Computer Security — Part 3: Information Security and Cryptography Sections 3 and 5 (week 3)

Dusko Pavlovic

Oxford
Michaelmas Term 2008

Outline

Information, channel security, noninterference

Encryption and decryption

Cryptanalysis and notions of secrecy

Cyphers and modes of operation

Key establishment

What did we learn?

Outline

Outline

Outline

Cryptanalytic attacks

Symmetric key attacks

When $K_E = K_D = K$, the attacks are

- cyphertext only (COA):

$$E(K, m_1), \ldots, E(K, m_\ell) \quad \vdash \quad K$$

- known plaintext (KPA), chosen plaintext (CPA):

$$m_1, \ldots, m_\ell, E(K, m_1), \ldots, E(K, m_\ell) \quad \vdash \quad K$$

- chosen cyphertext (CCA):

$$c_1, \ldots, c_\ell, D(K, c_1), \ldots, D(K, c_\ell) \quad \vdash \quad K$$

Cryptanalytic attacks

Asymmetric key attacks

When K_E is publicly known

- cyphertext only (COA):

$$K_E, E(K_E, m_1), \ldots, E(K_E, m_\ell) \quad \vdash \quad K_D$$

- known plaintext (KPA), chosen plaintext (CPA):

$$K_E, m_1, \ldots, m_\ell, E(K_E, m_1), \ldots, E(K_E, m_\ell) \quad \vdash \quad K_D$$

- chosen cyphertext (CCA):

$$K_E, c_1, \ldots, c_\ell, D(K_D, c_1), \ldots, D(K_D, c_\ell) \quad \vdash \quad K_D$$

- adaptive chosen cyphertext (CCA2): . . . (later!)

COA on monoalphabetic shift cypher

- $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}$
- $\mathcal{K} = \mathbb{Z}_{26}$
- $\mathsf{K}_E = \mathsf{K}_D = k$
- $\mathsf{E}(k, m) = m + k \mod 26$
- $\mathsf{D}(k, c) = c - k \mod 26$

COA on monoalphabetic shift cypher

- $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}$
- $\mathcal{K} = \mathbb{Z}_{26}$
- $K_E = K_D = k$
- $E(k,m) = m + k \mod 26$
- $D(k,c) = c - k \mod 26$

Idea

Since there are just $\#\mathcal{K} = 26$ possible keys, simply try one after the other.

COA on monoalphabetic shift cypher

CY:	N	Y	N	X	A	J	W	D	H	T	Q	I
$\vec{c}$	13	24	13	23	0	9	22	3	7	19	16	8
k_1	1	1	1	1	1	1	1	1	1	1	1	1
$\vec{m}_1$	12	23	12	22	25	8	21	2	6	18	15	7
tx_1:	m	x	m	w	z	i	v	c	g	s	p	h

COA on monoalphabetic shift cypher

CY:	N	Y	N	X	A	J	W	D	H	T	Q	I
$\vec{c}$	13	24	13	23	0	9	22	3	7	19	16	8
k_2	2	2	2	2	2	2	2	2	2	2	2	2
$\vec{m}_2$	11	22	11	21	24	7	20	1	5	17	14	6
tx_2:	l	w	l	v	y	h	u	b	f	r	o	g

COA on monoalphabetic shift cypher

CY:	N	Y	N	X	A	J	W	D	H	T	Q	I
$\vec{c}$	13	24	13	23	0	9	22	3	7	19	16	8
k_5	5	5	5	5	5	5	5	5	5	5	5	5
$\vec{m}_5$	8	19	8	18	21	4	17	24	2	14	11	3
tx_5:	i	t	i	s	v	e	r	y	c	o	l	d

COA on substitution cypher

- $\mathcal{M} = \mathcal{C} = \Sigma = \{a, b, c, \ldots, z\}$,
- $\mathcal{K} = S(\Sigma) =$ the permutations of Σ
- $K_E = K_D = \sigma$
- $E(\sigma, m) = \sigma(m)$
- $D(\sigma, c) = \sigma^{-1}(c)$

COA on substitution cypher

- $\mathcal{M} = \mathcal{C} = \Sigma = \{a, b, c, \ldots, z\}$,
- $\mathcal{K} = S(\Sigma) =$ the permutations of Σ
- $\mathsf{K_E} = \mathsf{K_D} = \sigma$
- $\mathsf{E}(\sigma, m) = \sigma(m)$
- $\mathsf{D}(\sigma, c) = \sigma^{-1}(c)$

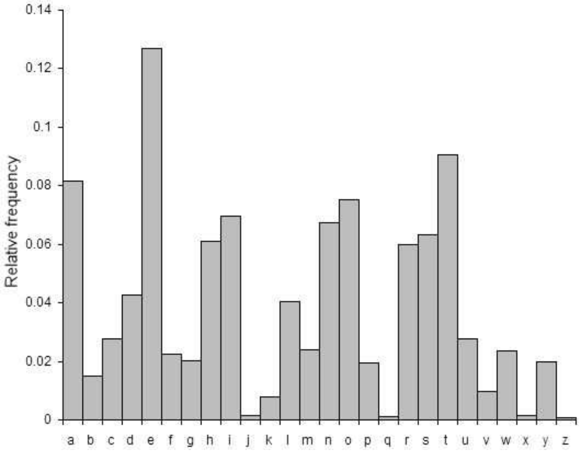
Fact

Since $\#\mathcal{K} = 26! \approx 4 \cdot 10^{26}$, enumerating the keys and *searching for a well-formed plaintext* will not help.

COA on substitution cypher

Idea

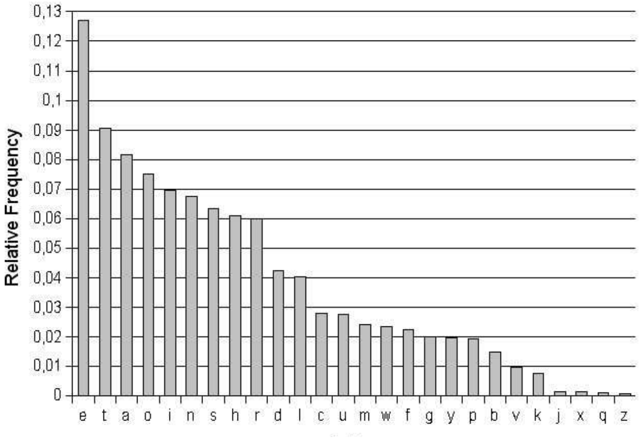
Align the letter frequencies of plaintext (e.g. English). . .



