AUTOMATIC PROOFS OF SECURITY BY CONTRADICTION

Hubert Comon (ENS Cachan)

GOALS OF THE LECTURE

Explain the “security by contradiction” idea and work it out on some instances.

REFERENCES

Forthcoming lecture notes ...

Papers by Bana et al.

Guillaume Scerri's thesis

SCARY
`http://www.lsv.ens-cachan.fr/~scerri/tool/`

SAFETY VS SECURITY

Formal methods

SAFETY VS SECURITY

Formal methods

- Safety: given P, ϕ,

$$P \quad \overset{?}{\models} \quad \phi$$

SAFETY VS SECURITY

Formal methods

- Safety: given P, ϕ,

$$P \quad \overset{?}{\models} \quad \phi$$

- Security: given P, ϕ,

$$\forall \mathcal{A}. \quad P \quad \overset{?}{\models}_{\mathcal{A}} \quad \phi$$

SAFETY VS SECURITY

Formal methods

- Safety: given P, ϕ,

$$P \quad \overset{?}{\models} \quad \phi$$

- Security (revised): given P, ϕ, $\mathcal{C}$,

$$\forall \mathcal{A} \in \mathcal{C}. \quad P \quad \overset{?}{\models}_{\mathcal{A}} \quad \phi$$

EXAMPLES OF ATTACKER'S MODELS

- Dolev-Yao model (used for instance in PROVERIF, TAMARIN,...)
- Interactive Probabilistic Polynomial Time Turing machines (PPT, used for instance in EASYCRYPT, CRYPTOVERIF)
- side channels ...

THE SCOPE OF SECURITY PROOFS (I)

Security by iterating attacks/fixes: an endless race.

THE SCOPE OF SECURITY PROOFS (I)

Security by iterating attacks/fixes: an endless race.

Solution: formally prove the security

THE SCOPE OF SECURITY PROOFS (I)

Security by iterating attacks/fixes: an endless race.

Solution: formally prove the security

A paradoxical situation:

- 1978: NS protocol; a proof of security
- 1995: an attack, a fix (now NSL) and a proof of security
- 2003: an attack, a fix and a proof of security
- 2011: an attack, a fix, a proof of security

THE SCOPE OF SECURITY PROOFS (I)

Security by iterating attacks/fixes: an endless race.

Solution: formally prove the security

A paradoxical situation:

- 1978: NS protocol; a proof of security
- 1995: an attack, a fix (now NSL) and a proof of security
- 2003: an attack, a fix and a proof of security
- 2011: an attack, a fix, a proof of security

The proofs are nevertheless correct ...

THE SCOPE OF SECURITY PROOFS (II)

The security proofs consider a *fixed* class of attackers $\mathcal{C}$.

Considering a larger class of attackers, there might be attacks on a proved program.

THE SCOPE OF SECURITY PROOFS (II)

The security proofs consider a *fixed* class of attackers $\mathcal{C}$.

Considering a larger class of attackers, there might be attacks on a proved program.

Implicit assumptions on external libraries.

SECURITY BY CONTRADICTION (I)

Formal assumptions on the attacker class/ the external libraries: Axioms

+

Proof that the program satisfies the security property, assuming Axioms

SECURITY BY CONTRADICTION (II)

From P, ϕ compute ϕ_P such that

$$P \models_{\mathcal{A}} \phi \qquad \text{iff} \qquad \models_{\mathcal{A}} \phi_P$$

Consider $\mathcal{C}$ as the class of models of Axioms.

$$\forall \mathcal{A} \in \mathcal{C}. \quad P \models_{\mathcal{A}} \phi$$

iff

$$\text{Axioms} \models \phi_P$$

iff

$$\text{Axioms} \cup \neg \phi_P \models \bot$$

SECURIY BY CONTRADICTION (III)

- Design Axioms
- Prove that (Axioms + ¬security + P) is inconsistent
- Conclusion: P is secure w.r.t. all attackers/libraries that satisfy Axioms

- If (Axioms + ¬security + P) was consistent: there is a model, witnessing an attack

ANOTHER FORMULATION

Dolev-Yao, PPT : classes $\mathcal{C}$ specified by the attacker's capabilities (what the attacker can do)

$\mathcal{C}$ as the models of Axioms: specify what the attacker cannot do.

Consistency: there are (library implementations + attacker) violating the security property, while only performing authorized actions.

