Security Science (SecSci) in string diagrams

Dusko Pavlovic
University of Hawaii

NIST
March 2018

Outline

Objective: Teaching security

Background: Geometry of computation

Approach: Geometry of security

Summary

Outline

Objective: Teaching security

Background: Geometry of computation

Approach: Geometry of security

Summary

SecSci (Security Science) Area

Objective and approach

- Teach security: tools
- Tools have a limited lifetime.
- Teach how to learn: science.

SecSci (Security Science) Area

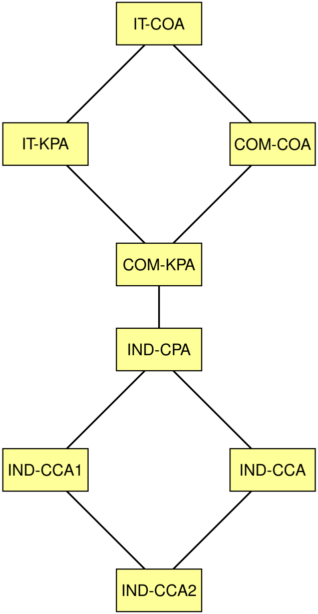
Objective and approach

- Teach security: tools
- Tools have a limited lifetime.
- Teach how to learn: science.

SecSci Science Problems

- Science whose subject does not like to be observed
- Science of complexity
- Complexity of science.

SecSci Concept: Secrecy



SecSci Concept: Crypto system

Definition

Given the types

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

SecSci Concept: Crypto system

Definition

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

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

When confusion seems unlikely, we abbreviate

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

SecSci Concept: 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, "perfect security"):

$$\Pr\left(m \leftarrow \mathcal{M} \mid c \leftarrow \mathbb{E}(\mathbb{K}, m)\right) = \Pr\left(m \leftarrow \mathcal{M}\right) \qquad \text{(IT-COA)}$$

SecSci Concept: 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:

$$\Pr\left(m \leftarrow \mathrm{A}(c) \mid c \leftarrow \mathbb{E}(\mathbb{K}, m)\right) = \Pr\left(m \leftarrow \mathrm{A}(0)\right) \quad \text{(COM-COA)}$$

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

SecSci Concept: 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:

$$\Pr\left(b \leftarrow 2 \,\middle|\, c \leftarrow \mathbb{E}(\mathbb{K}, m_b), m_0, m_1 \leftarrow \mathcal{M}\right) =$$
$$\Pr\left(b \leftarrow 2 \,\middle|\, m_0, m_1 \leftarrow \mathcal{M}\right) = \frac{1}{2} \quad \text{(IT-KPA)}$$

SecSci Concept: 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:

$$\Pr\left(b \leftarrow \mathrm{A}(m_0, m_1, c) \,\middle|\, c \leftarrow \mathbb{E}(m_b), m_0, m_1 \leftarrow \mathcal{M}\right) \leq$$
$$\Pr\left(b \leftarrow \mathrm{A}(m_0, m_1, 0) \,\middle|\, m_0, m_1 \leftarrow \mathcal{M}\right) \leq \frac{1}{2}$$

(COM-KPA)

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

SecSci Concept: Crypto system

Definition

. . . that together provide

- unique decryption:

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

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

$$\Pr\Big(b \leftarrow A(m_0, m_1, c) \,\Big|\, c \leftarrow \mathbb{E}(m_b), m_0, m_1 \leftarrow A_0\Big) \leq \frac{1}{2}$$

(IND-CPA)

for any probabilistic algorithm $A = \langle A_0, A_1 \rangle \ldots$

SecSci Concept: 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 (under chosen cyphertext attack):

$$\Pr\left(b \leftarrow \mathrm{A}_2(c_0, m, m_0, m_1, c) \;\middle|\; \begin{array}{r} c_0 \leftarrow \mathrm{A}_0 \\ m \leftarrow \mathrm{D}(c_0) \\ m_0, m_1 \leftarrow \mathrm{A}_1(c_0, m) \\ c \leftarrow \mathbb{E}(m_b) \end{array}\right) \leq \frac{1}{2} \qquad \text{(IND-CCA)}$$

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

SecSci Concept: 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 (under *adaptive* chosen cyphertext attack):