COA on substitution cypher

Idea

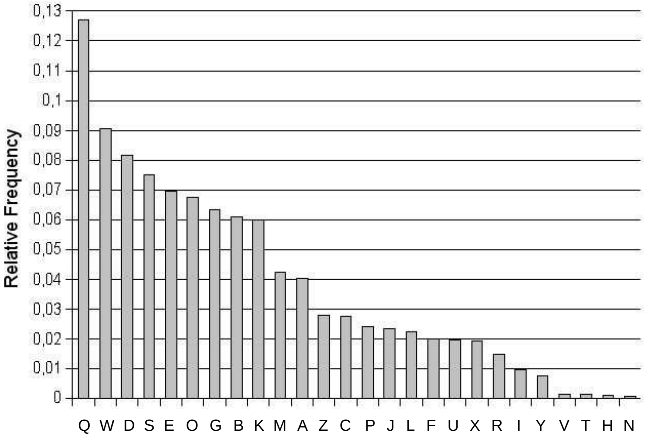
Align the letter frequencies of plaintext (e.g. English). . .



COA on substitution cypher

Idea

. . . with the letter frequencies of the cyphertext



COA on substitution cypher

Summary

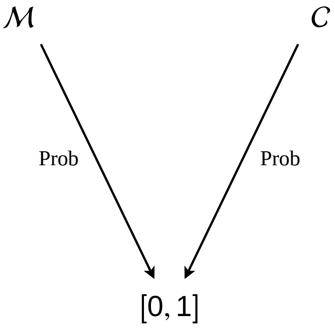
- the messages are drawn from a source $\mathcal{X}$ and coded along $f : \mathcal{X} \longrightarrow \mathcal{G} \subseteq \mathcal{M}^*$
- the frequency distribution $\mathrm{Prob}_{\mathcal{X}} : \mathcal{X} \longrightarrow [0,1]$ induces the frequency distribution $\mathrm{Prob}_{\mathcal{M}} : \mathcal{M} \longrightarrow [0,1]$

$$\mathrm{Prob}_{\mathcal{M}}\left(\vec{m}\right) = \mathrm{Prob}_{\mathcal{X}}\left(f^{-1}(\vec{m})\right)$$

- the frequency distribution $\mathrm{Prob}_{\mathcal{C}} : \mathcal{C} \longrightarrow [0,1]$ can be extracted if there is enough cyphertext

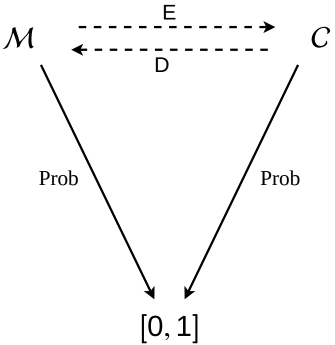
COA on substitution cypher

The patterns



COA on substitution cypher

The patterns are aligned to reconstruct



KPA on the one-time-pad

- $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}^N$
- $\mathsf{E}(\vec{k}, \vec{m}) = \vec{m} + \vec{k}$
- $\mathsf{D}(\vec{k}, \vec{c}) = \vec{c} - \vec{k}$

KPA on the one-time-pad

- $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}^N$
- $\mathsf{E}(\vec{k}, \vec{m}) = \vec{m} + \vec{k}$
- $\mathsf{D}(\vec{k}, \vec{c}) = \vec{c} - \vec{k}$

Attack

Given $\vec{m}$ and $\mathsf{E}(\vec{k}, \vec{m}) = \vec{m} + \vec{k}$ the cryptanalyst derives

$$\vec{k} = \mathsf{E}(\vec{k}, \vec{m}) - \vec{m}$$

Can we prove that there are no attacks?

Can we prove that there are no attacks?

Proposition

If all keys are equally likely, then the one-time-pad is secure, in the sense that the cyphertext provides no information about the plaintext.

Can we prove that there are no attacks?

We need tools for such proofs!

Guessing

Attack scenario: KPA, CPA

The cryptanalyst knows which crypto system is used. He wants to derive the key from the known or chosen plaintext, and its encryptions

$$m_1, \ldots, m_\ell, \mathsf{E}(\mathsf{K}, m_1), \ldots, \mathsf{E}(\mathsf{K}, m_\ell) \quad \vdash \quad \mathsf{K}$$

Guessing

Attack scenario: KPA, CPA

The cryptanalyst knows which crypto system is used. He wants to derive the key from the known or chosen plaintext, and its encryptions

$$m_1, \ldots, m_\ell, \mathsf{E}(\mathsf{K}, m_1), \ldots, \mathsf{E}(\mathsf{K}, m_\ell) \quad \vdash \quad \mathsf{K}$$

In some cases, he

- may not know the plaintext, but
- can recognize well-formed messages.

Guessing

Terminology

A *random variable* is a function $X : \mathcal{X} \longrightarrow V$ where

- $\mathcal{X}$ is a source and
- V is a set, representing values.

Guessing

Terminology

A *random variable* is a function $X : \mathcal{X} \longrightarrow V$ where

- $\mathcal{X}$ is a source and
- V is a set, representing values.

Notation

We write

$$\begin{aligned} \mathrm{Prob}(X = v) &= \mathrm{Prob}\{x \in \mathcal{X} \mid X(x) = v\} \\ &= \sum_{X(x)=v} \mathrm{Prob}(x) \end{aligned}$$

Guessing

Guessing process

Given a probability distribution over the key space $\mathcal{K}$, a *guessing attack* is a random variable $G : \mathcal{K}^* \longrightarrow \mathbb{N}$, where

$$G(k_1, k_2, \ldots, k_n) \quad = i$$

means that $k_i = \mathrm{K_D}$.

Guessing

Guessing process

Given a probability distribution over the key space $\mathcal{K}$, a *guessing attack* is a random variable $G : \mathcal{K}^* \longrightarrow \mathbb{N}$, where

$$G(k_1, k_2, \ldots, k_n) \quad = i$$

means that $k_i = \mathrm{K_D}$.

Remark

The intuition is that we are given some cyphertext $\vec{c}$, and we test whether $\mathrm{D}(k_i, \vec{c})$ is a well-formed message for one k_i after the other.

Guessing

Exercise

Suppose that there are $\ell = \#\mathcal{K}$ keys, and that they are all equally likely. What is the probability that

- $G = 1$, i.e. the key is guessed at once,
- $G = n$, i.e. the key is guessed after exactly n tries.
- $G \leq n$, i.e. the key is guessed in at most n tries.

Guessing

Solution

- Since there are $\ell = \#\mathcal{K}$ equally likely keys,
  - the probability that the right key is drawn at once is $\mathrm{Prob}(G = 1) = p_1 = \frac{1}{\ell}$;

Guessing

Solution

- Since there are $\ell = \#\mathcal{K}$ equally likely keys,
  - the probability that the right key is drawn at once is $\mathrm{Prob}(G = 1) = p_1 = \frac{1}{\ell}$;
  - the probability that the right key is *not* drawn at once is $q_1 = \mathrm{Prob}(G \neq 1) = 1 - p_1 = \frac{\ell - 1}{\ell}$.

Guessing

Solution

- Since there are $\ell = \#\mathcal{K}$ equally likely keys,
  - the probability that the right key is drawn at once is $\mathrm{Prob}(G = 1) = p_1 = \frac{1}{\ell}$;
  - the probability that the right key is *not* drawn at once is $q_1 = \mathrm{Prob}(G \neq 1) = 1 - p_1 = \frac{\ell-1}{\ell}$. In this case, we draw again, from $\ell - 1$ untested keys.

