G \neq \emptyset$, $+ : G \times G \to G$ a function. $(G, +)$ is said to be an (Abelian) group if and only if the following properties hold.

- *Associativity:* $\forall x, y, z \in G : (x + y) + z = x + (y + z)$
- *Commutativity:* $\forall x, y \in G : x + y = y + x$
- *Neutral Element:* There exists an alement n such that $\forall x \in G : x + n = x$ (denoted as 0 or 1 for multiplicative groups)

Groups, Rings, and Fields

Groups

Let $G \neq \emptyset$, $+ : G \times G \to G$ a function. $(G, +)$ is said to be an (Abelian) group if and only if the following properties hold.

- *Associativity:* $\forall x, y, z \in G : (x + y) + z = x + (y + z)$
- *Commutativity:* $\forall x, y \in G : x + y = y + x$
- *Neutral Element:* There exists an alement n such that $\forall x \in G : x + n = x$ (denoted as 0 or 1 for multiplicative groups)
- *Inverse Elements:* $\forall x \in G \exists y \in G : x + y = 0$. It is called *inverse* of x, also denoted as $-x$ or x^{-1} if multiplicatively denoted.

The operation on G ist sometimes additively denoted, e.g. with $+$, or multiplicatively iehter with $\cdot$ or $\circ$.

Groups, Rings, and Fields

Rings with 1

Let R be a set with at least two neutral elements $0, 1$ w.r.t. the operations $+, \cdot : R \times R \to R$. $(R, +, \cdot)$ is said to be a ring (with 1) if and only if the following properties hold.

Groups, Rings, and Fields

Rings with 1

Let R be a set with at least two neutral elements $0, 1$ w.r.t. the operations $+, \cdot : R \times R \to R$. $(R, +, \cdot)$ is said to be a ring (with 1) if and only if the following properties hold.

- $(R, +)$ is an Abelian group.

Groups, Rings, and Fields

Rings with 1

Let R be a set with at least two neutral elements $0, 1$ w.r.t. the operations $+, \cdot : R \times R \to R$. $(R, +, \cdot)$ is said to be a ring (with 1) if and only if the following properties hold.

- $(R, +)$ is an Abelian group.
- For $(R, \cdot)$ Associativity and Commutativity hold and 1 is the neutral element.

Groups, Rings, and Fields

Rings with 1

Let R be a set with at least two neutral elements $0, 1$ w.r.t. the operations $+, \cdot : R \times R \to R$. $(R, +, \cdot)$ is said to be a ring (with 1) if and only if the following properties hold.

- $(R, +)$ is an Abelian group.
- For $(R, \cdot)$ Associativity and Commutativity hold and 1 is the neutral element.
- *Distributivity:* $\forall x, y, z \in R : (x + y) \cdot z = x \cdot z + y \cdot z$.

Example

An example is the ring of integers $\mathbb{Z}$.

Groups, Rings, and Fields

Fields

Let $\mathbb{F}$ be a set with at least two neutral elements $0, 1$ w.r.t. the operations $+, \cdot : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$. $(F, +, \cdot)$ is said to be a field if and only if the following properties hold.

- $(\mathbb{F}, +)$ is an Abelian group.
- $(\mathbb{F}, \cdot)$ is an Abelian group.
- Distributivity holds.

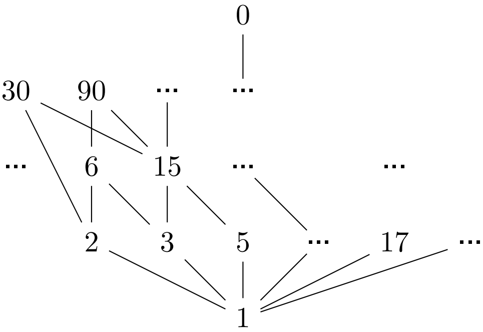
Example

Examples are the rational numbers $\mathbb{Q}$ and the real numbers $\mathbb{R}$.

Important relations on integers

Divisibility Relation

For $a, b \in \mathbb{Z}$ we define the divisibility relation $| \subset \mathbb{Z} \times \mathbb{Z}$ as $a|b :\Leftrightarrow \exists m : a \cdot m = b$. It is an order (reflexive, antisymmetric, transitive).



Important relations on integers

Equivalence mod n

For $a, b, n \in \mathbb{Z}$, we define the relation $\equiv \subset \mathbb{Z} \times \mathbb{Z}$ as $a \equiv b \mod n :\Leftrightarrow n|(a-b)$. It is an equivalence relation (reflexive, symmetric, transitive).

Important relations on integers

Equivalence mod n

For $a, b, n \in \mathbb{Z}$, we define the relation $\equiv \subset \mathbb{Z} \times \mathbb{Z}$ as $a \equiv b \mod n :\Leftrightarrow n|(a-b)$. It is an equivalence relation (reflexive, symmetric, transitive).

An equivalence class for an element $a \in \mathbb{Z}$ is defined as

$$[a] := \{b \in \mathbb{Z} | a \equiv b \mod n\}$$

Important relations on integers

Equivalence mod n

For $a, b, n \in \mathbb{Z}$, we define the relation $\equiv \subset \mathbb{Z} \times \mathbb{Z}$ as $a \equiv b \mod n :\Leftrightarrow n|(a-b)$. It is an equivalence relation (reflexive, symmetric, transitive).

An equivalence class for an element $a \in \mathbb{Z}$ is defined as

$$[a] := \{b \in \mathbb{Z} | a \equiv b \mod n\}$$

Elements in $[a]$ yield all the same residue by dividing through n. Note that

$$\forall x \in \mathbb{Z} \exists \xi, \eta \in \mathbb{Z} : x = \xi \cdot n + \eta \wedge 0 \leq \eta < n$$

and ξ, η are unique.

Factor Ring

Factor Ring

The set of equivalence classes $\mod n$ is defined as

$$\mathbb{Z}/n\mathbb{Z} := \{[a] | a \in \mathbb{Z}\} = \{[0], [1], ..., [n-1]\}$$

Factor Ring

Factor Ring

The set of equivalence classes $\mod n$ is defined as

$$\mathbb{Z}/n\mathbb{Z} := \{[a] | a \in \mathbb{Z}\} = \{[0], [1], ..., [n-1]\}$$

We define operations $+, \cdot$ on it as

$$[a] + [b] := [a+b], [a] \cdot [b] := [a \cdot b]$$

This yields a ring $(\mathbb{Z}/n\mathbb{Z}, +, \cdot)$.

Subgroups

Definition

Subgroup Let $(G, \cdot)$ be a group. A nonempty subset $U \subset G$ ist called **subgroup** of G if and only if one of the following equivalent conditions is met:

- $\forall x, y \in U : x \cdot y \in U$, $\forall x \in U : x^{-1} \in U$.

Subgroups

Definition

Subgroup Let $(G, \cdot)$ be a group. A nonempty subset $U \subset G$ ist called **subgroup** of G if and only if one of the following equivalent conditions is met:

- $\forall x, y \in U : x \cdot y \in U, \forall x \in U : x^{-1} \in U$.
- $\forall x, y \in U : x \cdot y^{-1} \in U$.

Subgroups

Definition

Subgroup Let $(G, \cdot)$ be a group. A nonempty subset $U \subset G$ ist called **subgroup** of G if and only if one of the following equivalent conditions is met:

- $\forall x, y \in U : x \cdot y \in U, \forall x \in U : x^{-1} \in U$.
- $\forall x, y \in U : x \cdot y^{-1} \in U$.

For a subgroup $(U, \cdot)$ of a group $(G, \cdot)$, the relation $x \sim y :\Leftrightarrow x \cdot y^{-1} \in U$ is an equivalence relation.

Subgroups

Definition

Subgroup Let $(G, \cdot)$ be a group. A nonempty subset $U \subset G$ ist called **subgroup** of G if and only if one of the following equivalent conditions is met:

- $\forall x, y \in U : x \cdot y \in U, \forall x \in U : x^{-1} \in U$.
- $\forall x, y \in U : x \cdot y^{-1} \in U$.

For a subgroup $(U, \cdot)$ of a group $(G, \cdot)$, the relation $x \sim y :\Leftrightarrow x \cdot y^{-1} \in U$ is an equivalence relation.

Note that since $x \sim y$ is an equivalence relation, $G/\sim$ forms a quotient set.

Subgroups

Definition

Subgroup Let $(G, \cdot)$ be a group. A nonempty subset $U \subset G$ ist called **subgroup** of G if and only if one of the following equivalent conditions is met:

- $\forall x, y \in U : x \cdot y \in U, \forall x \in U : x^{-1} \in U$.
- $\forall x, y \in U : x \cdot y^{-1} \in U$.

For a subgroup $(U, \cdot)$ of a group $(G, \cdot)$, the relation $x \sim y :\Leftrightarrow x \cdot y^{-1} \in U$ is an equivalence relation.

Note that since $x \sim y$ is an equivalence relation, $G/\sim$ forms a quotient set.

It is $G/\sim = \{x \cdot U | x \in G\}$, and via $x \cdot U \circ y \cdot U = xy \cdot U$ again a group.

Questions and Exercises

Prove the subgroup criteria!

Tranfer of the Subgroup Concept to the Factor Ring

We have defined the equivalence relation $x \equiv y \Leftrightarrow n|(x-y)$. In other words, $x \equiv y$ if $x - y \in n\mathbb{Z}$, which is a subgroup of $(\mathbb{Z}, +)$.

Tranfer of the Subgroup Concept to the Factor Ring

We have defined the equivalence relation $x \equiv y \Leftrightarrow n|(x-y)$. In other words, $x \equiv y$ if $x - y \in n\mathbb{Z}$, which is a subgroup of $(\mathbb{Z}, +)$.