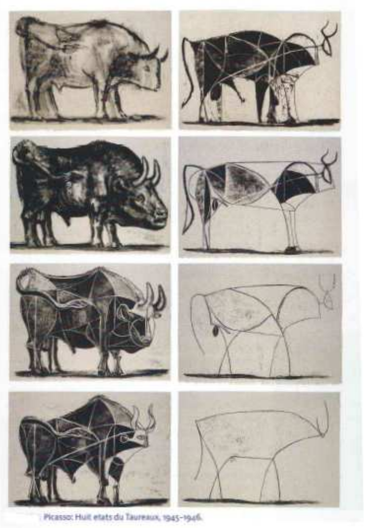
$$\Pr\left(b \leftarrow \mathsf{A}_3\left(\begin{array}{l} c_0, m,\ m_0, m_1, c, \\ c_1, \widetilde{m} \end{array} \right) \;\middle|\; \begin{array}{r} c_0 \leftarrow \mathsf{A}_0 \\ m \leftarrow \mathsf{D}(c_0) \\ m_0, m_1 \leftarrow \mathsf{A}_1(c_0, m) \\ c \leftarrow \mathbb{E}(m_b) \\ c_1 \leftarrow \mathsf{A}_2(c_0, m, m_0, m_1, c), \\ \widetilde{m} \leftarrow \mathsf{D}(c_1 \neq c) \end{array} \right) \leq \frac{1}{2}$$

(IND-CCA2)

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

Abstraction: Hide irrelevant structure

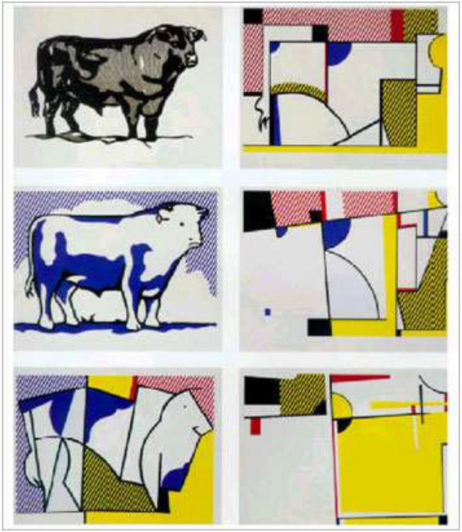
"Close the black box"



Pablo Picasso

Abstraction: Play with structure

"Paint the black box"



Roy Lichtenstein

Objective
Background
Approach
Summary

No abstraction: Drown in structure

"Open the black box"

Gunther von Hagens

Outline

Objective: Teaching security

Background: Geometry of computation

Approach: Geometry of security

Summary

Computer Science in 4 concepts

Lecture notes ($\subseteq$ textbook)

http://www.asecolab.org/courses/ics-222/

Concept 1: Induction

Counting generates numbers

$$
\begin{aligned}
1 &= \{🍎\} \\
2 &= \{🍎, 🍎\} \\
3 &= \{🍎, 🍎, 🍎\} \\
4 &= \{🍎, 🍎, 🍎, 🍎\}
\end{aligned}
$$

Concept 1: Induction

Counting begins from nothing

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{🍎\} \\
2 &= \{🍎, 🍎\} \\
3 &= \{🍎, 🍎, 🍎\} \\
4 &= \{🍎, 🍎, 🍎, 🍎\}
\end{aligned}
$$

Objective
Background
Approach
Summary

Concept 1: Induction

Counting boils down to nothing

$$\begin{aligned}
0 &= \{\} \\
1 &= \{\{\}\} \\
2 &= \{\{\}, \{\{\}\}\} \\
3 &= \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\} \\
4 &= \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}, \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\}\}
\end{aligned}$$

Concept 1: Induction

Counting boils down to nothing

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} \\
2 &= \{0, 1\} \\
3 &= \{0, 1, 2\} \\
4 &= \{0, 1, 2, 3\}
\end{aligned}
$$

Concept 1: Induction

Counting goes on on forever

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} \\
2 &= \{0, 1\} \\
3 &= \{0, 1, 2\} \\
4 &= \{0, 1, 2, 3\} \\
&\cdots \\
1 + n &= \{0, 1, 2, 3, \ldots, n\} \\
&\cdots
\end{aligned}
$$

Concept 1: Induction

Finite specifications of infinite processes

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} \\
2 &= \{0, 1\} \\
3 &= \{0, 1, 2\} \\
4 &= \{0, 1, 2, 3\} \\
&\cdots \\
\sigma(n) &= n \cup \{n\} \\
&\cdots
\end{aligned}
$$

Concept 1: Induction

Inductive definition

Natural numbers are generated by the schema

$$\frac{0 \in \mathbb{N} \qquad\qquad \mathbb{N} \xrightarrow{\sigma} \mathbb{N}}{n = \underbrace{\sigma \circ \sigma \circ \cdots \circ \sigma}_{n \text{ times}}(0) \in \mathbb{N}}$$