Guessing

Solution

- Since there are $\ell = \#\mathcal{K}$ equally likely keys,
  - the probability that the right key is drawn at once is $\mathrm{Prob}(G = 1) = p_1 = \frac{1}{\ell}$;
  - the probability that the right key is *not* drawn at once is $q_1 = \mathrm{Prob}(G \neq 1) = 1 - p_1 = \frac{\ell-1}{\ell}$. In this case, we draw again, from $\ell - 1$ untested keys. This time,
    - the probability that the right key is drawn immediately is now $p_2 = \frac{1}{\ell-1}$, and thus $\mathrm{Prob}(G = 2) = q_1 \cdot p_2 = \frac{\ell-1}{\ell} \cdot \frac{1}{\ell-1} = \frac{1}{\ell}$;

Guessing

Solution

- Since there are $\ell = \#\mathcal{K}$ equally likely keys,
  - the probability that the right key is drawn at once is $\mathrm{Prob}(G = 1) = p_1 = \frac{1}{\ell}$;
  - the probability that the right key is *not* drawn at once is $q_1 = \mathrm{Prob}(G \neq 1) = 1 - p_1 = \frac{\ell-1}{\ell}$. In this case, we draw again, from $\ell - 1$ untested keys. This time,
    - the probability that the right key is drawn immediately is now $p_2 = \frac{1}{\ell-1}$, and thus $\mathrm{Prob}(G = 2) = q_1 \cdot p_2 = \frac{\ell-1}{\ell} \cdot \frac{1}{\ell-1} = \frac{1}{\ell}$;
    - whereas the probability that the right key is still not drawn is $q_2 = \frac{\ell-2}{\ell-1} \ldots$

Guessing

In general, with $p_i = \frac{1}{\ell - i + 1}$ and $q_i = \frac{\ell - i}{\ell - i + 1}$, the probability that a particular key is drawn in the n-th draw is

$$\begin{aligned} \mathrm{Prob}(G = n) &= q_1 \cdot q_2 \cdots q_{n-1} \cdot p_n \\ &= \frac{\ell - 1}{\ell} \cdot \frac{\ell - 2}{\ell - 1} \cdots \frac{\ell - n + 1}{\ell - n + 2} \cdot \frac{1}{\ell - n + 1} \\ &= \frac{1}{\ell} \end{aligned}$$

Guessing

In general, with $p_i = \frac{1}{\ell - i + 1}$ and $q_i = \frac{\ell - i}{\ell - i + 1}$, the probability that a particular key is drawn in the n-th draw is

$$
\begin{aligned}
\mathrm{Prob}(G = n) &= q_1 \cdot q_2 \cdots q_{n-1} \cdot p_n \\
&= \frac{\ell - 1}{\ell} \cdot \frac{\ell - 2}{\ell - 1} \cdots \frac{\ell - n + 1}{\ell - n + 2} \cdot \frac{1}{\ell - n + 1} \\
&= \frac{1}{\ell}
\end{aligned}
$$

The probability that a particular key is drawn in at most n tries is

$$
\mathrm{Prob}(G \leq n) = \sum_{i=1}^{n} \mathrm{Prob}(G = i) = \frac{n}{\ell}
$$

Elements of probability

Notation

Given a source $\mathcal{X}$ and events $\alpha, \beta, \gamma \ldots \subseteq \mathcal{X}$, we write

$$\begin{aligned} \left[\alpha\right] &= \sum_{x \in \alpha} \mathrm{Prob}(x) \\ \left[\alpha \vdash \beta\right] &= \frac{\left[\alpha \cap \beta\right]}{\left[\alpha\right]} \end{aligned}$$

Elements of probability

Remark

Traditionally, our $\left[\alpha \vdash \beta\right]$ is written $\mathrm{Prob}\left(\beta \mid \alpha\right)$, and called conditional probability.

Elements of probability

Remark

Traditionally, our $\left[\alpha \vdash \beta\right]$ is written $\mathrm{Prob}\left(\beta \mid \alpha\right)$, and called conditional probability.

While the traditional notations need to be respected, cryptography puts conditional probability to heavy use, and abuse.

Elements of probability

Remark

Traditionally, our $\left[\alpha \vdash \beta\right]$ is written $\mathrm{Prob}\,(\beta \mid \alpha)$, and called conditional probability.

While the traditional notations need to be respected, cryptography puts conditional probability to heavy use, and abuse.

$\left[\alpha \vdash \beta\right]$ tells how likely it is to guess β from α.

Elements of probability

Homework

$$[\alpha \vdash \neg\beta] = 1 - [\alpha \vdash \beta]$$

$$[\beta] = [\alpha] \cdot [\alpha \vdash \beta] + [\neg\alpha] \cdot [\neg\alpha \vdash \beta]$$

$$[\alpha \vdash \beta \cup \gamma] = [\alpha \vdash \beta] + [\alpha \vdash \gamma] - [\alpha \vdash \beta \cap \gamma]$$

Elements of probability

Homework

$$[\alpha \vdash \neg\beta] = 1 - [\alpha \vdash \beta]$$

$$[\beta] = [\alpha] \cdot [\alpha \vdash \beta] + [\neg\alpha] \cdot [\neg\alpha \vdash \beta]$$

$$[\alpha \vdash \beta \cup \gamma] = [\alpha \vdash \beta] + [\alpha \vdash \gamma] - [\alpha \vdash \beta \cap \gamma]$$

Moreover

$$[\alpha \cap \beta] = [\alpha] \cdot [\beta] \iff [\alpha \vdash \beta] = [\beta]$$

$$\iff [\beta \vdash \alpha] = [\alpha]$$

Elements of probability

Bayes theorem

$$[\beta \vdash \alpha] \quad = \quad \frac{[\alpha][\alpha \vdash \beta]}{[\alpha][\alpha \vdash \beta] + [\neg\alpha][\neg\alpha \vdash \beta]}$$

Elements of probability

Proposition

$$
\begin{array}{ccc}
\left[\beta \vdash \alpha\right] & = & \left[\gamma \vdash \alpha\right] \\
 & \Downarrow & \\
\left[\alpha \vdash \beta\right] \cdot \left[\beta \vdash \gamma\right] & = & \left[\alpha \vdash \gamma\right] \cdot \left[\gamma \vdash \beta\right]
\end{array}
$$

Elements of probability

Proposition

Since

$$\left[\alpha \vdash \beta \cap \gamma\right] \quad = \quad \left[\alpha \vdash \beta\right] \cdot \left[\alpha \cap \beta \vdash \gamma\right]$$

it follows that

$$\left[\alpha \vdash \beta\right] \cdot \left[\alpha \cap \beta \vdash \gamma\right] \quad \leq \quad \left[\alpha \vdash \gamma\right]$$

with the equality when $\left[\alpha \cap \gamma \vdash \beta\right] = 1$, *so that* $\left[\alpha \vdash \gamma\right] = \left[\alpha \vdash \beta \cap \gamma\right]$.