Inconsistency: any attacker/library satisfying Axioms cannot break the security

ADVANTAGES OF THE APPROACH

- recursive FO spec. : automation
- all assumptions are explicit
- minimal spec.
- modularity
- if Axioms are computationally sound, then we get security in the computational model

EXAMPLE

$$A \to B : \quad \mathsf{enc}(< A, n_A >, \mathsf{pk}_B, r)$$
$$B \to A : \quad \mathsf{enc}(n_A, \mathsf{pk}_A, r')$$

Security goal: agreement on n_A.

EXAMPLE

$$A \to B : \quad \mathsf{enc}(< A, n_A >, \mathsf{pk}_B, r)$$
$$B \to A : \quad \mathsf{enc}(n_A, \mathsf{pk}_A, r')$$

Security goal: agreement on n_A.

$$P_A(a,b) : \ \nu n_a, r. \quad \mathtt{out}(\mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r));\ \mathtt{in}(x)$$
$$\text{if } \mathsf{dec}(x, \mathsf{sk}_a) = n_a \text{ then } \mathtt{out}(OK)$$

EXAMPLE

$A \rightarrow B : \quad \mathsf{enc}(< A, n_A >, \mathsf{pk}_B, r)$

$B \rightarrow A : \quad \mathsf{enc}(n_A, \mathsf{pk}_A, r')$

Security goal: agreement on n_A.

$P_A(a,b) : \nu n_a, r. \quad \mathtt{out}(\mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r));\ \mathtt{in}(x)$

$\quad$ if $\mathsf{dec}(x, \mathsf{sk}_a) = n_a$ then $\mathtt{out}(OK)$

$P_B(b) : \nu r'. \quad \mathtt{in}(y)$. let $x_a = \pi_1(\mathsf{dec}(y, \mathsf{sk}_b))$ in

$\quad$ let $y_{n_a} = \mathsf{dec}(y, \mathsf{sk}_b)$ in $\mathtt{out}(\mathsf{enc}(y_{n_a}, \mathsf{pk}_{x_a}, r'))$

EXAMPLE

$$A \to B : \quad \mathsf{enc}(< A, n_A >, \mathsf{pk}_B, r)$$

$$B \to A : \quad \mathsf{enc}(n_A, \mathsf{pk}_A, r')$$

Security goal: agreement on n_A.

$$P_A(a,b) : \nu n_a, r. \quad \mathtt{out}(\mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r));\ \mathtt{in}(x)$$
$$\text{if } \mathsf{dec}(x, \mathsf{sk}_a) = n_a \text{ then } \mathtt{out}(OK)$$

$$P_B(b) : \nu r'. \quad \mathtt{in}(y).\ \text{let } x_a = \pi_1(\mathsf{dec}(y, \mathsf{sk}_b)) \text{ in}$$
$$\text{let } y_{n_a} = \mathsf{dec}(y, \mathsf{sk}_b) \text{ in } \mathtt{out}(\mathsf{enc}(y_{n_a}, \mathsf{pk}_{x_a}, r'))$$

Security: when OK is emitted, $y_{n_a} = n_a$

Write on the board.

EXAMPLE (II)

An execution trace (out of), interacting with an (arbitrary) active attacker:

$$\texttt{out}P_A;\ \texttt{in}P_B\ \texttt{out}P_B;\ \texttt{in}P_A;\ \texttt{out}P_A$$

EXAMPLE (II)

An execution trace (out of 10), interacting with an (arbitrary) active attacker:

$$\texttt{out}P_A;\ \texttt{in}P_B\ \texttt{out}P_B;\ \texttt{in}P_A;\ \texttt{out}P_A$$

EXAMPLE (II)

An execution trace (out of 10), interacting with an (arbitrary) active attacker:

$$\mathtt{out}P_A;\ \mathtt{in}P_B\ \mathtt{out}P_B;\ \mathtt{in}P_A;\ \mathtt{out}P_A$$

$\phi_0 = a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

EXAMPLE (II)

An execution trace (out of 10), interacting with an (arbitrary) active attacker:

$$\texttt{out}P_A;\ \texttt{in}P_B\ \texttt{out}P_B;\ \texttt{in}P_A;\ \texttt{out}P_A$$

$\phi_0 = a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

$y = f_y(\phi_0)$, $x_a = \pi_1(\mathsf{dec}(y, \mathsf{sk}_b))$, $y_{n_a} = \mathsf{dec}(y, \mathsf{sk}_b)$

EXAMPLE (II)