Concept 1: Induction

The Induction Schema

The *induction schema* is the derivation

$$\frac{b \in B \qquad\qquad B \xrightarrow{\beta} B}{\mathbb{N} \xrightarrow{f} B}$$

where B is an arbitrary type, $b \in B$ its element, β a given function, and f is defined by the following equations

$$f(0) = b \qquad\qquad f(1+n) = \beta\big(f(n)\big)$$

Concept 1: Induction

The Induction Schema

The *induction schema* is the derivation

$$\frac{b \in B \qquad\qquad B \xrightarrow{\beta} B}{\mathbb{N} \xrightarrow{(\!|b,\beta|\!)} B}$$

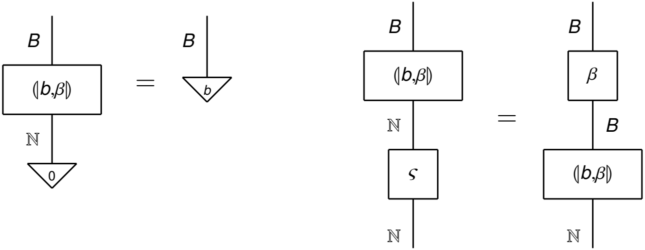
where B is an arbitrary type, $b \in B$ its element, β a given function, and f is defined by the following equations

$$(\!|b,\beta|\!)(0) = b \qquad\qquad (\!|b,\beta|\!) \circ \sigma(n) = \beta \circ (\!|b,\beta|\!)(n)$$

Concept 1: Induction

Induction in pictures

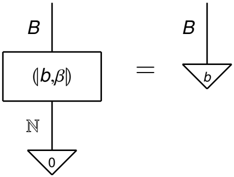
$$(\!|b,\beta|\!)(0) \;=\; b \qquad\qquad (\!|b,\beta|\!)\circ\varsigma(n) \;=\; \beta\circ(\!|b,\beta|\!)(n)$$

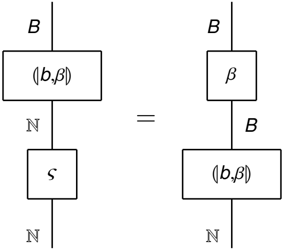


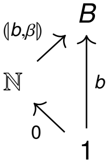
Concept 1: Induction

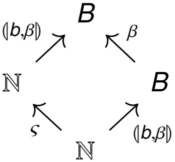
Induction in pictures

$$(\!|b,\beta|\!)(0) \quad = \quad b \qquad\qquad (\!|b,\beta|\!) \circ \varsigma(n) \quad = \quad \beta \circ (\!|b,\beta|\!)(n)$$









Concept 2: Coinduction

The Coinduction Schema

The *coinduction schema* is the derivation

$$\cfrac{s \in Q \qquad \cfrac{Q \times A \xrightarrow{\chi_0} Q}{Q \times A^* \xrightarrow{\chi_0^*} Q} \qquad Q \times A \xrightarrow{\chi_1} B}{A^+ \xrightarrow{[\![\chi]\!]_s} B}$$

where A, B are arbitrary types, Q is the state space, s is an initial state, χ is a given machine. Then the *behavior* of the process $\langle s, \chi \rangle$ is defined by

$$[\![\chi]\!]_s\left(\vec{y}{::}x\right) = \chi_1\left(\chi_0^*\left(s, \vec{y}\right), x\right) \text{ where}$$

$$\chi_0^*(q, ()) = q$$

$$\chi_0^*\left(q, \vec{y}{::}x\right) = \chi_0\left(\chi_0^*\left(q, \vec{y}\right), x\right)$$

Concept 2: Coinduction

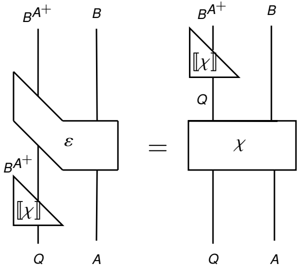
Proposition

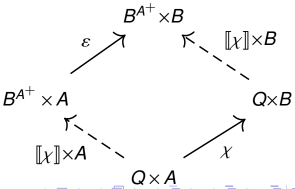
Every machine χ induces the *behavioral semantics* $[\![\chi]\!]$

$$\frac{Q \times A \xrightarrow{\chi} Q \times B}{Q \xrightarrow{[\![\chi]\!]} B^{A^+}}$$

such that

$$\begin{aligned}
[\![\chi]\!]_q^{::a} &= \varepsilon_0\left([\![\chi]\!]_q \,, a\right) \\
&= [\![\chi]\!]_{\chi_0(q,a)} \\
[\![\chi]\!]_q(a) &= \varepsilon_1\left([\![\chi]\!]_q \,, a\right) \\
&= \chi_1(q,a)
\end{aligned}$$