Hence, $\mathbb{Z}/\equiv$ or $\mathbb{Z}/n\mathbb{Z}$ respectively form via $x + n\mathbb{Z} + y + n\mathbb{Z} = x + y + \mathbb{Z}$ a new group $(\mathbb{Z}/n\mathbb{Z}, +)$.

Tranfer of the Subgroup Concept to the Factor Ring

We have defined the equivalence relation $x \equiv y \Leftrightarrow n|(x-y)$. In other words, $x \equiv y$ if $x - y \in n\mathbb{Z}$, which is a subgroup of $(\mathbb{Z}, +)$.

Hence, $\mathbb{Z}/\equiv$ or $\mathbb{Z}/n\mathbb{Z}$ respectively form via $x + n\mathbb{Z} + y + n\mathbb{Z} = x + y + \mathbb{Z}$ a new group $(\mathbb{Z}/n\mathbb{Z}, +)$.

Moreover, we also defined the same for the multiplication (which is not a subgroup), which together forms the already known ring $(\mathbb{Z}/n\mathbb{Z}, +, \cdot)$.

Ring’s multiplikative subgroups

Ring’s multiplicative subgroup

For a ring R, we denote the set $R^* := \{x \in R | \exists y : x \cdot y = 1\}$, which is a multiplicative subgroup of $(R, \cdot)$.

Ring's multiplikative subgroups

Ring's multiplicative subgroup

For a ring R, we denote the set $R^* := \{x \in R | \exists y : x \cdot y = 1\}$, which is a multiplicative subgroup of $(R, \cdot)$.

Sufficient condition for $a \in (\mathbb{Z}/n\mathbb{Z})^*$:

$$\gcd(a, n) = 1$$

Ring's multiplikative subgroups

Ring's multiplicative subgroup

For a ring R, we denote the set $R^* := \{x \in R | \exists y : x \cdot y = 1\}$, which is a multiplicative subgroup of $(R, \cdot)$.

Sufficient condition for $a \in (\mathbb{Z}/n\mathbb{Z})^*$:

$$\gcd(a, n) = 1$$

Examples

$[1], [3] \in (\mathbb{Z}/4\mathbb{Z})^*$:
$[1]^{-1} = [1], [3]^{-1} = 3, [2] \notin (\mathbb{Z}/4\mathbb{Z})^* \; ([2] \cdot [2] = [0], [2] \cdot [3] = [2])$
$1, 2, 3, 4 \in (\mathbb{Z}/5\mathbb{Z})^*$

Finite Fields

Corollary

If p is a prime number, then $\mathbb{Z}/p\mathbb{Z}$ is a field, which we also denote as $\mathbb{F}_p$. We call those ***finite fields****.*

Finite Fields

Corollary

*If p is a prime number, then $\mathbb{Z}/p\mathbb{Z}$ is a field, which we also denote as $\mathbb{F}_p$. We call those **finite fields**.*

Finite fields play an extraordinary role in number theory and cryptography. For example elliptic curves are defined over finite fields.

Questions and Exercises

Consider $\mathbb{Z}/10\mathbb{Z}$ and $(\mathbb{Z}/10\mathbb{Z})^*$

(i) How many elements are in $(\mathbb{Z}/10\mathbb{Z})^*$?

(ii) Which elements are in $(\mathbb{Z}/10\mathbb{Z})^*$?

(iii) Depict a number line ('Zahlenstrahl') with all elements in $\mathbb{Z}/10\mathbb{Z}$ and draw the series $k \cdot x$ for x being an element in $(\mathbb{Z}/10\mathbb{Z})^*$ and $1 \leq k < n$.

(iv) Do the same for the series x^k and highlight the elements of $(\mathbb{Z}/10\mathbb{Z})^*$ on the number line.

(v) Is $(\mathbb{Z}/10\mathbb{Z})^*$ cyclic?

(vi) If it is cyclic, what is a generator for $(\mathbb{Z}/10\mathbb{Z})^*$?

Euler's phi function

Euler's Phi

We define $\varphi(n) := \#\left(\mathbb{Z}/n\mathbb{Z}\right)^*$.

Euler's phi function

Euler's Phi

We define $\varphi(n) := \#\left(\mathbb{Z}/n\mathbb{Z}\right)^*$.

Formula for calculating Euler's phi

For $n = \prod_{p|n} p^{k_p}$, we can calculate

$$\varphi(n) = \prod_{p|n} p^{k_p-1}(p-1) = \prod_{p|n}\left(1 - \frac{1}{p}\right)$$

Euler's phi function

Euler's Phi

We define $\varphi(n) := \#\left(\mathbb{Z}/n\mathbb{Z}\right)^*$.

Formula for calculating Euler's phi

For $n = \prod_{p|n} p^{k_p}$, we can calculate

$$\varphi(n) = \prod_{p|n} p^{k_p - 1}(p-1) = \prod_{p|n}\left(1 - \frac{1}{p}\right)$$

Particularly if $n = p \cdot q$, we have

$$\varphi(n) = (p-1)\cdot(q-1)$$

Cyclic groups

For a given element $g \in G$ in a group $(G, \cdot)$, we introduce $g^0 = 1, g^{i+1} = g \cdot g^i$ and

$$< g > := \{g^i | i \in \mathbb{Z}\}$$

Cyclic groups

For a given element $g \in G$ in a group $(G, \cdot)$, we introduce $g^0 = 1, g^{i+1} = g \cdot g^i$ and

$$< g >:= \{g^i | i \in \mathbb{Z}\}$$

Note that e.g. $\left(g^i\right)^{-1} = \left(g^{-1}\right)^i$ and the usual power laws also hold. $< g >$ is always a subgroup of G for each element in G.

Cyclic groups

For a given element $g \in G$ in a group $(G, \cdot)$, we introduce $g^0 = 1, g^{i+1} = g \cdot g^i$ and

$$< g > := \{g^i | i \in \mathbb{Z}\}$$

Note that e.g. $\left(g^i\right)^{-1} = \left(g^{-1}\right)^i$ and the usual power laws also hold. $< g >$ is always a subgroup of G for each element in G.

Definition (Cyclic Groups and Primitive Roots)

If the elements of a group G can be enumerated in a way $G = \{1, g, g^2, ..., g^{n-1}\}$, i.e., if there exists an element $g \in G$ such that $G =< g >$, then we call G a **cyclic** group and g a **primitive root** or generator of G.

Sufficient Conditions for Cyclic Groups in $\mathbb{Z}/n\mathbb{Z}$

Firstly, $(\mathbb{Z}/n\mathbb{Z}, +)$ is always cyclic. A generator is 1: By adding 1 to itself, you reach all elements in the group.

Sufficient Conditions for Cyclic Groups in $\mathbb{Z}/n\mathbb{Z}$

Firstly, $(\mathbb{Z}/n\mathbb{Z}, +)$ is always cyclic. A generator is 1: By adding 1 to itself, you reach all elements in the group.

However, the situation in $(\mathbb{Z}/n\mathbb{Z})^*$ is more subtle:
Gauss showed that it is cyclic if and only if n is 1, 2, 4, pk or $2pk$, where p is an odd prime and $k > 0$.

Questions and Exercises

Determine a primitive root or generator for the following groups:

- $(\mathbb{Z}, +)$
- $(\mathbb{Z}/7\mathbb{Z}, +)$
- $((\mathbb{Z}/7\mathbb{Z})^*, \cdot)$

Order of Elements and Groups

Definition (Order of an Element)

For an element $g \in G$, we define the order of the element g as $ord(g) := \min\{n \in \mathbb{N}^+ | g^n = e\}$ if it exists, ∞ otherwise.

Order of Elements and Groups

Definition (Order of an Element)

For an element $g \in G$, we define the order of the element g as $ord(g) := \min\{n \in \mathbb{N}^+ | g^n = e\}$ if it exists, ∞ otherwise.

Example

- Let $G = ((\mathbb{Z}/7\mathbb{Z})^*, \cdot)$. What is $ord(2)$?
- Let $G = ((\mathbb{Z}/7\mathbb{Z}, +))$. What is $ord(2)$?

Definition (Order of a Group)

For a group G, we call $\#G := ord(G)$ the order of G.

Order of Elements and Groups

Definition (Order of an Element)

For an element $g \in G$, we define the order of the element g as $ord(g) := \min\{n \in \mathbb{N}^+ | g^n = e\}$ if it exists, ∞ otherwise.

Example

- Let $G = ((\mathbb{Z}/7\mathbb{Z})^*, \cdot)$. What is $ord(2)$?
- Let $G = ((\mathbb{Z}/7\mathbb{Z}, +))$. What is $ord(2)$?

Definition (Order of a Group)

For a group G, we call $\#G := ord(G)$ the order of G.

Hence, the order of an element g is also the order of the cyclic subgroup $< g >$ in G.

Order of Elements and Groups

Definition (Order of an Element)

For an element $g \in G$, we define the order of the element g as $ord(g) := \min\{n \in \mathbb{N}^+ | g^n = e\}$ if it exists, ∞ otherwise.

Example

- Let $G = ((\mathbb{Z}/7\mathbb{Z})^*, \cdot)$. What is $ord(2)$?
- Let $G = ((\mathbb{Z}/7\mathbb{Z}, +))$. What is $ord(2)$?

Definition (Order of a Group)

For a group G, we call $\#G := ord(G)$ the order of G.

Hence, the order of an element g is also the order of the cyclic subgroup $< g >$ in G.

Lagrange's Theorem

Definition (Lagrange's theorem)

For a subgroup U of a group G, always $ord(U)|ord(G)$ holds.

Lagrange’s Theorem

Definition (Lagrange’s theorem)

For a subgroup U of a group G, always $ord(U)|ord(G)$ holds.