Problem with simple crypto systems

Leaking partial information

The trapdoor decryption condition

$$\forall m. A(E(K_E, m)) = m \quad \Longrightarrow \quad \forall c. A(c) = D(K_D, c)$$

only talks about *total* decryptions.

Problem with simple crypto systems

Leaking partial information

The trapdoor decryption condition

$$\forall m.\mathsf{A}(\mathsf{E}(\mathsf{K_E}, m)) = m \implies \forall c.\mathsf{A}(c) = \mathsf{D}(\mathsf{K_D}, c)$$

only talks about *total* decryptions.

A simple crypto system can leak *partial* information.

Problem with simple crypto systems

Two kinds of leaks

The attacker may observe traffic and build

- a *partial* map $A : \mathcal{C} \rightharpoonup \mathcal{M}$
  - e.g., by recognizing $E(K, \text{"yes"}), E(K, \text{"no"}), E(K, \text{"buy"}) \ldots$
- a map $A : \mathcal{C} \longrightarrow \Delta\mathcal{M}$, extracting *partial information*
  - e.g., by comparing $E(K, m_0), E(K, m_1) \ldots$

Example: Reusing one-time-pad

Proposition

If the same one-time-pad key is used to encrypt more than one block, then a CPA attacker can extract partial information.

Example: Reusing one-time-pad

Proposition

If the same one-time-pad key is used to encrypt more than one block, then a CPA attacker can extract partial information.

E.g., the attacker can form two messages such that, if she is given the encryption of one of them, then she can tell which one. (This is one bit of information extracted.)

Example: Reusing one-time-pad

Proof

The CPA attacker forms two messages in the form:

$$\vec{m}_0 = \vec{m}@\vec{m} \qquad \vec{m}_1 = \vec{m}@\vec{\ell}$$

where $\vec{x}@\vec{y}$ is concatenation and $\vec{\ell} \neq \vec{m}$ are of length N.

Example: Reusing one-time-pad

Proof

The CPA attacker forms two messages in the form:

$$\vec{m}_0 = \vec{m}@\vec{m} \qquad\qquad \vec{m}_1 = \vec{m}@\vec{\ell}$$

where $\vec{x}@\vec{y}$ is concatenation and $\vec{\ell} \neq \vec{m}$ are of length N.

Encrypting with the key $\vec{k}$ of length N gives

$$\mathsf{E}(\vec{k}, \vec{m}_0) = \vec{c}@\vec{c} \qquad\qquad \mathsf{E}(\vec{k}, \vec{m}_1) = \vec{c}@\vec{d}$$

where $\vec{c} = \vec{m} + \vec{k}$ and $\vec{d} = \vec{m} + \vec{\ell}$.

Probabilistic crypto system

Definition

Given the types

- $\mathcal{M}$ of *messages* (or *plaintexts*)
- $\mathcal{C}$ of *cyphertexts*
- $\mathcal{K}$ of *keys*
- $\mathcal{R}$ of *random seeds*

Probabilistic crypto system

Definition

...a probabilistic crypto-system is a triple of algorithms:

- key generation $\langle \mathrm{K_E}, \mathrm{K_D} \rangle : \mathcal{R} \longrightarrow \mathcal{K} \times \mathcal{K}$,
- encryption $\mathrm{E} : \mathcal{R} \times \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$, and
- decryption $\mathrm{D} : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$,

When no confusion seems likely, we abbreviate

- $\mathrm{K}(r)$ to $\mathbb{K}$ and
- $\mathrm{E}(r, k, m)$ to $\mathbb{E}(k, m)$ and even $\mathbb{E}(m)$.

Probabilistic crypto system

Definition

...that together provide

- unique decryption:

$$\mathsf{D}(\mathbb{K}_{\mathsf{D}}, \mathbb{E}(\mathbb{K}_{\mathsf{E}}, m)) \quad = \quad m$$

- secrecy (Shannon: "unconditional security"):

$$\left[c \in \mathbb{E}(\mathbb{K}, m) \vdash m \in \mathcal{M}\right] = \left[m \in \mathcal{M}\right] \qquad \text{(IT-SEC)}$$

Probabilistic crypto system

Definition

...that together provide

- unique decryption:

$$\mathrm{D}(\mathbb{K}_{\mathrm{D}}, \mathbb{E}(\mathbb{K}_{\mathrm{E}}, m)) \quad = \quad m$$

- secrecy:

$$\left[c \in \mathbb{E}(\mathbb{K}, m) \vdash m \in \mathbb{A}(c)\right] = \left[m \in \mathbb{A}(0)\right] \quad \text{(COM-SEC)}$$

for every feasible probabilistic algorithm $\mathbb{A} : \mathcal{C} \longrightarrow \mathcal{M}$,
(i.e. $A : \mathcal{R} \times \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$)

Probabilistic crypto system

Definition

...that together provide

- unique decryption:

$$D(\mathbb{K}_D, \mathbb{E}(\mathbb{K}_E, m)) \quad = \quad m$$

- secrecy:

$$\left[m_0, m_1 \in \mathcal{M}, c \in \mathbb{E}(\mathbb{K}, m_b) \vdash b \in \{0,1\}\right] =$$

$$\left[m_0, m_1 \in \mathcal{M} \vdash b \in \{0,1\}\right] = \frac{1}{2} \qquad \text{(IT-IND)}$$

Probabilistic crypto system

Definition

. . . that together provide

- unique decryption:

$$\mathsf{D}(\mathbb{K}_\mathsf{D}, \mathbb{E}(\mathbb{K}_\mathsf{E}, m)) \quad = \quad m$$

- secrecy:

$$\left[m_0, m_1 \in \mathcal{M}, c \in \mathbb{E}(m_b) \vdash b \in \mathbb{A}(m_0, m_1, c)\right] \leq$$

$$\left[m_0, m_1 \in \mathcal{M} \vdash b \in \mathbb{A}(m_0, m_1, 0)\right] \leq \frac{1}{2} \quad \text{(COM-IND)}$$

for any feasible probabilistic $\mathbb{A} : \mathcal{M} \times \mathcal{M} \times \mathcal{C} \longrightarrow \{0, 1\}$ (with K_E and the seed implicit)

Probabilistic crypto system

Definition

...that together provide

- unique decryption:

$$\mathsf{D}(\mathbb{K}_D, \mathbb{E}(\mathbb{K}_E, m)) = m$$

- secrecy (Goldwasser-Micali: "semantic security")

$$\Big[m_0, m_1 \in \mathbb{A}_0, c \in \mathbb{E}(m_b) \vdash b \in \mathbb{A}_1(m_0, m_1, c)\Big] \leq \frac{1}{2} \quad \text{(IND-CPA)}$$

for any probabilistic algorithm $\mathbb{A} = \langle \mathbb{A}_0, \mathbb{A}_1 \rangle \ldots$