Concept 2: Coinduction

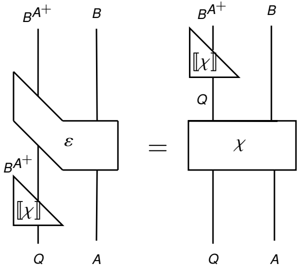
Proposition

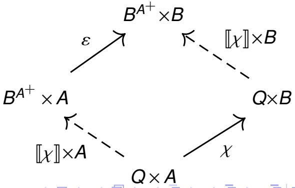
Every machine χ induces the *behavioral semantics* $[\![\chi]\!]$

$$\frac{Q \times A \stackrel{\chi}{\to} Q \times B}{Q \xrightarrow{[\![\chi]\!]} B^{A^+}}$$

such that

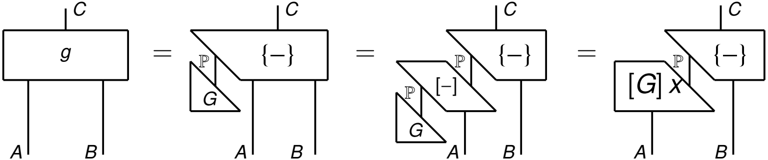
$$\begin{aligned}
[\![\chi]\!]_q\left(\vec{y}::a\right) &= \varepsilon_0\Big([\![\chi]\!]_q , a\Big)\left(\vec{y}\right) \\
&= [\![\chi]\!]_{\chi_0(q,a)}\left(\vec{y}\right) \\
[\![\chi]\!]_q\left(a\right) &= \varepsilon_1\Big([\![\chi]\!]_q , a\Big) \\
&= \chi_1(q,a)
\end{aligned}$$





Concept 3: Computer

Monoidal computer



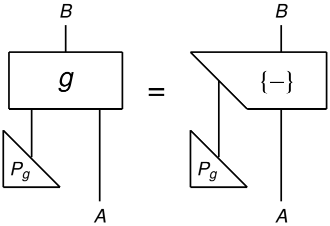
$$g(x, y) \quad = \quad \{G\}(x, y) \quad = \quad \{[G]\,x\}y$$

Concept 4: Completeness and complexity

Fundamental Theorem of Computation

For every computation $g : \mathbb{P} \times A \longrightarrow B$ there is a program $P_g \in \mathbb{P}$ such that

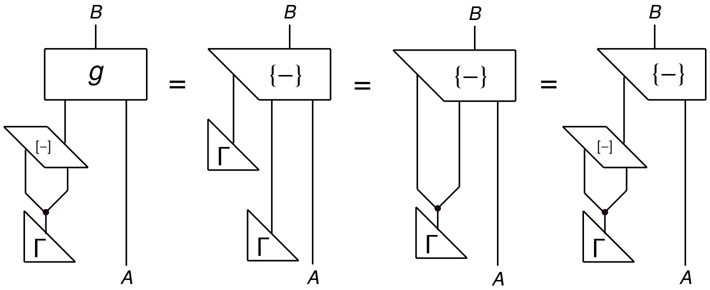
$$g\left(P_g, \vec{x}\right) = \left\{P_g\right\}\left(\vec{x}\right)$$



The program P_g is *Kleene's fixed point* of g.