Corollary to Lagrange’s Theorem

For every element $g \in G$ it holds $g^{ord(G)} = 1$.

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
    - Basic numbertheoretic structures
    - Divisibility Relation and Factor Rings
    - Some important grouptheoretic results
    - Some important ringtheoretic results
4. Cryptographic Algorithms
    - Crypto Scheme
    - Symmetric Ciphers for Encryption
    - Asymmetric Ciphers for Encryption and Authentication
    - Key Exchange
    - Hash Functions for Ensuring Integrity
    - Selected Algorithms for Authentication
5. Summary

The principle of cryptography

Cryptographic System

An encryption is a function $enc : K \times P \to C$ from the cartesian product of the set of keys K and the set of plaintexts P to the set cyphertexts C. A decryption is a function $dec : K \times C \to P$. A suitable decryption has a key k' such that

$$\forall k \in K \forall x \in P : dec(k', enc(k, x)) = x$$

Alternative notation for $enc(k, m)$ is $enc_k(m)$.

Desired Properties for Cryptographic Systems

1. $enc_k(m)$ must be efficiently computable

Desired Properties for Cryptographic Systems

1. $enc_k(m)$ must be efficiently computable
2. $dec_{k'}(m)$ must be efficiently computable

Desired Properties for Cryptographic Systems

1. $enc_k(m)$ must be efficiently computable
2. $dec_{k'}(m)$ must be efficiently computable
3. *Resilience against known ciphertext attack:* Given $c_1, ..., c_n \in C$, $dec_{k'}(c_i)$ must not be efficiently computable (without knowledge of k')

Desired Properties for Cryptographic Systems

1. $enc_k(m)$ must be efficiently computable
2. $dec_{k'}(m)$ must be efficiently computable
3. *Resilience against known ciphertext attack:* Given $c_1, ..., c_n \in C$, $dec_{k'}(c_i)$ must not be efficiently computable (without knowledge of k')
4. *Resilience against chosen ciphertext attack:* Given $(m_1, c_1), ..., (m_n, c_n)$, $dec_{k'}(c)$ with $c \neq c_i$ should not be efficiently computable

There are more in **cryptanalysis**, which we omit here.

Kerckhoff’s Principle

Additional key principle / assumption for the quality of encryption schemes.

- The security of an encryption should rely on **key confidentiality**, not on keeping the encryption process secret.
- Hence, an encryption should be secure, even if everything except the key is public knowledge

Symmetric and Asymmetric Cryptography

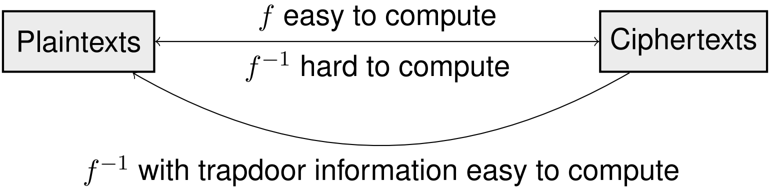
Symmetric and Asymmetric Ciphers

If k' emanates from k efficiently computable, then we speak of a symmetric cipher. Otherwise, it is an asymmetric cipher.

Symmetric and Asymmetric Cryptography

Symmetric and Asymmetric Ciphers

If k' emanates from k efficiently computable, then we speak of a symmetric cipher. Otherwise, it is an asymmetric cipher.



Plaintexts

Ciphertexts

f easy to compute

f^{-1} hard to compute

f^{-1} with trapdoor information easy to compute

What does *easy to compute* or *infeasible computing* mean?

Infeasible computation is a ”hard” instance of an NP-Hard problem of sufficient size.

What does *easy to compute* or *infeasible computing* mean?

Infeasible computation is a ”hard” instance of an NP-Hard problem of sufficient size.
”Hard” means that there is no better way than trying all possible solutions.

What does *easy to compute* or *infeasible computing* mean?

Infeasible computation is a ”hard” instance of an NP-Hard problem of sufficient size.
”Hard” means that there is no better way than trying all possible solutions.
Sufficient means, that the size is large enough that it can be considered not computable with classical computers.

What does *easy to compute* or *infeasible computing* mean?

Infeasible computation is a ”hard” instance of an NP-Hard problem of sufficient size.
”Hard” means that there is no better way than trying all possible solutions.
Sufficient means, that the size is large enough that it can be considered not computable with classical computers.

Can be defined more precisely ⇒ Theory of Computation.

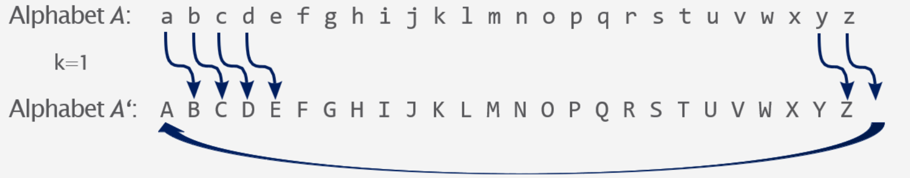
A simple example of a symmetric cipher: Caesar

The key for Caesar is in $k \in \{1, ..., 25\}$ and lets the alphabet rotate by k letters, i.e., if we identify numbers with letters via $0 \mapsto A, ..., 25 \mapsto Z$, then the Cipher of a letter α is $\alpha + k \mod 26$.
The decryption key is simply $-k$.

Example

IT Security as cleartext maps to KV UGEWTKVA for key 2

Caesar is an example of a **monoalphabetic cipher**.



Alphabet A: a b c d e f g h i j k l m n o p q r s t u v w x y z

k=1

Alphabet A‘: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Vigenere chiffe

Idea: Use a secret word, write it beneath the clear text continuously, and rotate the clear text by the amount that the secret word indicates, i.e.,

$$a \mapsto 0, b \mapsto 1, ...$$

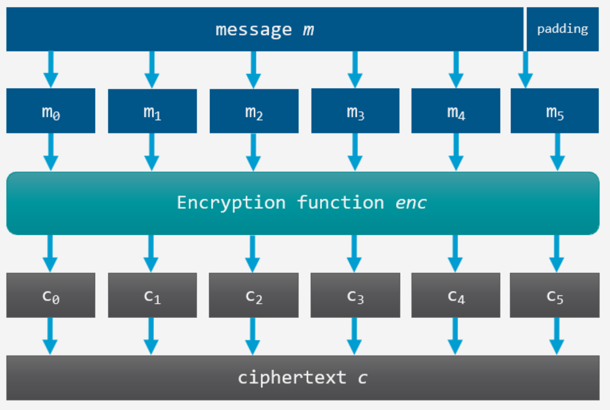
Example

clear text:	THISISATEST
secret word:	ABCABCABCAB
chiffre:	TIKSJUAUGSU

What do you think about the secrecy of the Vigenere cipher?

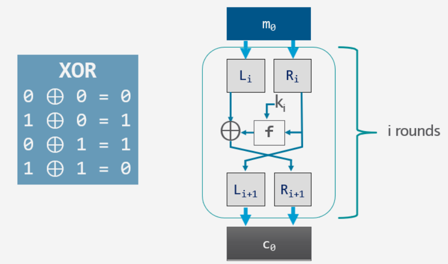
Block Cyphers

Block cyphers split the cleartext into blocks, encrypt the blocks and concatenate the encrypted blocks.



A Block Cipher Example: Data Encryption Standard (DES)

- Algorithm published by NIST, 1977
- Feistel architecture (cf. image)
- block length 64 bit or 128 bit
- broken in 1999: calculation of key in approx. 22 hours
- current standards: 3DES and Advanced Encryption Standard (AES)



XOR

$0 \oplus 0 = 0$

$1 \oplus 0 = 1$

$0 \oplus 1 = 1$

$1 \oplus 1 = 0$

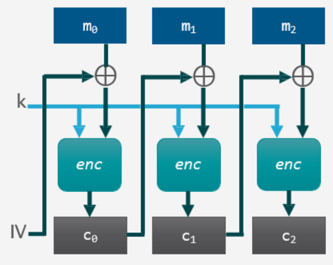
Advanced Encryption Standard (AES)

AES is an improved and still secure algorithm.

- cipher block chaining mode
- encryption depends on previous cipher block:

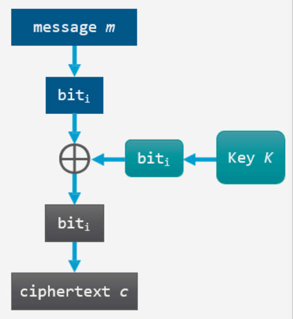
$$c_i = enc_k(m_i) \oplus c_{i-1}$$

- prevents also unnoticed block deletion and swapping
- needs initialization vector IV



Stream Ciphers

- m and K processed bitwise
- m_i and K_i cannot be deduced from c_i, knowledge about plaintext m is needed
- if K is uses only once it is calles **One Time Pad**, which is **perfectly secure**
- drawback: key management



Perfect Secrecy and the One Time Pad

- If we have N keys, the probability of choosing a particular key k is $P(K = k) = \frac{1}{N}$ (K being the random variable)
- Selecting plaintexts to be transmitted also follows a probability distribution $p(M)$, i.e., $p(M = m)$ is the probability of selecting message m for encryption and subsequent transmission.
- Same applies for cyphertexts whose probability of appearing we denote by $p(C = c)$

Perfect Secrecy and the One Time Pad

- If we have N keys, the probability of choosing a particular key k is $P(K = k) = \frac{1}{N}$ (K being the random variable)
- Selecting plaintexts to be transmitted also follows a probability distribution $p(M)$, i.e., $p(M = m)$ is the probability of selecting message m for encryption and subsequent transmission.
- Same applies for cyphertexts whose probability of appearing we denote by $p(C = c)$

Definition

Perfect Secrecy We say that a cryptosystem is **perfectly secret** if $P(M = m|C = c) = P(M = m)$

This means, knowledge of the ciphertext never changes the probability that a given plaintext occurs. On other words, no matter how much ciphertext you have, it does not convey anything about what the plaintext and key were.