Probabilistic crypto system

Definition

. . . that together provide

- unique decryption:

$$\mathsf{D}(\mathbb{K}_D, \mathbb{E}(\mathbb{K}_E, m)) \quad = \quad m$$

- secrecy (under chosen cyphertext attack):

$$\left[\begin{array}{l} c_0 \in \mathbb{A}_0,\ m \in \mathsf{D}(c_0), \\ m_0, m_1 \in \mathbb{A}_1(c_0, m), c \in \mathbb{E}(m_b) \end{array} \right. \vdash$$

$$\left. b \in \mathbb{A}_2(c_0, m, m_0, m_1, c) \right] \quad \leq \quad \frac{1}{2} \quad \text{(IND-CCA)}$$

for any probabilistic algorithm $\mathbb{A} = \langle \mathbb{A}_0, \mathbb{A}_1, \mathbb{A}_2 \rangle \ldots$

Probabilistic crypto system

Definition

...that together provide

- unique decryption:

$$\mathsf{D}(\mathbb{K}_D, \mathbb{E}(\mathbb{K}_E, m)) \quad = \quad m$$

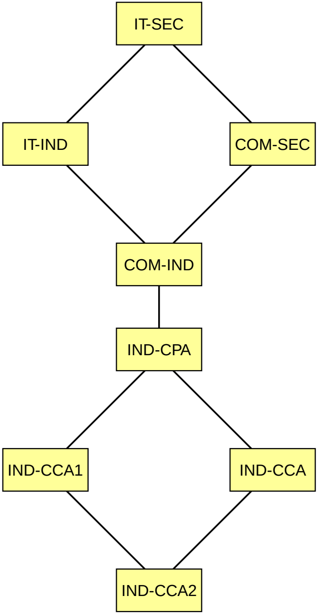
- secrecy (under *adaptive* chosen cyphertext attack):

$$\left[\begin{array}{l} c_0 \in \mathbb{A}_0,\ m \in \mathsf{D}(c_0), \\ m_0, m_1 \in \mathbb{A}_1(c_0, m), c \in \mathbb{E}(m_b) \\ c_1 \in \mathbb{A}_2(c_0, m, m_0, m_1), \widetilde{m} \in \mathsf{D}(c_1 \neq c) \end{array} \right. \qquad \vdash$$

$$b \in \mathbb{A}_3(c_0, m, m_0, m_1, c, c_1, \widetilde{m}) \Big] \quad \leq \quad \frac{1}{2} \qquad \text{(IND-CCA2)}$$

for any probabilistic algorithm $\mathbb{A} = \langle \mathbb{A}_0, \mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3 \rangle \ldots$

Taxonomy of secrecy properties



Example: El Gamal

Fix a finite field $\mathbb{F}$ and $g \in \mathbb{F}^*$.

$$\mathcal{M} = \mathcal{R} = \mathbb{F} \qquad K_E(a) = g^a$$

$$\mathcal{C} = \mathbb{F}^* \times \mathbb{F} \qquad K_D(a) = a$$

$$\mathcal{K} = \mathbb{F}^* \times \mathbb{F}^* \qquad E(r,k,m) = \left\langle g^r, k^r \cdot m \right\rangle$$

$$D\left(\overline{k}, \langle c_1, c_2 \rangle\right) = \frac{c_2}{c_1^{\overline{k}}}$$

Example: El Gamal

Fix a finite field $\mathbb{F}$ and $g \in \mathbb{F}^*$.

$$\mathcal{M} = \mathcal{R} = \mathbb{F} \qquad \mathsf{K_E}(a) = g^a$$

$$\mathcal{C} = \mathbb{F}^* \times \mathbb{F} \qquad \mathsf{K_D}(a) = a$$

$$\mathcal{K} = \mathbb{F}^* \times \mathbb{F}^* \qquad \mathsf{E}(r,k,m) = \left\langle g^r, k^r \cdot m \right\rangle$$

$$\mathsf{D}\left(\overline{k}, \langle c_1, c_2 \rangle\right) = \frac{c_2}{c_1^{\overline{k}}}$$

Unique decryption

$$\begin{aligned}
\mathsf{D}\left(\mathsf{K_D}(a), \mathsf{E}(r, \mathsf{K_E}(a), m)\right) &= \mathsf{D}\left(a, \mathsf{E}(r, g^a, m)\right) \\
&= \mathsf{D}\left(a, \left\langle g^r, (g^a)^r \cdot m \right\rangle\right) \\
&= \frac{g^{ar} \cdot m}{(g^r)^a} = m
\end{aligned}$$

Perfect security of one-time-pad

Proposition

If all keys are equally likely, then the one-time-pad is unconditionally secure, i.e. it satisfies (IT-SEC).

Perfect security of one-time-pad

Proposition

If all keys are equally likely, then the one-time-pad is unconditionally secure, i.e. it satisfies (IT-SEC).

Proof

$\left[c \in \mathcal{C} \vdash m \in \mathcal{M}\right] = \left[m \in \mathcal{M}\right]$ follows from
$\left[m \in \mathcal{M} \vdash c \in \mathcal{C}\right] = \left[c \in \mathcal{C}\right]$ because

$$\left[c \in \mathcal{C} \vdash m \in \mathcal{M}\right] \quad = \quad \frac{\left[m \in \mathcal{M}\right] \cdot \left[m \in \mathcal{M} \vdash c \in \mathcal{C}\right]}{\left[c \in \mathcal{C}\right]}$$

...

Perfect security of one-time-pad

Proof (continued)

On one hand, it is obvious that for all messages *m* and cyphertexts *c* holds

$$\left[m \in \mathcal{M} \vdash c \in \mathcal{C}\right] \;=\; \left[k = c - m \in \mathcal{K}\right] \;=\; \frac{1}{26^N}$$

Perfect security of one-time-pad

Proof (continued)

On the other hand, we have

$$\begin{aligned}
[c \in \mathcal{C}] &= \sum_{m+k=c} [m \in \mathcal{M}] \cdot [k \in \mathcal{K}] \\
&= \sum_{m \in \mathcal{M}} [m \in \mathcal{M}] \cdot [c - m \in \mathcal{K}] \\
&= \frac{1}{26^N} \sum_{m \in \mathcal{M}} [m \in \mathcal{M}] \\
&= \frac{1}{26^N}
\end{aligned}$$

Security of El Gamal

Computational Diffie-Hellman Assumption (CDH)

There is no feasible probabilistic algorithm CDH : $\mathbb{F}^2 \longrightarrow \mathbb{F}$ such that for all $a, b \in \mathbb{F}$ holds with a high probability

$$\mathrm{CDH}(g^a, g^b) \quad = \quad g^{ab}$$

Security of El Gamal

Computational Diffie-Hellman Assumption (CDH)

There is no feasible probabilistic algorithm $\text{CDH} : \mathbb{F}^2 \longrightarrow \mathbb{F}$ such that for all $a, b \in \mathbb{F}$ holds with a high probability

$$\text{CDH}(g^a, g^b) \quad = \quad g^{ab}$$

Decision Diffie-Hellman Assumption (DDH)

There is no feasible prob. algorithm $\text{DDH} : \mathbb{F}^3 \longrightarrow \{0, 1\}$ such that for all $a, b \in \mathbb{F}$ holds with a probability $> \frac{1}{2}$

$$\text{DDH}(x, y, z) \quad = \quad \begin{cases} 1 & \text{if } \exists uv.\, x = g^u \wedge y = g^v \wedge z = g^{uv} \\ 0 & \text{otherwise} \end{cases}$$

Security of El Gamal

Proposition

El Gamal satisfies (IND-CPA) if and only if (DDH) holds.
El Gamal does not safisty (IND-CCA).