Concept 4: Completeness and complexity

Proof



Outline

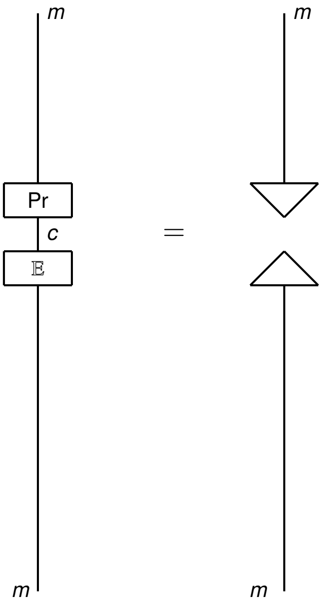
Objective: Teaching security

Background: Geometry of computation

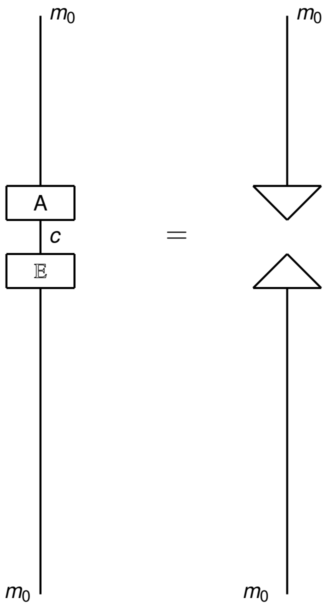
Approach: Geometry of security

Summary

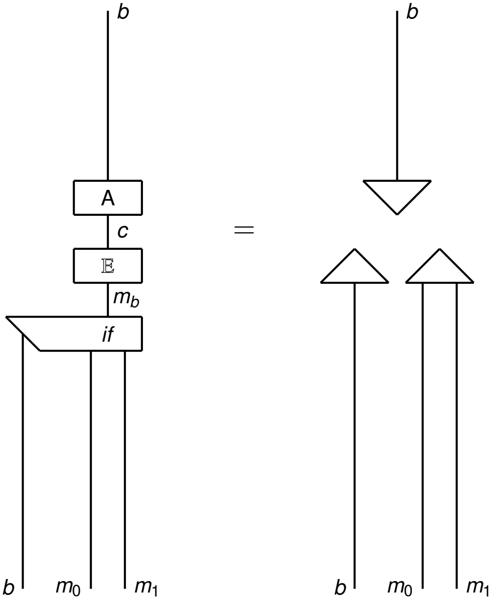
IT-COA (Shannon)