Perfect Secrecy and the One Time Pad

Theorems for Perfect Secrecy

- If the key is chosen uniformly randomly from all keys of a given length, then the one-time pad is perfectly secret.
- If a cryptosystem is perfectly secret, then the number of possible keys is greater than or equal to the number of possible plaintexts.

A one time pad is perfectly secure. It requires the use of a single-use pre-shared key that is not smaller than the message being sent.

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
   - Basic numbertheoretic structures
   - Divisibility Relation and Factor Rings
   - Some important grouptheoretic results
   - Some important ringtheoretic results
4. Cryptographic Algorithms
   - Crypto Scheme
   - Symmetric Ciphers for Encryption
   - Asymmetric Ciphers for Encryption and Authentication
   - Key Exchange
   - Hash Functions for Ensuring Integrity
   - Selected Algorithms for Authentication
5. Summary

RSA

- named after the inventors Rivest, Shamit, Adleman
- published 1977
- still widely used in web browsers, email, VPNs, communications, also TLS (Transport Layer Security)
- can be used for both encryption and signing
- often also used to exchange symmetric keys securely, e.g. can be selected in TLS handshakes

RSA Encryption

Preparation - the public key

Select two primes p and q and calculate $n = p \cdot q$.

RSA Encryption

Preparation - the public key

Select two primes p and q and calculate $n = p \cdot q$.
Select e with $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$.

RSA Encryption

Preparation - the public key

Select two primes p and q and calculate $n = p \cdot q$.
Select e with $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$.
The pair (n, e) is the public key.

RSA Encryption

Preparation - the public key

Select two primes p and q and calculate $n = p \cdot q$.
Select e with $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$.
The pair (n, e) is the public key.

For example, $n = 5 \cdot 11 = 55$, $\varphi(n) = 4 \cdot 10 = 40$, $e = 7$.

RSA Encryption

Preparation - the public key

Select two primes p and q and calculate $n = p \cdot q$.
Select e with $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$.
The pair (n, e) is the public key.

For example, $n = 5 \cdot 11 = 55$, $\varphi(n) = 4 \cdot 10 = 40$, $e = 7$.

Encryption

A message m with $1 < m < n$ is encrypted as follows:

$$m^e \equiv c \mod n$$

RSA Encryption

Preparation - the public key

Select two primes p and q and calculate $n = p \cdot q$.
Select e with $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$.
The pair (n, e) is the public key.

For example, $n = 5 \cdot 11 = 55$, $\varphi(n) = 4 \cdot 10 = 40$, $e = 7$.

Encryption

A message m with $1 < m < n$ is encrypted as follows:

$$m^e \equiv c \mod n$$

Say, we encrypt $m = 8$. This yields $m^e = 8^7 \equiv 2 \mod 55$, which means 8 is encrypted by 2.

RSA Decryption

Private Key

The private key d fulfills

$$d \cdot e \equiv 1 \mod \varphi(n)$$

d can be obtained via Enhanced Euclidean Algorithm.

Enhanced Euclidean Algorithm

Purpose of the Enhanced Euclidean Algorithm (EEA)

The purpose of the EEA is to obtain a representation of the form

$$\gcd(a, b) = \xi \cdot a + \eta \cdot b$$

Enhanced Euclidean Algorithm

Purpose of the Enhanced Euclidean Algorithm (EEA)

The purpose of the EEA is to obtain a representation of the form

$$\gcd(a, b) = \xi \cdot a + \eta \cdot b$$

In our example $e = 7, \varphi(n) = 40$,

Enhanced Euclidean Algorithm

Purpose of the Enhanced Euclidean Algorithm (EEA)

The purpose of the EEA is to obtain a representation of the form

$$\gcd(a, b) = \xi \cdot a + \eta \cdot b$$

In our example $e = 7, \varphi(n) = 40$, this yields

$$40 = 5 \cdot 7 + 5, 7 = 1 \cdot 5 + 2, 5 = 2 \cdot 2 + 1$$

Enhanced Euclidean Algorithm

Purpose of the Enhanced Euclidean Algorithm (EEA)

The purpose of the EEA is to obtain a representation of the form

$$\gcd(a, b) = \xi \cdot a + \eta \cdot b$$

In our example $e = 7, \varphi(n) = 40$, this yields

$$40 = 5 \cdot 7 + 5, 7 = 1 \cdot 5 + 2, 5 = 2 \cdot 2 + 1$$

That yields in turn

$$\begin{aligned} \gcd(40, 7) = 1 = 5 - 2 \cdot 2 = 5 - 2 \cdot (7 - 5) &= -2 \cdot 7 + 3 \cdot 5 \\ = -2 \cdot 7 + 3 \cdot (40 - 5 \cdot 7) = 3 \cdot 40 &- 17 \cdot 7 \end{aligned}$$

Enhanced Euclidean Algorithm

Purpose of the Enhanced Euclidean Algorithm (EEA)

The purpose of the EEA is to obtain a representation of the form

$$\gcd(a,b) = \xi \cdot a + \eta \cdot b$$

In our example $e = 7, \varphi(n) = 40$, this yields

$$40 = 5 \cdot 7 + 5, 7 = 1 \cdot 5 + 2, 5 = 2 \cdot 2 + 1$$

That yields in turn

$$\begin{aligned}\gcd(40,7) = 1 = 5 - 2 \cdot 2 = 5 - 2 \cdot (7-5) &= -2 \cdot 7 + 3 \cdot 5 \\ = -2 \cdot 7 + 3 \cdot (40 - 5 \cdot 7) = 3 \cdot 40 &- 17 \cdot 7\end{aligned}$$

We were looking for d such that $d \cdot e \equiv 1 \mod \varphi(n)$. This translates in the example to $d = -17 \equiv 23 \mod 40$

Enhanced Euclidean Algorithm

In general, the Euclidean Algorithm to determine $\gcd(x_0, x_1)$ works via the recursive definition

$$x_{i-1} = q_i x_i + x_{i+1}$$

This is executed until we reach $x_{n+1} = 0$. Then, $x_n = \gcd(x_0, x_1)$ and by calculating backwards through the equations, we obtain the representation $\gcd(x_0, x_1) = \lambda_0 x_0 + \lambda_1 x_1$, which is the result of the EEA.

Enhanced Euclidean Algorithm

In general, the Euclidean Algorithm to determine $\gcd(x_0, x_1)$ works via the recursive definition

$$x_{i-1} = q_i x_i + x_{i+1}$$

This is executed until we reach $x_{n+1} = 0$. Then, $x_n = \gcd(x_0, x_1)$ and by calculating backwards through the equations, we obtain the representation $\gcd(x_0, x_1) = \lambda_0 x_0 + \lambda_1 x_1$, which is the result of the EEA.
Exercise: Why does that work? Implement the Euclidean Algorithm.

Exercise Enhanced Euclidean Algorithm

Apply the EEA on the numbers 57 and 13.

Decryption Example

In our example, we have obtained $d = 23$. We had $m = 8$ and obtained $c = 2$. To decrypt, we calculate

$$c^d = 2^{23} \equiv 8 \mod 55$$

Exercise RSA

Let $p = 41$, $q = 73$, $e = 7$, $m = 100$. Encrypt m via RSA with the public key (n, e) and decrypt the ciphertext.

Soundness of RSA for $gcd(m, n) = 1$

If $gcd(m, n) = 1$, then $m \in (\mathbb{Z}/n\mathbb{Z})^*$.

Soundness of RSA for $gcd(m, n) = 1$

If $gcd(m, n) = 1$, then $m \in (\mathbb{Z}/n\mathbb{Z})^*$.
We know that $\# (\mathbb{Z}/n\mathbb{Z})^* = \varphi(n)$ and $\forall x \in (\mathbb{Z}/n\mathbb{Z})^* : x^{\varphi(n)} = 1$.

Soundness of RSA for $gcd(m, n) = 1$

If $gcd(m, n) = 1$, then $m \in (\mathbb{Z}/n\mathbb{Z})^*$.
We know that $\# (\mathbb{Z}/n\mathbb{Z})^* = \varphi(n)$ and $\forall x \in (\mathbb{Z}/n\mathbb{Z})^* : x^{\varphi(n)} = 1$.
Hence, $c^d = (m^e)^d = m^{e \cdot d} = m^{k \cdot \varphi(n)+1} = m^{k \cdot \varphi(n)} \cdot m = 1 \cdot m = m$

□

Soundness of RSA for $gcd(m, n) \neq 1$

Let w.l.o.g. $\gcd(m, n) = p$, i.e., $p|m$.

Soundness of RSA for $gcd(m, n) \neq 1$

Let w.l.o.g. $\gcd(m, n) = p$, i.e., $p|m$.
Then $m = p$ or $m = 2p$ or $m = p^2$ etc. However, $\gcd(m, q) = 1$.

Soundness of RSA for $gcd(m, n) \neq 1$

Let w.l.o.g. $\gcd(m, n) = p$, i.e., $p|m$.
Then $m = p$ or $m = 2p$ or $m = p^2$ etc. However, $\gcd(m, q) = 1$.
This means $m \in (\mathbb{Z}/q\mathbb{Z})^*$, which in turn yields $m^{q-1} = 1$, and therefore $m^{k\cdot(q-1)+1} \equiv m \mod q$.

