Modern vs. traditional cryptography

Recap: traditional crypto

- An „art” rather than science
- Ad-hoc constructions
- Easy(ish) to break

Basic principles of modern crypto

1. formal and precise definitions
2. build on precisely stated assumptions
3. strictly proven security

Principle 1: precise definitions

Precise definitions

- design (What's the purpose / goal? Rather than „ex post facto")
- usage (appropriate for goal?)
- analysis (comparison)
- intuition not enough

„Define" secure encryption

An encryption method is secure if ???

Principle 1: precise definitions

Definition of secure encryption

An encryption method is considered secure if

1. no attacker can recover the key from the ciphertext itself.
2. no attacker can recover the plaintext message from the ciphertext itself (even if only a small part of the message is missing)
3. no attacker can recover a single character of the message from the ciphertext (probabilities / order of magnitude of computation needed)
4. no attacker can learn anything imprtant about the message knowing only the ciphertext (what counts as important?)
5. no attacker can recover any function (e.g. length, letter statistics, etc.) from the ciphertext

Principle 1: precise definitions

Definition of secure encryption

An encryption method is considered secure if

1. no attacker can recover the key from the ciphertext itself.
2. no attacker can recover the plaintext message from the ciphertext itself (even if only a small part of the message is missing)
3. no attacker can recover a single character of the message from the ciphertext (probabilities / order of magnitude of computation needed)
4. no attacker can learn anything imprtant about the message knowing only the ciphertext (what counts as important?)
5. no attacker can recover any function (e.g. length, letter statistics, etc.) from the ciphertext

Principle 1: precise definitions

Definition of secure encryption

An encryption method is considered secure if

1. no attacker can recover the key from the ciphertext itself.
2. no attacker can recover the plaintext message from the ciphertext itself (even if only a small part of the message is missing)
3. no attacker can recover a single character of the message from the ciphertext (probabilities / order of magnitude of computation needed)
4. no attacker can learn anything imprtant about the message knowing only the ciphertext (what counts as important?)
5. no attacker can recover any function (e.g. length, letter statistics, etc.) from the ciphertext

Principle 1: precise definitions

Definition of secure encryption

An encryption method is considered secure if

1. no attacker can recover the key from the ciphertext itself.
2. no attacker can recover the plaintext message from the ciphertext itself (even if only a small part of the message is missing)
3. no attacker can recover a single character of the message from the ciphertext (probabilities / order of magnitude of computation needed)
4. no attacker can learn anything imprtant about the message knowing only the ciphertext (what counts as important?)
5. no attacker can recover any function (e.g. length, letter statistics, etc.) from the ciphertext

Principle 1: precise definitions

Definition of secure encryption

An encryption method is considered secure if

1. no attacker can recover the key from the ciphertext itself.
2. no attacker can recover the plaintext message from the ciphertext itself (even if only a small part of the message is missing)
3. no attacker can recover a single character of the message from the ciphertext (probabilities / order of magnitude of computation needed)
4. no attacker can learn anything imprtant about the message knowing only the ciphertext (what counts as important?)
5. no attacker can recover any function (e.g. length, letter statistics, etc.) from the ciphertext

Principle 1: precise definitions

Definition of secure encryption

An encryption method is considered secure if

1. no attacker can recover the key from the ciphertext itself.
2. no attacker can recover the plaintext message from the ciphertext itself (even if only a small part of the message is missing)
3. no attacker can recover a single character of the message from the ciphertext (probabilities / order of magnitude of computation needed)
4. no attacker can learn anything imprtant about the message knowing only the ciphertext (what counts as important?)
5. no attacker can recover any function (e.g. length, letter statistics, etc.) from the ciphertext

Principle 1: precise definitions

How to make the definition formal?

- What does „break a cipher” / „recover the message” mean?
- What does „no attacker” mean (what powers do they posess)?

Example

A cryptographic protocol meant for a specific purpose is secure if no attacker with the specified (computational) power can perform a specified form of attack.

Principle 1: precise definitions

Math vs. practice

- hardware-based attacks
- human factors

Good definition of security/secrecy should

- support the intuitive view
- be supported by examples
- be backed up by ongoing analysis over time

Principle 2: precise assumptions

Two variants

- unconditional secrecy
- computational secrecy

Why?

- validation of assumptions
- comparison of methods
- facilitate security proofs

Principle 3: formal proofs of secrecy

Why?

- established difficulty vs. naive intuition
- works $\not\Rightarrow$ unbreakable
- risks of poor cryptosystem or poor software product

Proof of secrecy by reduction

Protocol X is considered secret (by a certain definition) if assumption Y is correct.

Perfect secrecy

Informálisan

What do we need to specify a scheme?

- three algorithms: Gen, Enc, Dec
- message space $\mathcal{M}$

Parameters

- *key space*: $\mathcal{K}$ set of possible keys ($k \in \mathcal{K}$)
- *message space*: $\mathcal{M}$ set of possible messages ($m \in \mathcal{M}$)
- *ciphertext space*: $\mathcal{C}$ set of possible ciphertexts
- Usually finite (esp. keyspace - „large” but finite)

Perfectly secret scheme

Parameters

- *Probability distributions* over $\mathcal{K}, \mathcal{M}, \mathcal{C}$
- $k \in \mathcal{K} : Pr(K = k)$ denotes the probability that key k is chosen.

e.g.: $\mathcal{K}$: bit sequences of length128,
$k \in_R \mathcal{K} \Rightarrow Pr(K = k) = 1/2^{128}$

- Similarly for $\mathcal{M}, \mathcal{C}$

e.g.: $|\mathcal{M}| = 2, Pr(\text{Attack tomorrow}) = 0.7, Pr(\text{No attack}) = 0.3$

- Distribution over $\mathcal{K}$ and $\mathcal{M}$ independent and arbitrary
- Distribution over $\mathcal{C}$ determined by the other two.
- *conditional probability*: $Pr(A \mid B)$: Probability of A, provided that we know B is true.