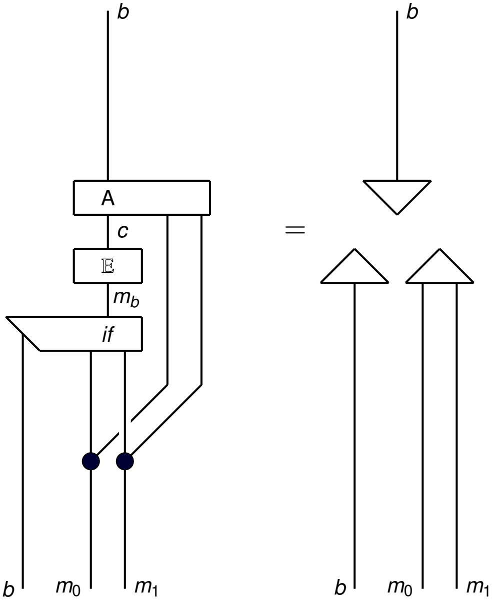
COM-COA



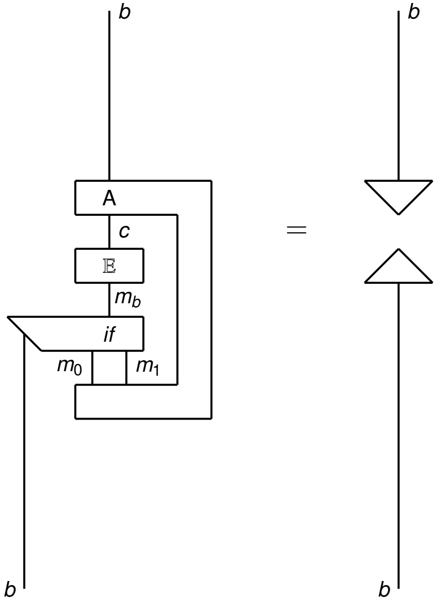
IND-COA



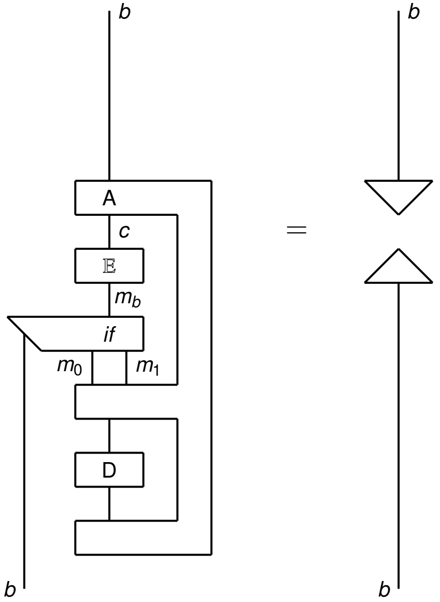
IND-KPA



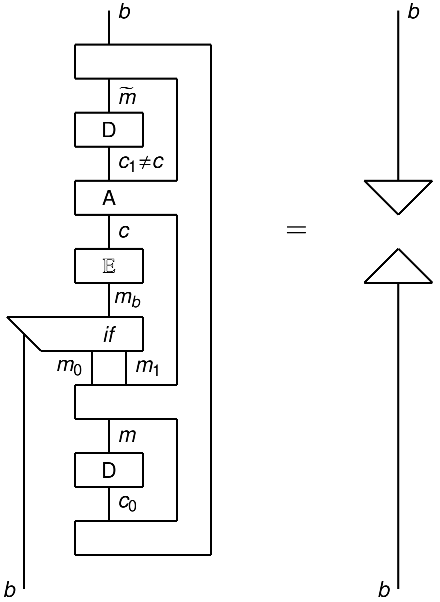
IND-CPA (Godwasser-Micali)



IND-CCA



IND-CCA2 (Luby-Rackoff)



Outline

Objective: Teaching security

Background: Geometry of computation

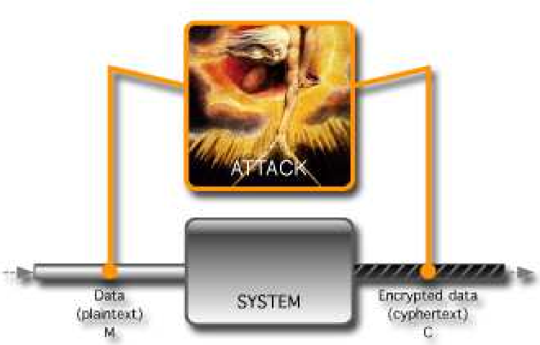
Approach: Geometry of security

Summary

Lecture notes ($\subseteq$ textbook)

http://www.asecolab.org/courses/ics-222/

Shannon's attacker: computationally unbounded
(omnipotent computer)



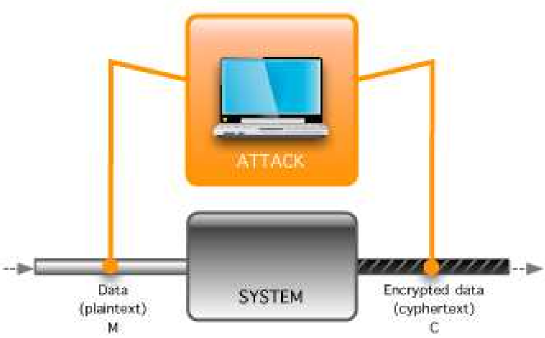
If a source contains some information,
then the attack will extract that iFnformation.

Shannon's attacker: computationally unbounded

(omnipotent computer)

$$\mathsf{Adv}_{\mathsf{E}}^{Sh} \;=\; \int_{m\leftarrow \mathsf{M}} \Pr\left(m\leftarrow \mathsf{M} \mid c = \mathsf{E}(m)\right) - \Pr\left(m\leftarrow \mathsf{M}\right)$$

Diffie-Hellman's attacker: computationally bounded
(real computer)



If attacker's computers have limited powers,
then information can be hard to extract.

Diffie-Hellman's attacker: computationally bounded

(real computer)

$$\mathrm{Adv}_{\mathsf{E}}^{DH}(\mathsf{A}) \;=\; \Pr\big(m \leftarrow \mathsf{A}(c) \mid c = \mathsf{E}(m)\big) - \Pr\big(m \leftarrow \mathsf{A}(0)\big)$$

Diffie-Hellman's attacker: computationally bounded
(real computer)

$$\mathrm{Adv}_{\mathrm{E}}^{DH}(\mathrm{A}) = \Pr\left(m \leftarrow \mathrm{A}(c) \mid c = \mathrm{E}(m)\right) - \Pr\left(m \leftarrow \mathrm{A}(0)\right)$$

$$\mathrm{Adv}_{\mathrm{E}}^{DH} = \bigvee_{\mathrm{A} \in PPT} \mathrm{Adv}_{\mathrm{E}}^{DH}(\mathrm{A})$$

Diffie-Hellman's attacker: computationally bounded

(real computer)