An execution trace (out of 10), interacting with an (arbitrary) active attacker:

$$\texttt{out}P_A;\ \texttt{in}P_B\ \texttt{out}P_B;\ \texttt{in}P_A;\ \texttt{out}P_A$$

$\phi_0 = a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

$y = f_y(\phi_0)$, $x_a = \pi_1(\mathsf{dec}(y, \mathsf{sk}_b))$, $y_{n_a} = \mathsf{dec}(y, \mathsf{sk}_b)$

$\phi_1 = \phi_0, \mathsf{enc}(< b, y_{n_a} >, \mathsf{pk}_{x_a}, r')$

EXAMPLE (II)

An execution trace (out of 10), interacting with an (arbitrary) active attacker:

$$\mathtt{out}P_A;\ \mathtt{in}P_B\ \mathtt{out}P_B;\ \mathtt{in}P_A;\ \mathtt{out}P_A$$

$\phi_0 = a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

$y = f_y(\phi_0)$, $x_a = \pi_1(\mathsf{dec}(y, \mathsf{sk}_b))$, $y_{n_a} = \mathsf{dec}(y, \mathsf{sk}_b)$

$\phi_1 = \phi_0, \mathsf{enc}(< b, y_{n_a} >, \mathsf{pk}_{x_a}, r')$

$x = f_x(\phi_1)$

EXAMPLE (II)

An execution trace (out of 10), interacting with an (arbitrary) active attacker:

$$\texttt{out}P_A;\ \texttt{in}P_B\ \texttt{out}P_B;\ \texttt{in}P_A;\ \texttt{out}P_A$$

$\phi_0 = a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

$y = f_y(\phi_0)$, $x_a = \pi_1(\mathsf{dec}(y, \mathsf{sk}_b))$, $y_{n_a} = \mathsf{dec}(y, \mathsf{sk}_b)$

$\phi_1 = \phi_0, \mathsf{enc}(< b, y_{n_a} >, \mathsf{pk}_{x_a}, r')$

$x = f_x(\phi_1)$

$\mathsf{dec}(x, \mathsf{sk}_a) = n_a$

EXAMPLE (III)

Recap of the security property: forall f_y, f_x,

$$\mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) = n_a \quad \Rightarrow \quad y_{n_a} = n_a$$

where

$$\phi_1 \equiv \phi_0, \mathsf{enc}(< b, y_{n_a} >, \mathsf{pk}_{x_a}, r)$$

$$y_{n_a} \equiv \mathsf{dec}(f_y(\phi_0), \mathsf{sk}_b)$$

$$\phi_0 \equiv a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$$

(IN)CONSISTENCY FORMULATION

Prove that the following is inconsistent (in FOL !!)

$$\mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) = n_a \quad \wedge \quad y_{n_a} \neq n_a \wedge \text{Axioms}$$

where

$\phi_1 \equiv \phi_0, \mathsf{enc}(< b, y_{n_a} >, \mathsf{pk}_{x_a}, r)$

$y_{n_a} \equiv \pi_2(\mathsf{dec}(f_y(\phi_0), \mathsf{sk}_b))$

$\phi_0 \equiv a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

Write on the board (long term)

If there is no axiom ...

AXIOMS: ATTACKER'S RESTRICTIONS

$\triangleleft$: a predicate whose intended meaning is:

$S \triangleleft t$ if the attacker can compute t from S.

AXIOMS: ATTACKER'S RESTRICTIONS

$\lhd$: a predicate whose intended meaning is:

$S \lhd t$ if the attacker can compute t from S.

Let us try (for all S not containing sk_z...):

$$\forall x, y, z, S. \quad S, \mathsf{enc}(x, \mathsf{pk}_z, r) \lhd \mathsf{enc}(y, \mathsf{pk}_z, r') \quad \Rightarrow \quad S \lhd \mathsf{enc}(y, \mathsf{pk}_z, r') \vee x = y$$

$$\forall \vec{x}. \quad \vec{x} \lhd f_y(\vec{x}) \qquad \forall \vec{x}. \quad \vec{x} \lhd f_x(\vec{x})$$

$$\forall x, y, z. \quad \mathsf{dec}(\mathsf{enc}(x, \mathsf{pk}_y, z), \mathsf{sk}_y) = x \qquad \forall x_1, x_2. \quad \pi_i(< x_1, x_2 >) = x_i$$

AXIOMS: ATTACKER'S RESTRICTION

For all S not containing sk_z as plaintext ...