Soundness of RSA for $gcd(m, n) \neq 1$

Let w.l.o.g. $\gcd(m, n) = p$, i.e., $p|m$.
Then $m = p$ or $m = 2p$ or $m = p^2$ etc. However, $\gcd(m, q) = 1$.
This means $m \in (\mathbb{Z}/q\mathbb{Z})^*$, which in turn yields $m^{q-1} = 1$, and therefore $m^{k\cdot(q-1)+1} \equiv m \mod q$.
Then $m^{k\cdot(q-1)\cdot(p-1)+1} \equiv m \mod q$, i.e., $m^{k\cdot\varphi(n)+1} \equiv m \mod q$, and of course $m^{k\cdot\varphi(n)+1} \equiv 0 \mod q$.

Soundness of RSA for $gcd(m, n) \neq 1$

Let w.l.o.g. $\gcd(m, n) = p$, i.e., $p|m$.
Then $m = p$ or $m = 2p$ or $m = p^2$ etc. However, $\gcd(m, q) = 1$.
This means $m \in (\mathbb{Z}/q\mathbb{Z})^*$, which in turn yields $m^{q-1} = 1$, and therefore $m^{k\cdot(q-1)+1} \equiv m \mod q$.
Then $m^{k\cdot(q-1)\cdot(p-1)+1} \equiv m \mod q$, i.e., $m^{k\cdot\varphi(n)+1} \equiv m \mod q$, and of course $m^{k\cdot\varphi(n)+1} \equiv 0 \mod q$.
Hence, $m^{k\cdot\varphi(n)+1} - m \equiv 0 \mod q$ and $m^{k\cdot\varphi(n)+1} - m \equiv 0 \mod p$.

Soundness of RSA for $gcd(m, n) \neq 1$

Let w.l.o.g. $\gcd(m, n) = p$, i.e., $p|m$.
Then $m = p$ or $m = 2p$ or $m = p^2$ etc. However, $\gcd(m, q) = 1$.
This means $m \in (\mathbb{Z}/q\mathbb{Z})^*$, which in turn yields $m^{q-1} = 1$, and therefore $m^{k\cdot(q-1)+1} \equiv m \mod q$.
Then $m^{k\cdot(q-1)\cdot(p-1)+1} \equiv m \mod q$, i.e., $m^{k\cdot\varphi(n)+1} \equiv m \mod q$, and of course $m^{k\cdot\varphi(n)+1} \equiv 0 \mod q$.
Hence, $m^{k\cdot\varphi(n)+1} - m \equiv 0 \mod q$ and $m^{k\cdot\varphi(n)+1} - m \equiv 0 \mod p$.
Since $\gcd(p, q) = 1$, this yields $m^{k\cdot\varphi(n)+1} - m \equiv 0 \mod p \cdot q$.

□

Digital Signature with RSA

Note that RSA is used either for encryption as shown before, i.e., with

1. encryption: $enc_e(m) = c$ with $m^e \equiv c \mod n$. Note that e is the public key.

Digital Signature with RSA

Note that RSA is used either for encryption as shown before, i.e., with

1. encryption: $enc_e(m) = c$ with $m^e \equiv c \mod n$. Note that e is the public key.
2. decryption: $dec_d(c) = m$ with $c^d \equiv m \mod n$. Note that d is the private key.

Digital Signature with RSA

Note that RSA is used either for encryption as shown before, i.e., with

1. encryption: $enc_e(m) = c$ with $m^e \equiv c \mod n$. Note that e is the public key.
2. decryption: $dec_d(c) = m$ with $c^d \equiv m \mod n$. Note that d is the private key.
3. the connection between keys e and d is $d \cdot e \equiv 1 \mod \varphi(n)$

Digital Signature with RSA

Note that RSA is used either for encryption as shown before, i.e., with

1. encryption: $enc_e(m) = c$ with $m^e \equiv c \mod n$. Note that e is the public key.
2. decryption: $dec_d(c) = m$ with $c^d \equiv m \mod n$. Note that d is the private key.
3. the connection between keys e and d is $d \cdot e \equiv 1 \mod \varphi(n)$

The same approach can be used to sign a message:

1. Alice wants to sign a message m. So, she uses her private key d and signs m with $s = dec_d(m)$.

Digital Signature with RSA

Note that RSA is used either for encryption as shown before, i.e., with

1. encryption: $enc_e(m) = c$ with $m^e \equiv c \mod n$. Note that e is the public key.
2. decryption: $dec_d(c) = m$ with $c^d \equiv m \mod n$. Note that d is the private key.
3. the connection between keys e and d is $d \cdot e \equiv 1 \mod \varphi(n)$

The same approach can be used to sign a message:

1. Alice wants to sign a message m. So, she uses her private key d and signs m with $s = dec_d(m)$.
2. Everybody else can verify that Alice signed m by calculating $enc_e(s)$. Note that e is the public key.

Digital Signature with RSA

Note that RSA is used either for encryption as shown before, i.e., with

1. encryption: $enc_e(m) = c$ with $m^e \equiv c \mod n$. Note that e is the public key.
2. decryption: $dec_d(c) = m$ with $c^d \equiv m \mod n$. Note that d is the private key.
3. the connection between keys e and d is $d \cdot e \equiv 1 \mod \varphi(n)$

The same approach can be used to sign a message:

1. Alice wants to sign a message m. So, she uses her private key d and signs m with $s = dec_d(m)$.
2. Everybody else can verify that Alice signed m by calculating $enc_e(s)$. Note that e is the public key.
3. She compares the result with m. This works only because $m = enc_e(dec_d(m))$.

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
   - Basic numbertheoretic structures
   - Divisibility Relation and Factor Rings
   - Some important grouptheoretic results
   - Some important ringtheoretic results
4. Cryptographic Algorithms
   - Crypto Scheme
   - Symmetric Ciphers for Encryption
   - Asymmetric Ciphers for Encryption and Authentication
   - Key Exchange
   - Hash Functions for Ensuring Integrity
   - Selected Algorithms for Authentication
5. Summary

Challenge for Key Exchange

(Symmetric) keys need to be exchanged over a potentially unsecure channel. There are two main options (in our scope - there are more in general):

1. Diffie-Hellman Key Exchange
2. Exchange of symmetric key via asymmetric cryptography, e.g. RSA

Diffie Hellman Key Exchange

One option is the **Diffie Hellman Key Exchange.** Let $(G, \circ)$ be a cyclic group with a primitive root g. An example would be $\mathbb{F}_p^*$ for a prime p.

Diffie Hellman Key Exchange

One option is the **Diffie Hellman Key Exchange.** Let $(G, \circ)$ be a cyclic group with a primitive root g. An example would be $\mathbb{F}_p^*$ for a prime p. Alice and Bob want to exchange keys. Both the cyclic group and the primitive root are publicly known (or agreed upon during a handshake).

Diffie Hellman Key Exchange

One option is the **Diffie Hellman Key Exchange.** Let $(G, \circ)$ be a cyclic group with a primitive root g. An example would be $\mathbb{F}_p^*$ for a prime p. Alice and Bob want to exchange keys. Both the cyclic group and the primitive root are publicly known (or agreed upon during a handshake).

1. Alice selects number α and sends g^α in G to Bob.

Diffie Hellman Key Exchange

One option is the **Diffie Hellman Key Exchange**. Let $(G, \circ)$ be a cyclic group with a primitive root g. An example would be $\mathbb{F}_p^*$ for a prime p. Alice and Bob want to exchange keys. Both the cyclic group and the primitive root are publicly known (or agreed upon during a handshake).

1. Alice selects number α and sends g^α in G to Bob.
2. Bob selects number β and sends g^β to Alice.

Diffie Hellman Key Exchange

One option is the **Diffie Hellman Key Exchange.** Let $(G, \circ)$ be a cyclic group with a primitive root g. An example would be $\mathbb{F}_p^*$ for a prime p. Alice and Bob want to exchange keys. Both the cyclic group and the primitive root are publicly known (or agreed upon during a handshake).

1. Alice selects number α and sends g^α in G to Bob.
2. Bob selects number β and sends g^β to Alice.
3. Alice then takes her number α and calculates with the received g^β the element $\left(g^\beta\right)^\alpha = g^{\alpha\beta} \in G$.

Diffie Hellman Key Exchange

One option is the **Diffie Hellman Key Exchange.** Let $(G, \circ)$ be a cyclic group with a primitive root g. An example would be $\mathbb{F}_p^*$ for a prime p. Alice and Bob want to exchange keys. Both the cyclic group and the primitive root are publicly known (or agreed upon during a handshake).

1. Alice selects number α and sends g^α in G to Bob.
2. Bob selects number β and sends g^β to Alice.
3. Alice then takes her number α and calculates with the received g^β the element $\left(g^\beta\right)^\alpha = g^{\alpha\beta} \in G$.
4. Bob does the same with his number β and the received g^α yielding $g^{\alpha\beta}$ as well.
5. Hence, both have agreed upon the key $g^{\alpha\beta}$

Soundness of Diffie Hellmann Key Exchange

- Soundness is clear
- Security depends on the difficulty to effectively calculate α or β respectively from g^α or g^β
- This is known as **Discrete Logarithm Problem**
- Note that over the insecure channel only g^α and g^β are exchanged

Discrete Logarithm Problem

Definition (Formal Definition of the DLP)

Let p be a prime number and g a primitive root modulo p. Then, the function

$$\phi : (\mathbb{Z}/(p-1)\mathbb{Z}, +) \to (\mathbb{Z}/p\mathbb{Z})^*, n \mapsto g^n$$

is an isomorphism of the two groups.

Discrete Logarithm Problem

Definition (Formal Definition of the DLP)

Let p be a prime number and g a primitive root modulo p. Then, the function

$$\phi : (\mathbb{Z}/(p-1)\mathbb{Z}, +) \rightarrow (\mathbb{Z}/p\mathbb{Z})^*, n \mapsto g^n$$

is an isomorphism of the two groups.The inverse function of ϕ,

$$ind_g : (\mathbb{Z}/p\mathbb{Z})^* \rightarrow (\mathbb{Z}/(p-1)\mathbb{Z}, +)$$

is an isomorphism as well and is called **index** or **discrete logarithm** to the base g.