Idea

$\mathrm{Adv}_{\mathsf{E}}^{DH} \sim 0$ iff E is a *one-way function*, i.e. for almost all m holds

$$\begin{aligned} \exists k.\ D\big(m, \mathsf{E}(m)\big) &\leq \mathcal{O}\big(\ell(m)^k\big) \\ \forall k.\ D\big(\mathsf{E}(m), m\big) &> \mathcal{O}\big(\ell(m)^k\big) \end{aligned}$$

where for ensembles a, b we define

$$D(a,b) = \bigwedge_{\{p\}(a)=b} time(p,a)$$

Adaptive attacker: computationally bounded
(real computer)

$$\mathrm{Adv}_s^{IND-CCA2}(\mathsf{A}) \quad = $$

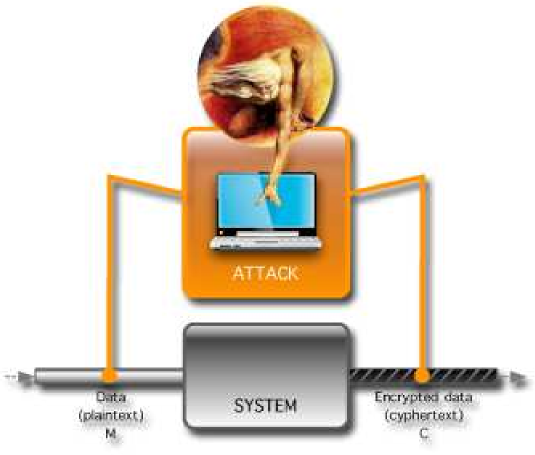
$$\Pr\Big(b \leftarrow \mathsf{A}_3\left({}^{\bullet}m,\ {}^{\bullet}c, \sigma_2, c_?, m_1, m_0, m^{\bullet}, c^{\bullet}\right) \ \Big| $$

$$\begin{array}{r} {}^{\bullet}m = \mathsf{D}_s\left(\overline{k},\ {}^{\bullet}c\right), \langle {}^{\bullet}c_{\neq c_?}, \sigma_2\rangle \leftarrow \mathsf{A}_2(c_?, m_1, m_0, \sigma_1, m^{\bullet}, c^{\bullet}) \\ c_? \leftarrow \mathsf{E}_s(k, m_b), b \leftarrow U_2, \langle m_1, m_0, \sigma_1\rangle \leftarrow \mathsf{A}_1(m^{\bullet}, c^{\bullet}, \sigma_0), \\ m^{\bullet} = \mathsf{D}_s(\overline{k}, c^{\bullet}), \langle c^{\bullet}, \sigma_0\rangle \leftarrow \mathsf{A}_0 \end{array} \Big)$$

$$- \quad \Pr\Big(b \leftarrow U_2 \Big)$$

Kerckhoffs' attacker: logically unbounded

(real computer, omnipotent programmer)



. . . but if there is a feasible attack algorithm,
then attacker's omnipotent programmers will find it.

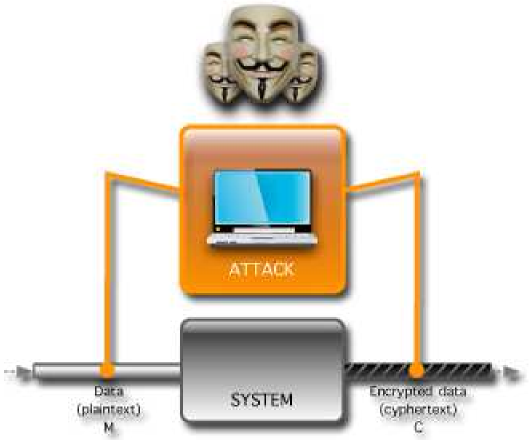
Kerckhoffs' attacker: logically unbounded

(real computer, omnipotent programmer)

$$\mathsf{Adv}_s^{IND-CCA2} \quad = \quad \bigvee_{A \in PPT} \Pr\Bigg(b \leftarrow A_3\left({}^{\bullet}m,\ {}^{\bullet}c, \sigma_2, c_?, m_1, m_0, m^{\bullet}, c^{\bullet}\right) \ \Bigg| \ {}^{\bullet}m = D_s\left(\overline{k},\ {}^{\bullet}c\right), \langle {}^{\bullet}c_{\neq c_?}, \sigma_2\rangle \leftarrow A_2\left(c_?, m_1, m_0, \sigma_1, m^{\bullet}, c^{\bullet}\right) \ c_? \leftarrow E_s(k, m_b), b \leftarrow U_2, \langle m_1, m_0, \sigma_1\rangle \leftarrow A_1\left(m^{\bullet}, c^{\bullet}, \sigma_0\right), \ m^{\bullet} = D_s(\overline{k}, c^{\bullet}), \langle c^{\bullet}, \sigma_0\rangle \leftarrow A_0 \Bigg) - \Pr\Big(b \leftarrow U_2\Big)$$

ASECO attacker: logically bounded

(real computer, **real** programmer)



If attacker's programmers have limited powers, then attack algorithms may be hard to find.