Security of El Gamal

Recall the definitions:

. . .

- unique decryption:

$$\mathsf{D}(\mathbb{K}_{\mathsf{D}}, \mathbb{E}(\mathbb{K}_{\mathsf{E}}, m)) \quad = \quad m$$

- secrecy (Goldwasser-Micali: "semantic security")

$$\Big[m_0, m_1 \in \mathbb{A}_0, c \in \mathbb{E}(m_b) \vdash b \in \mathbb{A}_1(m_0, m_1, c)\Big] \;\leq\; \frac{1}{2} \quad \text{(IND-CPA)}$$

for any probabilistic algorithm $\mathbb{A} = \langle \mathbb{A}_0, \mathbb{A}_1 \rangle \ldots$

Security of El Gamal

Recall the definitions:

...

- unique decryption:

$$\mathrm{D}(\mathbb{K}_D, \mathbb{E}(\mathbb{K}_E, m)) \quad = \quad m$$

- secrecy (under chosen cyphertext attack):

$$\left[\begin{array}{l} c_0 \in \mathbb{A}_0,\ m \in \mathrm{D}(c_0), \\ m_0, m_1 \in \mathbb{A}_1(c_0, m), c \in \mathbb{E}(m_b) \end{array} \quad \vdash \right.$$

$$\left. b \in \mathbb{A}_2(c_0, m,\ m_0, m_1, c) \right] \quad \leq \quad \frac{1}{2} \quad \text{(IND-CCA)}$$

for any probabilistic algorithm $\mathbb{A} = \langle \mathbb{A}_0, \mathbb{A}_1, \mathbb{A}_2 \rangle \ldots$

Security of El Gamal

Proof of (DDH)⇒(IND-CPA)

Suppose $\neg$(IND-CPA).

Security of El Gamal

Proof of (DDH)⇒(IND-CPA)

Suppose ¬(IND-CPA).

This means that there is a feasible probabilistic algorithm $\mathbb{A} = \langle \mathbb{A}_0, \mathbb{A}_1 \rangle$ which

- generates $m_0, m_1 \in \mathbb{A}_0(k)$, and then
- guesses $b \in \mathbb{A}_1(k, m_0, m_1, c_b)$ with a probability $> \frac{1}{2}$
  - where $c_b = \mathrm{E}(s, k, m_b)$ for $b \in \{0, 1\}$.

Security of El Gamal

Proof of (DDH)⇒(IND-CPA)

Suppose ¬(IND-CPA).

This means that there is a feasible probabilistic algorithm $\mathbb{A} = \langle \mathbb{A}_0, \mathbb{A}_1 \rangle$ which

- generates $m_0, m_1 \in \mathbb{A}_0(k)$, and then
- guesses $b \in \mathbb{A}_1(k, m_0, m_1, c_b)$ with a probability $> \frac{1}{2}$
  - where $c_b = \mathsf{E}(s, k, m_b)$ for $b \in \{0, 1\}$.

We construct the algorithm $\mathsf{DDH} : \mathbb{F}^3 \longrightarrow \{0, 1\}$ to decide whether a triple $\langle x, y, z \rangle$ is in the form $\langle g^u, g^v, g^{uv} \rangle$ for some $u, v \in \mathbb{F}$.

Security of El Gamal

Proof (continued)

If the private key $K_D = u$, then El Gamal encrypts

$$E(v, g^u, m) \quad = \quad \langle g^v, g^{uv} \cdot m \rangle$$

Security of El Gamal

Proof (continued)

If the private key $K_D = u$, then El Gamal encrypts

$$E(v, g^u, m) \quad = \quad \langle g^v, g^{uv} \cdot m \rangle$$

This means that

$$DDH(x, y, z) = 1 \quad \iff \quad \forall m. \mathbb{E}(x, m) = \langle y, z \cdot m \rangle$$

Security of El Gamal

Proof (continued)

If the private key $K_D = u$, then El Gamal encrypts

$$E(v, g^u, m) \quad = \quad \langle g^v, g^{uv} \cdot m \rangle$$

This means that

$$DDH(x, y, z) = 1 \quad \iff \quad \forall m. \mathbb{E}(x, m) = \langle y, z \cdot m \rangle$$

But ¬(IND-CPA) says that $\mathbb{A} = \langle A_0, A_1 \rangle$ can decide the right-hand side, so that $m_0, m_1 \in A_0(x)$ gives

$$DDH(x, y, z) \quad = \quad \begin{cases} 1 & \text{if } A_1(x, m_0, m_1, \langle y, z \cdot m_0 \rangle) = 0 \\ & \text{and } A_1(x, m_0, m_1, \langle y, z \cdot m_1 \rangle) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Security of El Gamal

Homework

Complete the proof of the Proposition, showing that

- (IND-CPA)$\Rightarrow$(DDH)
- (IND-CCA) does not hold.

Outline

Information, channel security, noninterference

Encryption and decryption

Cryptanalysis and notions of secrecy

Cyphers and modes of operation

Key establishment

What did we learn?

Outline

Information, channel security, noninterference

Encryption and decryption

Cryptanalysis and notions of secrecy

Cyphers and modes of operation

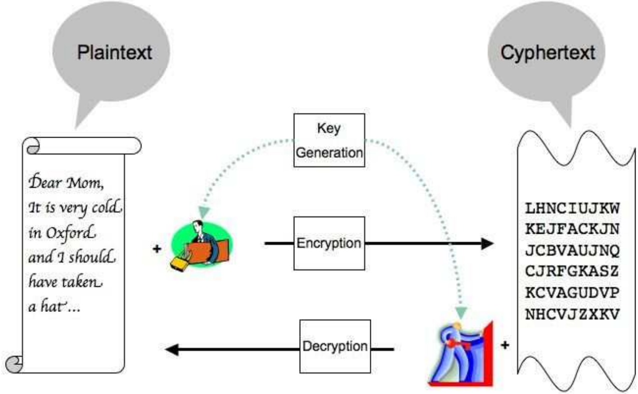
Key establishment

"Programming Satan's computer"

Diffie-Hellman Key Agreement

Needham-Schroeder Public Key Protocol

Key establishment



Where do the keys come from?

Key establishment

- Traditionally, keys sent through a secure channel
  - messenger, direct handover, physical protection

Key establishment

- Traditionally, keys sent through a secure channel
  - messenger, direct handover, physical protection
- In cyberspace, there are no secure channels
  - only you and me and cryptography

Key establishment in cyberspace

What is cyberspace?

Key establishment in cyberspace

What is cyberspace?

- space of costless communication
  - instantaneous message delivery
  - any two nodes are neighbors: no notion of distance

Key establishment in cyberspace

What is cyberspace?