How hard is the Discrete Logarithm Problem?

- For some groups it is very easy:
  - $\mathbb{Z}/n\mathbb{Z}$ under addition (Euclidean algorithm)
  - $\mathbb{R}^*$ or $\mathbb{C}^*$ under multiplication

How hard is the Discrete Logarithm Problem?

- For some groups it is very easy:
  - $\mathbb{Z}/n\mathbb{Z}$ under addition (Euclidean algorithm)
  - $\mathbb{R}^*$ or $\mathbb{C}^*$ under multiplication
- For some it is difficult, e.g.
  - $\mathbb{F}_p^*$ (our case)
  - the group over an elliptic curve

How hard is the Discrete Logarithm Problem?

- For some groups it is very easy:
  - $\mathbb{Z}/n\mathbb{Z}$ under addition (Euclidean algorithm)
  - $\mathbb{R}^*$ or $\mathbb{C}^*$ under multiplication
- For some it is difficult, e.g.
  - $\mathbb{F}_p^*$ (our case)
  - the group over an elliptic curve
- The best known algorithm to solve DLP in $\mathbb{F}_p^*$ takes time $\mathcal{O}\left(e^{c\sqrt[3]{(\log p)(\log\log p)^2}}\right)$
- This is called *subexponential*.

RSA for key exchange

- RSA can also be used for key exchange.
- RSA is used to submit the symmetric keys via asymmetric encryption.
- With the symmetric keys, the following communication is done via symmetric encryption.

RSA for key exchange

- RSA can also be used for key exchange.
- RSA is used to submit the symmetric keys via asymmetric encryption.
- With the symmetric keys, the following communication is done via symmetric encryption.
- Reason: Symmetric ciphers, e.g. AES are more efficient.

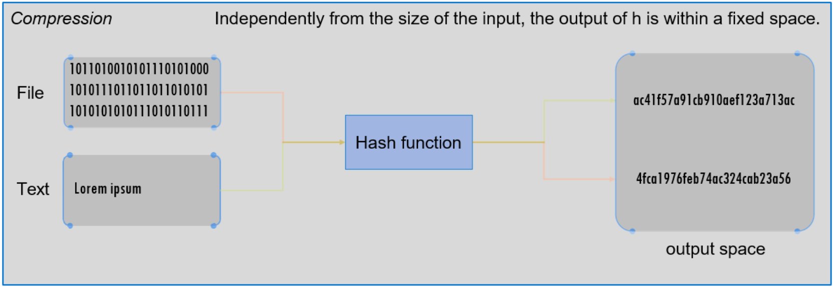
Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
   - Basic numbertheoretic structures
   - Divisibility Relation and Factor Rings
   - Some important grouptheoretic results
   - Some important ringtheoretic results
4. Cryptographic Algorithms
   - Crypto Scheme
   - Symmetric Ciphers for Encryption
   - Asymmetric Ciphers for Encryption and Authentication
   - Key Exchange
   - Hash Functions for Ensuring Integrity
   - Selected Algorithms for Authentication
5. Summary

Hash Functions

A **hash function** is a function h that converts variable-sized text into a small datum, usually a fixed-size integer.

$$h : \{0,1\}^* \to \{0,1\}^n$$


Compression
Independently from the size of the input, the output of h is within a fixed space.
File
1011010010101110101000
1010111011011011010101
1010101010111010110111
Text
Lorem ipsum
Hash function
ac41f57a91cb910aef123a713ac
4fca1976feb74ac324cab23a56
output space

Moreover, the function h must be easy to compute, i.e., given x, $h(x)$ is efficiently computable.

Cryptographic Hash Functions

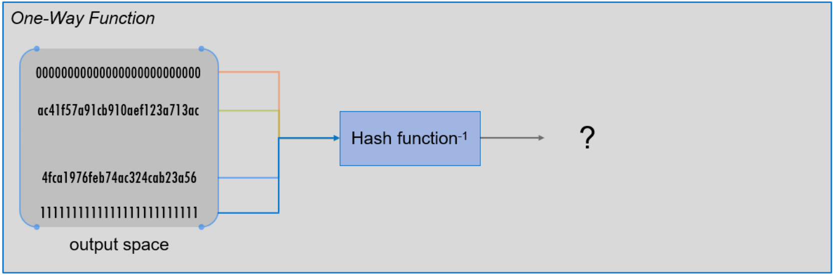
A **cryptographic hash function** has three additional requirements:

1. first preimage resistance or one way property, i.e., constructing a text for a given hash is computationally infeasible
2. second preimage resistance or weak collistion resistance, i.e., constructing another separate text from a given one that maps to the same hash is computationally infeasible
3. (strong) collision resistance, i.e., low likelihood to find any two distinct values mapping to the same hash

The primary purpose of a hash function is to protect **integrity**.

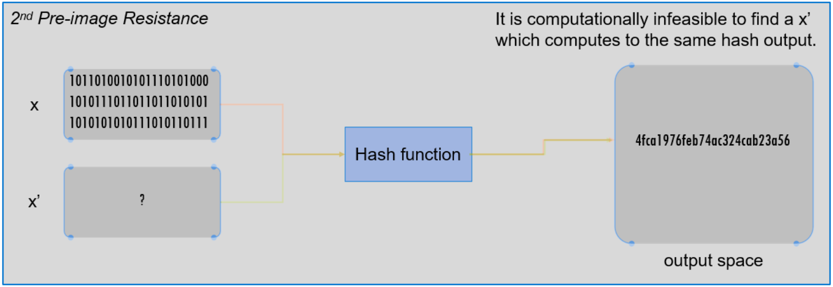
First Preimage Resistance

A function h is **preimage resistant** if, given x, it is hard to find any m such that $x = h(m)$.
h is then also called **one-way function**.



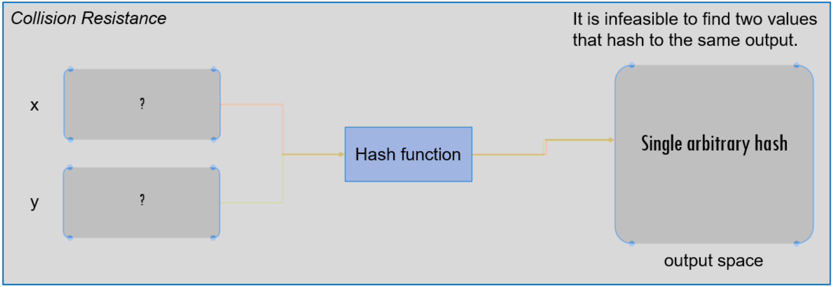
Second Preimage Resistance

A function h is **second preimage resistant** if, given an input m_1, it is hard to find $m_2 \neq m_1$ such that $f(m_1) = f(m_2)$.
This is also sometimes called **weak collision resistance.**



Collision Resistance

A function h is (strong) collision resistant if it is hard to find two messages m_1 and m_2 such that $f(m_1) = f(m_2)$.



What is the Purpose of Hash Functions?

Hash functions are used for **integrity**.

What is the Purpose of Hash Functions?

Hash functions are used for **integrity**.
A cryptographic hash function *binds* data together in a way that makes any alterations apparent. The data is made *tamper-resistant*.

What is the Purpose of Hash Functions?

Hash functions are used for **integrity**.

A cryptographic hash function *binds* data together in a way that makes any alterations apparent. The data is made *tamper-resistant*.

Process of applying a hash function:

1. Calculate hash for data and submit both the data and the hash.
2. The receiver calculates the hash of the data and compares with the received hash.

What is the Purpose of Hash Functions?

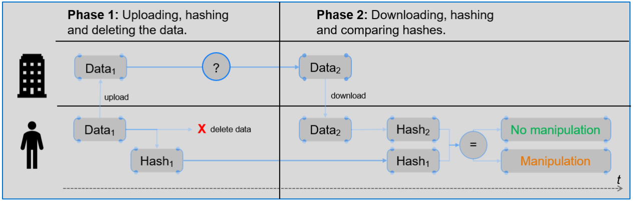
Hash functions are used for **integrity**.

A cryptographic hash function *binds* data together in a way that makes any alterations apparent. The data is made *tamper-resistant*.

Process of applying a hash function:

1. Calculate hash for data and submit both the data and the hash.
2. The receiver calculates the hash of the data and compares with the received hash.

This cannot only be used for sending/receiving data but also e.g. password hashes are stored in `/etc/shadow`



Salt for Hash Functions

Instead of storing the hash of a message $h(msg)$, a **salt**, i.e., a randomly chosen string $salt$ is chosen and $h(salt + msg)$, the concatenation of both is hashed. All values, msg, $salt$, and

$h(salt + msg)$ are stored for verification.

Salt for Hash Functions

Instead of storing the hash of a message $h(msg)$, a **salt**, i.e., a randomly chosen string $salt$ is chosen and $h(salt + msg)$, the concatenation of both is hashed. All values, msg, $salt$, and

$h(salt + msg)$ are stored for verification. Benefits: It prevents password

dictionary attacks or precalculation attacks. The attacker would need to calculate dictionaries for each possible salt upfront.

Common Hash Functions

There are many different hash algorithms:

- Message Digest MD4/MD5 *considered broken*
- Secure Hash Algorithm SHA-1 *considered broken*
- Secure Hash Algorithm SHA-2 / SHA-3 *considered safe*

The SHA-family describes a group of standardized hash functions by the National Institute for Standards and Technology (NIST). The SHA-1 & SHA-2 algorithms were developed by NIST and NSA.