Perfectly secret scheme

Parameters

- *Probability distributions* over $\mathcal{K}, \mathcal{M}, \mathcal{C}$
- $k \in \mathcal{K} : Pr(K = k)$ denotes the probability that key k is chosen.

e.g.: $\mathcal{K}$: bit sequences of length128,
$k \in_R \mathcal{K} \Rightarrow Pr(K = k) = 1/2^{128}$

- Similarly for $\mathcal{M}, \mathcal{C}$

e.g.: $|\mathcal{M}| = 2, Pr(\text{Attack tomorrow}) = 0.7, Pr(\text{No attack}) = 0.3$

- Distribution over $\mathcal{K}$ and $\mathcal{M}$ independent and arbitrary
- Distribution over $\mathcal{C}$ determined by the other two.
- *conditional probability*: $Pr(A \mid B)$: Probability of A, provided that we know B is true.

Perfectly secret scheme

Definition

A scheme is a triple $\Pi = (Gen, Enc, Dec)$ *where :*

- Gen *is key generation, a probabilistic algorithm that returns a key* $k \in_R \mathcal{K}$ *(maybe using an input called the security parameter)*
- Enc *is encryption, a probabilistic algorithm that returns a ciphertext* $c \in \mathcal{C}$ *on inputs* $k \in \mathcal{K}$ *and* $m \in \mathcal{M}$*, i.e.* $c := Enc_k(m)$*.*
- Dec *is decryption a deterministic algorithm that returns a plaintext upon inputs* k *and* $c \in \mathcal{C}$*: the return value is* $Dec_k(c)\ in \mathcal{M}$*.*

Secrecy definitions

Intuiton

- we know the distribution of messages
- knowing the ciphertext, no information about the message should be learnt
- attacker's computational power: infinite

Definition

A scheme over $\mathcal{M}$ is perfectly secret if for any distribution over $\mathcal{M}$ and $\forall m \in \mathcal{M}, \forall c \in \mathcal{C}$:

$$Pr(M = m) = Pr(M = m | C = c).$$

Secrecy definitions

Equivalent formulation

- We cannot distinguish the ciphertexts corresponding to two different messages.

Lemma (Perfect indistinguishability)

A scheme provides perfect secrecy if for any distribution over $\mathcal{M}$, and $\forall m_1, m_2 \in \mathcal{M}, \forall c \in \mathcal{C}$:

$$Pr(C = c|M = m_1) = Pr(C = c|M = m_2).$$

Equivalent formulation

- indistinguishability game: two players: adversary and tester

Indistinguishability experiment with eavesdropper $PrivK^{eav}_{\mathcal{A},\Pi}$

1. Adversary $\mathcal{A}$ issues messages m_0, m_1 with $|m_0| = |m_1|$
2. Tester randomly chooses jey k and bit $b \in_R \{0,1\} : c = Enc_k(m_b)$. Send c to $\mathcal{A}$.
3. $\mathcal{A}$ answers by outputting $b' \in \{0,1\}$
4. $PrivK^{eav}_{\mathcal{A},\Pi} = 1$ if $b = b'$, otherwise 0. (Wins the game if guesses correctly.)

Secrecy definitions

Indistinguishability experiment with eavesdropper $PrivK^{eav}_{\mathcal{A},\Pi}$

1. Adversary $\mathcal{A}$ issues messages m_0, m_1 with $|m_0| = |m_1|$
2. Tester randomly chooses jey k and bit $b \in_R \{0,1\} : c = Enc_k(m_b)$. Send c to $\mathcal{A}$.
3. $\mathcal{A}$ answers by outputting $b' \in \{0,1\}$
4. $PrivK^{eav}_{\mathcal{A},\Pi} = 1$ if $b = b'$, otherwise 0. (Wins the game if guesses correctly.)

Definition

A scheme Π is perfectly secret over $\mathcal{M}$ if $\forall \mathcal{A}$:

$$Pr(PrivK^{eav}_{\mathcal{A},\Pi} = 1) = \frac{1}{2}.$$

Lemma

These definitions are equivalent.

One-time pad

One-time pad (OTP)

Initialize $\mathcal{K} = \mathcal{M} = \{0,1\}^n$

Gen let $k \in_R \{0,1\}^n$ uniformly random

Enc for $k \in \{0,1\}^n$ and $m \in \{0,1\}^n$, let $c = Enc_k(m) = k \oplus m$.

Dec for k and $c \in \{0,1\}^n$ let $Dec_k(c) = c \oplus k$.

Theorem

One-time pad is perfectly secret.

Drawbacks of perfect secrecy

One-time pad properties

- $|k| = |m| \Rightarrow$ too long keys / short messages only
- "one-time" (really, never reuse!)
- these are not unique to OTP, but inherent to perfect secrecy

Theorem

Let Π be a perfectly secret shceme over $\mathcal{M}$ and let $\mathcal{K}$ be the key space determined by Gen. Then $|\mathcal{K}| \geq |\mathcal{M}|$.