- space of costless communication
  - instantaneous message delivery
  - any two nodes are neighbors: no notion of distance
- end-to-end architecture (TCP, UDP)
  - simple network links
  - smart network nodes ("ends")

Key establishment in cyberspace

What is cyberspace?

- space of costless communication
  - instantaneous message delivery
  - any two nodes are neighbors: no notion of distance
- end-to-end architecture (TCP, UDP)
  - simple network links
  - smart network nodes ("ends")
- "Satan's computer" (Ross Anderson)
  - network controlled by the adversaries: Eve, Satan
  - security only through crypto at the "ends"

Key establishment in cyberspace

Generate your own public key

- **El Gamal:** Alice generates $\mathrm{K} = \langle g^a, a \rangle$
  - she picks $K_{\mathrm{D}} = a$
  - computes $K_{\mathrm{E}} = g^a$ and
  - sends K_{E} to Bob

Key establishment in cyberspace

Generate your own public key

- **El Gamal:** Alice generates $\mathrm{K} = \langle g^a, a\rangle$
  - she picks $K_D = a$
  - computes $K_E = g^a$ and
  - sends K_E to Bob
- **RSA**: Alice generates $\mathrm{K} = \big\langle \langle n, e\rangle, d\big\rangle$
  - she picks large primes p and q and sets $n = pq$
  - picks $e \in \mathbb{Z}^*_{(p-1)(q-1)}$
  - computes $\mathrm{K_D} = d = e^{-1} \mod (p-1)(q-1)$
  - sends $\mathrm{K_E} = \langle n, e\rangle$ to Bob

Key establishment in cyberspace

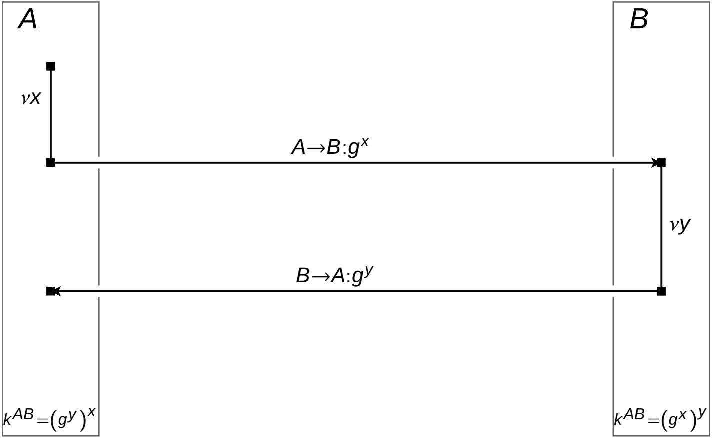
Problem

Eve can impersonate Alice

- Eve can generate K_E and K_D,
- send K_D to Bob
- and say *"Hi, Alice here, this is my key"*.
  - Bob encrypts his messages to Alice by K_E
  - Eve decrypts them by K_D.

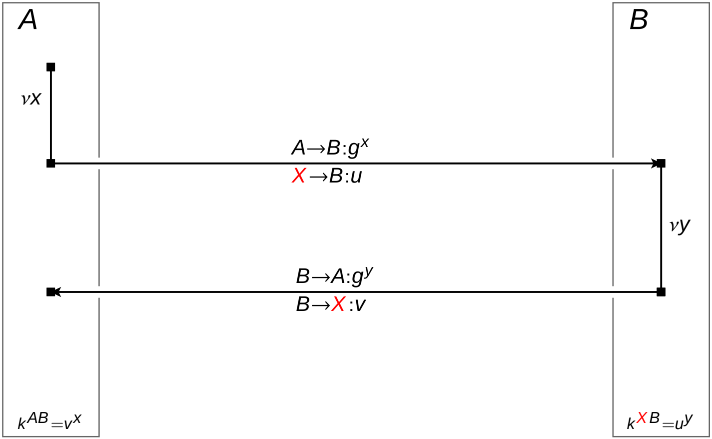
Two party key agreement

Diffie-Hellman Key Agreement Protocol (DHKA)



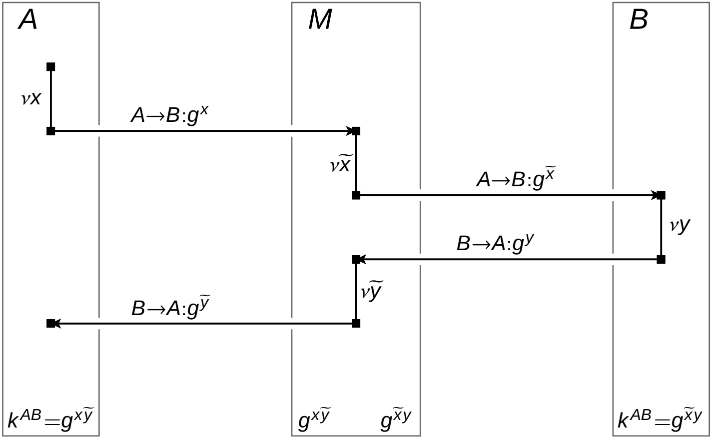
Two party key agreement

Diffie-Hellman Key Agreement Protocol (DHKA)



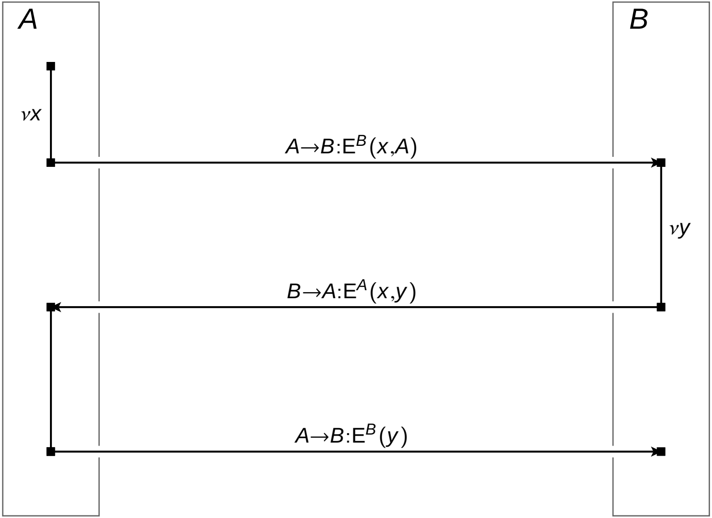
Two party key agreement

Attack on DHKA



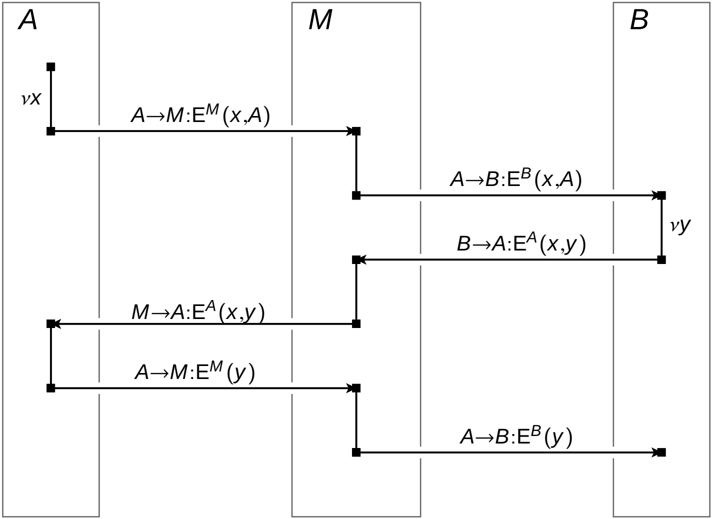
Bootstrapping key agreement

Needham-Schroeder Public Key Protocol (NSPK)