$$\forall x, y, z, S. \quad S, \mathsf{enc}(x, \mathsf{pk}_z, r) \lhd \mathsf{enc}(y, \mathsf{pk}_z, r') \quad \Rightarrow \quad S \lhd \mathsf{enc}(y, \mathsf{pk}_z, r') \vee x = y$$

$$\forall \vec{x}. \quad \vec{x} \lhd f_y(\vec{x}) \qquad \forall \vec{x}. \quad \vec{x} \lhd f_x(\vec{x})$$

$$\forall x, y, z. \quad \mathsf{dec}(\mathsf{enc}(x, \mathsf{pk}_y, z), \mathsf{sk}_y) = x \qquad \forall x_1, x_2. \quad \pi_i(< x_1, x_2 >) = x_i$$

AXIOMS: ATTACKER'S RESTRICTION

For all S not containing sk_z as plaintext ...

$$\forall x, y, z, S. \quad S, \mathsf{enc}(x, \mathsf{pk}_z, r) \lhd \mathsf{enc}(y, \mathsf{pk}_z, r') \quad \Rightarrow \quad S \lhd \mathsf{enc}(y, \mathsf{pk}_z, r') \vee x = y$$

$$\forall \vec{x}. \quad \vec{x} \lhd f_y(\vec{x}) \qquad \forall \vec{x}. \quad \vec{x} \lhd f_x(\vec{x})$$

$$\forall x, y, z. \quad \mathsf{dec}(\mathsf{enc}(x, \mathsf{pk}_y, z), \mathsf{sk}_y) = x \qquad \forall x_1, x_2. \quad \pi_i(< x_1, x_2 >) = x_i$$

There is model (i.e., an attack):

f_x is computing something, which is *not* a ciphertext, but that can be decrypted to n_a.

f_y is computing something, which is *not* a ciphertext, but that can be decrypted to a pair $< n_a, v >$ with $v \neq n_a$

AXIOMS: ATTACKER'S RESTRICTIONS

Let us try again: for S not containing sk as plaintext, and n only under encryption with pk,

$$\forall x, y, z. \quad S \lhd x \wedge S, \mathsf{dec}(x, \mathsf{sk}) \lhd n \quad \Rightarrow \bigvee_{\mathsf{enc}(u, \mathsf{pk}, r) \in S} x = \mathsf{enc}(u, \mathsf{pk}, r)$$

$$\forall \vec{x}. \quad \vec{x} \lhd f_y(\vec{x}) \qquad \forall \vec{x}. \quad \vec{x} \lhd f_x(\vec{x})$$

$$\forall x, y, z. \quad \mathsf{dec}(\mathsf{enc}(x, \mathsf{pk}_y, z), \mathsf{sk}_y) = x \qquad \forall x_1, x_2. \pi_i(< x_1, x_2 >) = x_i$$

AXIOMS: ATTACKER'S RESTRICTIONS

Let us try again: for S not containing sk as plaintext, and n only under encryption with pk,

$$\forall x, y, z. \quad S \lhd x \wedge S, \mathsf{dec}(x, \mathsf{sk}) \lhd n \quad \Rightarrow \quad \bigvee_{\mathsf{enc}(u, \mathsf{pk}, r) \in S} x = \mathsf{enc}(u, \mathsf{pk}, r)$$

$$\forall \vec{x}. \quad \vec{x} \lhd f_y(\vec{x}) \qquad \forall \vec{x}. \quad \vec{x} \lhd f_x(\vec{x})$$

$$\forall x, y, z. \quad \mathsf{dec}(\mathsf{enc}(x, \mathsf{pk}_y, z), \mathsf{sk}_y) = x \qquad \forall x_1, x_2. \pi_i(< x_1, x_2 >) = x_i$$

From $\mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) = n_a$ we derive $\phi_1, \mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) \lhd n_a$.).

AXIOMS: ATTACKER'S RESTRICTIONS

Let us try again: for S not containing sk as plaintext, and n only under encryption with pk,

$$\forall x, y, z. \quad S \lhd x \wedge S, \mathsf{dec}(x, \mathsf{sk}) \lhd n \quad \Rightarrow \bigvee_{\mathsf{enc}(u,\mathsf{pk},r) \in S} x = \mathsf{enc}(u, \mathsf{pk}, r)$$

$$\forall \vec{x}. \quad \vec{x} \lhd f_y(\vec{x}) \qquad \forall \vec{x}. \quad \vec{x} \lhd f_x(\vec{x})$$

$$\forall x, y, z. \quad \mathsf{dec}(\mathsf{enc}(x, \mathsf{pk}_y, z), \mathsf{sk}_y) = x \qquad \forall x_1, x_2 . \pi_i(< x_1, x_2 >) = x_i$$

From $\mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) = n_a$ we derive $\phi_1, \mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) \lhd n_a$.).