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
    - Basic numbertheoretic structures
    - Divisibility Relation and Factor Rings
    - Some important grouptheoretic results
    - Some important ringtheoretic results
4. Cryptographic Algorithms
    - Crypto Scheme
    - Symmetric Ciphers for Encryption
    - Asymmetric Ciphers for Encryption and Authentication
    - Key Exchange
    - Hash Functions for Ensuring Integrity
    - Selected Algorithms for Authentication
5. Summary

Introduction to Authentication

Definition

Entity authentication is the process whereby one party is assured of the identity of a second party involved in a protocol.

Introduction to Authentication

Definition

Entity authentication is the process whereby one party is assured of the identity of a second party involved in a protocol.

Key properties for authentication protocols:

- If parties A and B are honest, authentication must be possible

Introduction to Authentication

Definition

Entity authentication is the process whereby one party is assured of the identity of a second party involved in a protocol.

Key properties for authentication protocols:

- If parties A and B are honest, authentication must be possible
- (Transferability) B cannot reuse an authentication exchange with A to successfully *impersonate* A to a third party C

Introduction to Authentication

Definition

Entity authentication is the process whereby one party is assured of the identity of a second party involved in a protocol.

Key properties for authentication protocols:

- If parties A and B are honest, authentication must be possible
- (Transferability) B cannot reuse an authentication exchange with A to successfully *impersonate* A to a third party C
- (Impersonation) The probability is negligible that any party C distinct from A carrying out the protocol and playing the role of A can cause B to accept $A's$ identity.

Challenge-Response Authentication

Idea:

- proof of identity by demonstrating the knowledge of a secret known to be associated with the entity
- without revealing the secret itself to the verifier

Challenge-Response Authentication

Idea:

- proof of identity by demonstrating the knowledge of a secret known to be associated with the entity
- without revealing the secret itself to the verifier

Solution: Providing a response to a time-variant challenge.
Examples:

- nonces - see next slide
- timestamps
- random numbers with timeout

Nonces for Hash Functions

Nonces (*number once*) are like salts, i.e., $h(n + msg)$ is submitted but incremental.

Nonces for Hash Functions

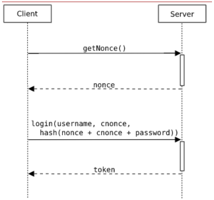
Nonces (*number once*) are like salts, i.e., $h(n + msg)$ is submitted but incremental.
This prevents **replay attacks**, i.e., attacks where the a recorded hash (by an attacker) may be used to guess the correct password. Since it is incremental, the sequence number prevents a replay.

Nonces for Hash Functions

Nonces (*number once*) are like salts, i.e., $h(n + msg)$ is submitted but incremental.
This prevents **replay attacks**, i.e., attacks where the a recorded hash (by an attacker) may be used to guess the correct password. Since it is incremental, the sequence number prevents a replay.
Example: To calculate the MD5 digest of the password in HTTP digest access authentication, every time a 401 authentication challenge response code is issued with a new nonce.

Nonces for Hash Functions

Nonces (*number once*) are like salts, i.e., $h(n + msg)$ is submitted but incremental.
This prevents **replay attacks**, i.e., attacks where the a recorded hash (by an attacker) may be used to guess the correct password. Since it is incremental, the sequence number prevents a replay.
Example: To calculate the MD5 digest of the password in HTTP digest access authentication, every time a 401 authentication challenge response code is issued with a new nonce.



Principles for Authentication

Usually *symmetric* or *asymmetric* or combinations of both together with one of the challenge response mechanisms is used

Example:

$$A \leftarrow B : h(r), B, P_A(r, B) \tag{1}$$

$$A \rightarrow B : r \tag{2}$$

1. B chooses a random r
2. B computes $x = h(r)$ (h being a hash function, together with a salt or nonce).
3. B computes $e = P_A(r, B)$, P_A being a public key cryptosystem
4. B sends the information from equation (1) to A
5. A decrypts e to recover r', computes $x' = h(r')$ and checks if $x = x'$.
6. If so, it sends r to B to confirm its acceptance.

Zero Knowledge Protocols

Goal: Achieve **authentication** with asymmetric techniques but **not** relying on digital signatures or public key encryption, and also avoid sequence numbers and timestamps.

Zero Knowledge Protocols

Goal: Achieve **authentication** with asymmetric techniques but **not** relying on digital signatures or public key encryption, and also avoid sequence numbers and timestamps.

Why is that of interest?

- Simple password protocols reveal the password of the *claimant* A and can be used by the *verifier* B to impersonate A
- Challenge-response protocols improve on this: A responds to B's challenge to demonstrate knowledge of A's secre in a time-variant manner; the information is not directly usable by B.
- However, some partial information may be revealed about A's secret or B may be able to strategically select challenges to obtain information about the secret

$\Rightarrow$ Zero Knowledge Protocols

Fiat-Shamir, an example of Zero-Knowledge Protocols

Summary: A proves knowledge of s to B in t executions

Fiat-Shamir, an example of Zero-Knowledge Protocols

Summary: A proves knowledge of s to B in t executions

Initial setup:

1. A trusted center T selects and publishes an RSA-like modulus $n = p \cdot q$, primes p and q are kept secret.

Fiat-Shamir, an example of Zero-Knowledge Protocols

Summary: A proves knowledge of s to B in t executions

Initial setup:

1. A trusted center T selects and publishes an RSA-like modulus $n = p \cdot q$, primes p and q are kept secret.
2. Claimant A selects a secret s with $\gcd(s, n) = 1$, $1 \leq s < n$, computes $v = s^2 \mod n$, and registers v as public key at T.

Fiat-Shamir Protocol

Protocol Actions and Messages: Each of t rounds has three steps as follows:

1. (commitment): A chooses random r, $1 \leq r < n$, sends $A \to B : x = r^2 \mod n$

Fiat-Shamir Protocol

Protocol Actions and Messages: Each of t rounds has three steps as follows:

1. (commitment): A chooses random r, $1 \leq r < n$, sends $A \rightarrow B : x = r^2 \mod n$
2. (challenge): B selects $e = 0$ or $e = 1$ and sends $A \leftarrow B : e \in \{0, 1\}$

Fiat-Shamir Protocol

Protocol Actions and Messages: Each of t rounds has three steps as follows:

1. (commitment): A chooses random r, $1 \leq r < n$, sends $A \rightarrow B : x = r^2 \mod n$
2. (challenge): B selects $e = 0$ or $e = 1$ and sends $A \leftarrow B : e \in \{0, 1\}$
3. (response): A computes the response, sends either $y = r$ if $e = 0$ or $y = rs$ if $e = 1$, all in $\mathbb{Z}/n\mathbb{Z}$, and sends $A \rightarrow B : y = r \cdot s^e \mod n$

Fiat-Shamir Protocol

Protocol Actions and Messages: Each of t rounds has three steps as follows:

1. (commitment): A chooses random r, $1 \leq r < n$, sends $A \rightarrow B : x = r^2 \mod n$
2. (challenge): B selects $e = 0$ or $e = 1$ and sends $A \leftarrow B : e \in \{0, 1\}$
3. (response): A computes the response, sends either $y = r$ if $e = 0$ or $y = rs$ if $e = 1$, all in $\mathbb{Z}/n\mathbb{Z}$, and sends $A \rightarrow B : y = r \cdot s^e \mod n$
4. B rejects if $y = 0$ and accepts if $y^2 \equiv x \cdot v^e \mod n$

B accepts if all t rounds succeed.

Soundness of Fiat-Shamir

- B checks $y^2 \equiv x \cdot v^e \mod n$.

Soundness of Fiat-Shamir

- B checks $y^2 \equiv x \cdot v^e \mod n$.
- It is $y^2 = (r \cdot s^e)^2 = r^2 \cdot \left(s^2\right)^e = x \cdot \left(s^2\right)^e$ (since $x \equiv r^2 \mod n$).

Soundness of Fiat-Shamir

- B checks $y^2 \equiv x \cdot v^e \mod n$.
- It is $y^2 = (r \cdot s^e)^2 = r^2 \cdot \left(s^2\right)^e = x \cdot \left(s^2\right)^e$ (since $x \equiv r^2 \mod n$).
- And $x \cdot \left(s^2\right)^e = x \cdot v^e \mod n$ (since v, the public key is $v = s^2 \mod n$). So, we have $y^2 = x \cdot v^e \mod n$.

Soundness of Fiat-Shamir

- B checks $y^2 \equiv x \cdot v^e \mod n$.
- It is $y^2 = (r \cdot s^e)^2 = r^2 \cdot \left(s^2\right)^e = x \cdot \left(s^2\right)^e$ (since $x \equiv r^2 \mod n$).
- And $x \cdot \left(s^2\right)^e = x \cdot v^e \mod n$ (since v, the public key is $v = s^2 \mod n$). So, we have $y^2 = x \cdot v^e \mod n$.

So, the equations make sense. But we need a discussion why this works and why does it not reveal the secret? We do this informally.

Informal Discussion of Soundness of Fiat-Shamir

- The challenge requires A to be capable of answering two questions:
  1. One where A needs to demonstrate knowledge of the secret s (the case where the challenge is $e = 1$)
  2. One easy question (for honest provers) to prevent cheating ($e = 0$, no secret necessary)

Informal Discussion of Soundness of Fiat-Shamir

- The challenge requires A to be capable of answering two questions:
  1. One where A needs to demonstrate knowledge of the secret s (the case where the challenge is $e = 1$)
  2. One easy question (for honest provers) to prevent cheating ($e = 0$, no secret necessary)
- Adversary impersonating A might try to cheat by selecting any r and setting $x = \frac{r^2}{v}$ (v is A's public key) and answering the challenge $e = 1$ with a correct $y = r$.
  - Note that the adversary sends in the first step $x = \frac{r^2}{v}$ to B and later $y = r$.