ASECO attacker: logically bounded

(real computer, **real** programmer)

$$\mathrm{Adv}_{s}^{IND-ASECO}(\mathbb{A}) \quad = \quad \bigvee_{a \leftarrow \mathbb{A}(s)} \Pr\left(b \leftarrow \{a_3\}\left({}^{\bullet}m,\ {}^{\bullet}c, c_?, m_1, m_0, m^{\bullet}, c^{\bullet}\right) \;\middle|\; \begin{array}{r} {}^{\bullet}m = \{d_s\}\left(\overline{k},\ {}^{\bullet}c\right), {}^{\bullet}c \leftarrow \{a_2\}\left(c_?, m_1, m_0, m^{\bullet}, c^{\bullet}\right) \\ c_? \leftarrow \{e_s\}(k, m_b), b \leftarrow U_2, \langle m_1, m_0\rangle \leftarrow \{a_1\}(m^{\bullet}, c^{\bullet}), \\ m^{\bullet} = \{d_s\}(\overline{k}, c^{\bullet}), c^{\bullet} \leftarrow \{a_0\} \end{array}\right) - \Pr\left(b \leftarrow U_2\right)$$

Adaptive security game

(both attacker and defender have real computers and real programmers)

$$\mathrm{Adv}^{IND-ASECO}(\mathbb{A}, \mathbb{S}) \quad = $$

$$\bigwedge_{s \leftarrow \mathbb{S}(a)} \quad \bigvee_{a \leftarrow \mathbb{A}(s)} \mathrm{Pr}\left(b \leftarrow \left\{a_3\right\}\left({}^{\bullet}m,\ {}^{\bullet}c, \ldots\right) \ \middle| \ \begin{array}{r} {}^{\bullet}m = \left\{d_s\right\}\left(\overline{k},\ {}^{\bullet}c\right), {}^{\bullet}c \leftarrow \left\{a_2\right\}\left(c_?, m_1, m_0, m^{\bullet}, c^{\bullet}\right) \\ c_? \leftarrow \left\{e_s\right\}(k, m_b), b \leftarrow U_2, \langle m_1, m_0\rangle \leftarrow \left\{a_1\right\}(m^{\bullet}, c^{\bullet}), \\ m^{\bullet} = \left\{d_s\right\}(\overline{k}, c^{\bullet}), c^{\bullet} \leftarrow \left\{a_0\right\} \end{array} \right)$$

$$- \quad \mathrm{Pr}\left(b \leftarrow U_2\right)$$

Adaptive security game

(both attacker and defender have real computers and real programmers)

Idea

$\mathsf{Adv}_{E}^{IND-ASECO}(\mathbb{A}, \mathbb{S}) \sim 0$ iff for $a \leftarrow \mathbb{A}(s)$ and $s \leftarrow \mathbb{S}(a)$ holds with overwhelming probability

$$\begin{array}{rcl} \exists k.\ D\big(a, s\big) & \leq & \mathcal{O}\big(\ell(a)^k\big) \\ \forall k.\ D\big(s, a\big) & > & \mathcal{O}\big(\ell(s)^k\big) \end{array}$$

where

$$D(a, b) = \bigwedge_{\{p\}(a)=b} time(p, a)$$

Adaptive security game

(both attacker and defender have real computers and real programmers)

Idea

$\mathsf{Adv}_{E}^{IND-ASECO}(\mathbb{A}, \mathbb{S}) \sim 0$ iff for $a \leftarrow \mathbb{A}(s)$ and $s \leftarrow \mathbb{S}(a)$ holds with overwhelming probability

$$\begin{array}{rcl} \exists k.\ D\big(a, s\big) & \leq & \mathcal{O}\Big(\ell(a)^k\Big) \\ \forall k.\ D\big(s, a\big) & > & \mathcal{O}\Big(\ell(s)^k\Big) \end{array}$$

where

$$D(a, b) \quad = \quad \bigwedge_{\{p\}(a)=b} time(p, a)$$

. . . with a couple of tweaks.

Summary: Beyond omnipotence

power	*unbounded*	*bounded*
rationality	Cournot	Simon
computational	Shannon	Diffie-Hellman
logical	Kerckhoffs	ASECO