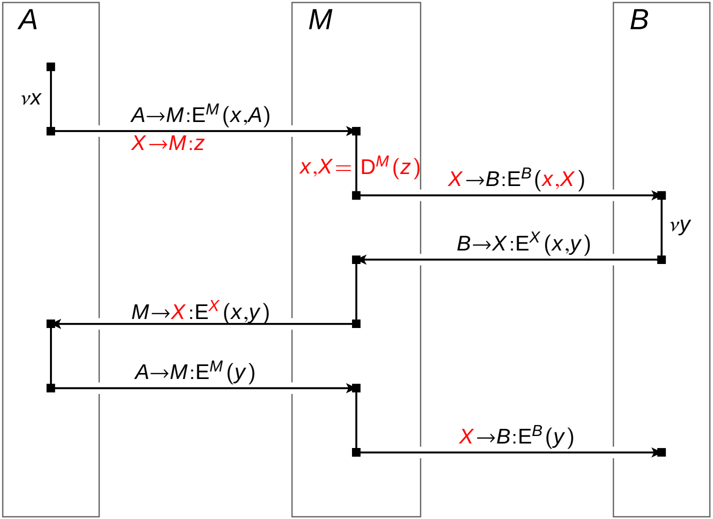
Bootstrapping key agreement

Attack on NSPK



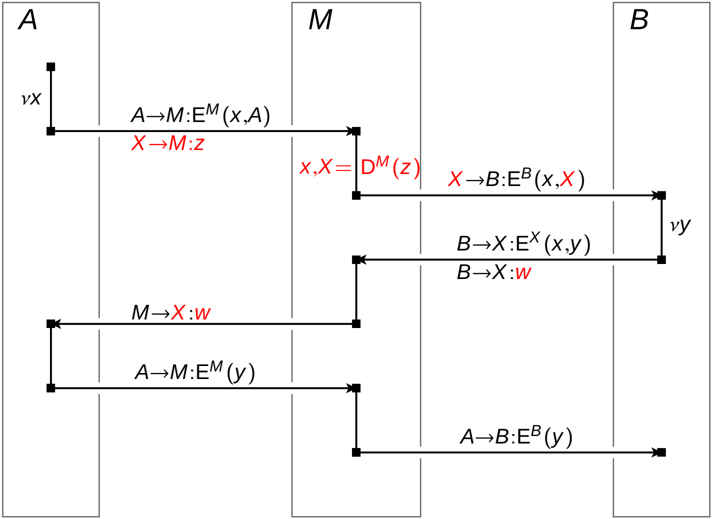
Bootstrapping key agreement

Attack on NSPK



Bootstrapping key agreement

Attack on NSPK



Bootstrapping key agreement

History of NSPK

- NSPK was proposed by in a seminal paper in 1978.

Bootstrapping key agreement

History of NSPK

- NSPK was proposed by in a seminal paper in 1978.
- It was often used and studied.

Bootstrapping key agreement

History of NSPK

- NSPK was proposed by in a seminal paper in 1978.
- It was often used and studied.
- In 1996, Gavin Lowe found the attack
  - using the FDR (Failure Divergence Refinement) checker
  - as a part of his project work at Comlab

Bootstrapping key agreement

History of NSPK

- NSPK was proposed by in a seminal paper in 1978.
- It was often used and studied.
- In 1996, Gavin Lowe found the attack
  - using the FDR (Failure Divergence Refinement) checker
  - as a part of his project work at Comlab
- Later he built Casper.
- More at practicals!

Outline

Information, channel security, noninterference

Encryption and decryption

Cryptanalysis and notions of secrecy

Cyphers and modes of operation

Key establishment

What did we learn?

Lessons about the bad information flows

- information leaks through interference of resources
  - covert channels are hard to eliminate
  - formal models help prevent Trojan intrusions

Lessons about the bad information flows

- information leaks through interference of resources
  - covert channels are hard to eliminate
  - formal models help prevent Trojan intrusions
- secrecy is achieved in complicated ways
  - some of the "purest" maths became the most applied
  - public key crypto needed a public science of crypto

Lessons about the bad information flows

- information leaks through interference of resources
  - covert channels are hard to eliminate
  - formal models help prevent Trojan intrusions
- secrecy is achieved in complicated ways
  - some of the "purest" maths became the most applied
  - public key crypto needed a public science of crypto
- but cryptanalysis is also hard
  - encryptions are not broken every day
  - most security failures arise from **protocol failures**

Lessons about computation

Lessons about computation

- The simple insights that
  - some computations are hard to invert
    - e.g., getting p or q from pq, or a from g^a and g
  - some informations are hard to guess
    - if the source is large and unbiased

Lessons about computation

- The simple insights that
  - some computations are hard to invert
    - e.g., getting p or q from pq, or a from g^a and g
  - some informations are hard to guess
    - if the source is large and unbiased
- point to the important lesson that
  - **complexity** and
  - **randomness**

  are **powerful computational resources**.

Lessons about computation

- The simple insights that
  - some computations are hard to invert
    - e.g., getting p or q from pq, or a from g^a and g
  - some informations are hard to guess
    - if the source is large and unbiased
- point to the important lesson that
  - **complexity** and
  - **randomness**

  are **powerful computational resources**.
- The negative can be used as the positive.

...are used to push good information flows

- The absence of bad information flows
- is a fulcrum to move the good information flows.

. . . are used to push good information flows

- The absence of bad information flows
  - "If noone can forge Alice's signature. . .
- is a fulcrum to move the good information flows.
  - . . . then this message must be from Alice :)))"

Guiding principles for the next part

Guiding principles for the next part

- **Every secret must be authenticated**
  - to prevent impersonation.
  - Most protocol failures are authentication failures .

Guiding principles for the next part

- **Every secret must be authenticated**
  - to prevent impersonation.
  - Most protocol failures are authentication failures .
- **Every authentication must be based on a secret**
  - (in cyberspace).
  - The chicken and the egg.

Guiding principles for the next part

- **Every secret must be authenticated**
  - to prevent impersonation.
  - Most protocol failures are authentication failures .
- **Every authentication must be based on a secret**
  - (in cyberspace).
  - The chicken and the egg.
- **Security is always bootstrapped**
  - secrecy and authenticity are based on each other
  - new secrets are derived from old secrets