Informal Discussion of Soundness of Fiat-Shamir

- The challenge requires A to be capable of answering two questions:
  1. One where A needs to demonstrate knowledge of the secret s (the case where the challenge is $e = 1$)
  2. One easy question (for honest provers) to prevent cheating ($e = 0$, no secret necessary)
- Adversary impersonating A might try to cheat by selecting any r and setting $x = \frac{r^2}{v}$ (v is A's public key) and answering the challenge $e = 1$ with a correct $y = r$.
  - Note that the adversary sends in the first step $x = \frac{r^2}{v}$ to B and later $y = r$.
  - Then B tries to verify $y^2 \equiv x \cdot v^e = x \cdot v$ (we are in the case $e = 1$), which is $\frac{r^2}{v} \cdot v = r^2$.

Informal Discussion of Soundness of Fiat-Shamir

- The challenge requires A to be capable of answering two questions:
  1. One where A needs to demonstrate knowledge of the secret s (the case where the challenge is $e = 1$)
  2. One easy question (for honest provers) to prevent cheating ($e = 0$, no secret necessary)
- Adversary impersonating A might try to cheat by selecting any r and setting $x = \frac{r^2}{v}$ (v is A's public key) and answering the challenge $e = 1$ with a correct $y = r$.
  - Note that the adversary sends in the first step $x = \frac{r^2}{v}$ to B and later $y = r$.
  - Then B tries to verify $y^2 \equiv x \cdot v^e = x \cdot v$ (we are in the case $e = 1$), which is $\frac{r^2}{v} \cdot v = r^2$.
  - So, B verifies $r^2 = r^2 \mod n$ and therefore does not recognise that the adversary did not even know the secret s in this case.

Informal Discussion of Soundness of Fiat-Shamir

- The challenge requires A to be capable of answering two questions:
  1. One where A needs to demonstrate knowledge of the secret s (the case where the challenge is $e = 1$)
  2. One easy question (for honest provers) to prevent cheating ($e = 0$, no secret necessary)
- Adversary impersonating A might try to cheat by selecting any r and setting $x = \frac{r^2}{v}$ (v is A's public key) and answering the challenge $e = 1$ with a correct $y = r$.
  - Note that the adversary sends in the first step $x = \frac{r^2}{v}$ to B and later $y = r$.
  - Then B tries to verify $y^2 \equiv x \cdot v^e = x \cdot v$ (we are in the case $e = 1$), which is $\frac{r^2}{v} \cdot v = r^2$.
  - So, B verifies $r^2 = r^2 \mod n$ and therefore does not recognise that the adversary did not even know the secret s in this case.
  - So, in this case, the adversary can get away by selecting $x = \frac{r^2}{v}$. However, this works only for one round.

Informal Discussion of Soundness of Fiat-Shamir

- The challenge requires A to be capable of answering two questions:
  1. One where A needs to demonstrate knowledge of the secret s (the case where the challenge is $e = 1$)
  2. One easy question (for honest provers) to prevent cheating ($e = 0$, no secret necessary)
- Adversary impersonating A might try to cheat by selecting any r and setting $x = \frac{r^2}{v}$ (v is A's public key) and answering the challenge $e = 1$ with a correct $y = r$.
  - Note that the adversary sends in the first step $x = \frac{r^2}{v}$ to B and later $y = r$.
  - Then B tries to verify $y^2 \equiv x \cdot v^e = x \cdot v$ (we are in the case $e = 1$), which is $\frac{r^2}{v} \cdot v = r^2$.
  - So, B verifies $r^2 = r^2 \mod n$ and therefore does not recognise that the adversary did not even know the secret s in this case.
  - So, in this case, the adversary can get away by selecting $x = \frac{r^2}{v}$. However, this works only for one round.

Informal Discussion of Soundness of Fiat-Shamir

So, we had that the adversary can get away for $e = 1$ if they choose $x = \frac{r^2}{v}$ and submit $y = r$ as answer.

Informal Discussion of Soundness of Fiat-Shamir

So, we had that the adversary can get away for $e = 1$ if they choose $x = \frac{r^2}{v}$ and submit $y = r$ as answer.

However, if $e = 0$ this does not work since the adversary would need to know the square root for $x = \frac{r^2}{v}$, which is hard to compute. Note that the three steps follow one after another: Firstly, the adversary selects x, then the challenge is submitted ($e = 0$ or $e = 1$), and then the adversary needs to respond. If $e = 1$, they succeed, if $e = 0$ they don't.

Informal Discussion of Soundness of Fiat-Shamir

So, we had that the adversary can get away for $e = 1$ if they choose $x = \frac{r^2}{v}$ and submit $y = r$ as answer.

However, if $e = 0$ this does not work since the adversary would need to know the square root for $x = \frac{r^2}{v}$, which is hard to compute. Note that the three steps follow one after another: Firstly, the adversary selects x, then the challenge is submitted ($e = 0$ or $e = 1$), and then the adversary needs to respond. If $e = 1$, they succeed, if $e = 0$ they don't.

So, in each round, the adversary has a 0.5 chance of escaping detection. Hence, for t rounds the chance is at 2^{-t}

Why can't we just do two rounds, one with $e = 1$, one with $e = 0$?

Informal Discussion of Soundness of Fiat-Shamir

So, we had that the adversary can get away for $e = 1$ if they choose $x = \frac{r^2}{v}$ and submit $y = r$ as answer.

However, if $e = 0$ this does not work since the adversary would need to know the square root for $x = \frac{r^2}{v}$, which is hard to compute. Note that the three steps follow one after another: Firstly, the adversary selects x, then the challenge is submitted ($e = 0$ or $e = 1$), and then the adversary needs to respond. If $e = 1$, they succeed, if $e = 0$ they don't.

So, in each round, the adversary has a 0.5 chance of escaping detection. Hence, for t rounds the chance is at 2^{-t}

Why can't we just do two rounds, one with $e = 1$, one with $e = 0$?
Because an adversary could then submit $x = \frac{r^2}{v}$ in the first case and $x = r^2$ in the second. e needs to be selected by chance.

Secret Protection in Fiat-Shamir

Why is the secret kept secret during protocol?

- For $e = 0$, the response $y = r$ is independent from the secret s.
- For $e = 1$, the response is $y = rs \mod n$. However, since r is randomly chosen, no information about s itself is revealed.

Outline

1. Introduction
2. Recapture of mathematical basics
3. Numbertheoretic background
    - Basic numbertheoretic structures
    - Divisibility Relation and Factor Rings
    - Some important grouptheoretic results
    - Some important ringtheoretic results
4. Cryptographic Algorithms
    - Crypto Scheme
    - Symmetric Ciphers for Encryption
    - Asymmetric Ciphers for Encryption and Authentication
    - Key Exchange
    - Hash Functions for Ensuring Integrity
    - Selected Algorithms for Authentication
5. Summary

Summary

- While not everything in security is cryptography, the mathematical foundations and cryptographic algorithms are behind many other concepts in security.
- Hence, understanding in depth cryptography is key to understanding security concepts

Summary

- While not everything in security is cryptography, the mathematical foundations and cryptographic algorithms are behind many other concepts in security.
- Hence, understanding in depth cryptography is key to understanding security concepts
- We went through
    1. Symmetric ciphers

Summary

- While not everything in security is cryptography, the mathematical foundations and cryptographic algorithms are behind many other concepts in security.
- Hence, understanding in depth cryptography is key to understanding security concepts
- We went through
  1. Symmetric ciphers
  2. asymmetric ciphers as means to encrypt (confidentiality) and sign (authentication) information

Summary

- While not everything in security is cryptography, the mathematical foundations and cryptographic algorithms are behind many other concepts in security.
- Hence, understanding in depth cryptography is key to understanding security concepts
- We went through
  1. Symmetric ciphers
  2. asymmetric ciphers as means to encrypt (confidentiality) and sign (authentication) information
  3. hash functions as means to ensure integrity of information

Summary

- While not everything in security is cryptography, the mathematical foundations and cryptographic algorithms are behind many other concepts in security.
- Hence, understanding in depth cryptography is key to understanding security concepts
- We went through
  1. Symmetric ciphers
  2. asymmetric ciphers as means to encrypt (confidentiality) and sign (authentication) information
  3. hash functions as means to ensure integrity of information
  4. algoriths to exchange symmetric keys, one with the use of asymmetric ciphers, the other (Diffie Hellman) independent from that

Summary

- While not everything in security is cryptography, the mathematical foundations and cryptographic algorithms are behind many other concepts in security.
- Hence, understanding in depth cryptography is key to understanding security concepts
- We went through
  1. Symmetric ciphers
  2. asymmetric ciphers as means to encrypt (confidentiality) and sign (authentication) information
  3. hash functions as means to ensure integrity of information
  4. algoriths to exchange symmetric keys, one with the use of asymmetric ciphers, the other (Diffie Hellman) independent from that
  5. selected algorithms for authentication, mainly challenge-response and zero-knowledge

Summary

- While not everything in security is cryptography, the mathematical foundations and cryptographic algorithms are behind many other concepts in security.
- Hence, understanding in depth cryptography is key to understanding security concepts
- We went through
  1. Symmetric ciphers
  2. asymmetric ciphers as means to encrypt (confidentiality) and sign (authentication) information
  3. hash functions as means to ensure integrity of information
  4. algoriths to exchange symmetric keys, one with the use of asymmetric ciphers, the other (Diffie Hellman) independent from that
  5. selected algorithms for authentication, mainly challenge-response and zero-knowledge
- With that, we are prepared to move on with the next topic: Authentication