From $\phi_1 \lhd f_x(\phi_1)$ and $\phi_1, \mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) \lhd n_a$, we derive $f_x(\phi_1) = \mathsf{enc}(y_{n_a}, \mathsf{pk}_{x_a}, r_2) \wedge \mathsf{pk}_{x_a} = \mathsf{pk}_a$.

AXIOMS: ATTACKER'S RESTRICTIONS

Let us try again: for S not containing sk as plaintext, and n only under encryption with pk,

$$\forall x, y, z. \quad S \lhd x \wedge S, \mathsf{dec}(x, \mathsf{sk}) \lhd n \quad \Rightarrow \quad \bigvee_{\mathsf{enc}(u,\mathsf{pk},r) \in S} x = \mathsf{enc}(u, \mathsf{pk}, r)$$

$$\forall \vec{x}. \quad \vec{x} \lhd f_y(\vec{x}) \qquad \forall \vec{x}. \quad \vec{x} \lhd f_x(\vec{x})$$

$$\forall x, y, z. \quad \mathsf{dec}(\mathsf{enc}(x, \mathsf{pk}_y, z), \mathsf{sk}_y) = x \qquad \forall x_1, x_2. \pi_i(< x_1, x_2 >) = x_i$$

From $\mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) = n_a$ we derive $\phi_1, \mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) \lhd n_a$.).

From $\phi_1 \lhd f_x(\phi_1)$ and $\phi_1, \mathsf{dec}(f_x(\phi_1), \mathsf{sk}_a) \lhd n_a$, we derive $f_x(\phi_1) = \mathsf{enc}(y_{n_a}, \mathsf{pk}_{x_a}, r_2) \wedge \mathsf{pk}_{x_a} = \mathsf{pk}_a$.

Which yields a contradiction using the properties of pairs/encryption.

Remark: the computability of f_y is not necessary.

CONCLUSION

For any encryption scheme and attacker's model, in which

Integrity:

$$\forall x, y, z. \quad S \lhd\ x \wedge S, \mathsf{dec}(x, \mathsf{sk}) \lhd\ n \quad \Rightarrow \quad \bigvee_{\mathsf{enc}(u,\mathsf{pk},r) \in S} x = \mathsf{enc}(u, \mathsf{pk}, r)$$

is satisfied (in some restricted contexts S), then the protocol is secure.

ANOTHER EXAMPLE

$$A \to B : \quad \mathsf{enc}(< A, n_A >, \mathsf{pk}_B, r)$$
$$B \to A : \quad n_A$$

Is it secure (agreement on n_A), for all attackers/implementations that satisfy the properties that we gave ?

SOLUTION

Is the following satisfiable ?

$$\pi_2(\mathsf{dec}(f_y(\phi_0), \mathsf{sk}_b)) \neq n_a \wedge \text{Axioms} \quad \wedge \quad f_x(\phi_1) = n_a$$

where

$\phi_1 \equiv \phi_0, y_{n_a}$

$y_{n_a} \equiv \pi_2(\mathsf{dec}(f_y(\phi_0), \mathsf{sk}_b))$

$\phi_0 \equiv a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

SOLUTION

Is the following satisfiable ?

$$\pi_2(\mathsf{dec}(f_y(\phi_0), \mathsf{sk}_b)) \neq n_a \wedge \text{Axioms} \quad \wedge \quad f_x(\phi_1) = n_a$$

where

$\phi_1 \equiv \phi_0, y_{n_a}$

$y_{n_a} \equiv \pi_2(\mathsf{dec}(f_y(\phi_0), \mathsf{sk}_b))$

$\phi_0 \equiv a, b, \mathsf{pk}_a, \mathsf{pk}_b, \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

It is unsatisfiable:

$\phi_0, \mathsf{dec}(f_y(\phi_0), \mathsf{sk}_b) \lhd \; n_a$

$\phi_0 \lhd \; f_y(\phi_0)$

$f_y(\phi_0) = \mathsf{enc}(< a, n_a >, \mathsf{pk}_b, r)$

$y_{n_a} = n_a$.

The new protocol is also secure, with the same assumptions.