Verification of Security Protocols
Part II

Véronique Cortier[1]

September, 2010

Fosad 2010

[1]LORIA, CNRS

Advertisement

ProSecure project

Goal : analysis and design of security systems

→ five years project (2011-2015), founded by the European Research Council.

→ Regular job offers !

- PhD positions and Post-doc positions
- One research associate position (up to 5 years, with budget for PhD grant and other costs)
- Permanent positions (CNRS, INRIA, Universities)

→ contact me cortier@loria.fr

LORIA (Nancy)

Size : 500 researchers, among which about 150 permanent researchers and 150 PhD students.

Where is it ?

Well connected to :

- Paris, France (90 minutes)
- Luxembourg (90-120 minutes)
- Saarbrucken, Germany (120 minutes)

Yesterday course

How to use formal methods for analysing cryptographic protocols ?

- Messages are abstracted by terms
- Intruder can compute new terms using a deduction system
- Protocols can be described by rules of the form $u \rightarrow v$, where u, v are terms with variables.

What formal methods allow to do ?

- In general, secrecy preservation is undecidable.

What formal methods allow to do ?

- In general, secrecy preservation is undecidable.
- For a bounded number of sessions, secrecy is co-NP-complete [RusinowitchTuruani CSFW01]
  $\longrightarrow$ several tools for detecting attacks (Casper, Avispa platform...)

What formal methods allow to do ?

- In general, secrecy preservation is undecidable.
- For a bounded number of sessions, secrecy is co-NP-complete [RusinowitchTuruani CSFW01]
  $\longrightarrow$ several tools for detecting attacks (Casper, Avispa platform...)
- For an unbounded number of sessions
  - for one-copy protocols, secrecy is DEXPTIME-complete [CortierComon RTA03] [SeildVerma LPAR04]
  - for message-length bounded protocols, secrecy is DEXPTIME-complete [Durgin et al FMSP99] [Chevalier et al CSL03]

  $\longrightarrow$ some tools for proving security (ProVerif, EVA Platform)

Limitations of this approach ?

Are you ready to use any protocol verified with this technique ?

- Only a finite scenario is checked.
  $\longrightarrow$ What happens if the protocol is used one more time ?
- The underlying mathematical properties of the primitives are abstracted away.
- The specification of the protocol is analysed, but not its implementation. $\longrightarrow$ Cédric Fournet course

Motivation

Back to our running example :

$$\begin{array}{rcl} A \rightarrow B & : & \{\mathrm{pin}\}_{k_a} \\ B \rightarrow A & : & \{\{\mathrm{pin}\}_{k_a}\}_{k_b} \\ A \rightarrow B & : & \{\mathrm{pin}\}_{k_b} \end{array}$$

We need the equation for the commutativity of encryption

$$\{\{z\}_x\}_y = \{\{z\}_y\}_x$$

Some other examples

Encryption-Decryption theory

$$\mathrm{dec}(\mathrm{enc}(x, y), y) = x \quad \pi_1(\langle x, y\rangle) = x \quad \pi_2(\langle x, y\rangle) = y$$

EXclusive Or

$$\begin{array}{rclcrcl} x \oplus (y \oplus z) & = & z & \quad & x \oplus y & = & y \oplus x \\ x \oplus x & = & 0 & \quad & x \oplus 0 & = & x \end{array}$$

Diffie-Hellmann

$$\exp(\exp(z, x), y) = \exp(\exp(z, y), x)$$

E-voting protocols

First phase :

$$V \rightarrow A : \quad \mathrm{sign}(\mathit{blind}(\mathrm{vote}, r), V)$$
$$A \rightarrow V : \quad \mathrm{sign}(\mathit{blind}(\mathrm{vote}, r), A)$$

Voting phase :

$$V \rightarrow C : \quad \mathrm{sign}(\mathrm{vote}, A)$$

...

Equational theory for blind signatures

[Kremer Ryan 05]

$$\begin{array}{rcl} \mathrm{checksign}(\mathrm{sign}(x, y), \mathrm{pk}(y)) & = & x \\ \mathrm{unblind}(\mathrm{blind}(x, y), y) & = & x \\ \mathrm{unblind}(\mathrm{sign}(\mathrm{blind}(x, y), z), y) & = & \mathrm{sign}(x, z) \end{array}$$

Deduction

$$\frac{}{T \vdash_E M}\ M \in T \qquad\qquad \frac{T \vdash_E M_1 \quad \cdots \quad T \vdash_E M_k}{T \vdash_E f(M_1, \ldots, M_k)}\ f \in \Sigma$$

$$\frac{T \vdash M}{T \vdash M'}\ M =_E M'$$

Deduction

$$\frac{}{T \vdash_E M}\ M \in T \qquad \frac{T \vdash_E M_1 \quad \cdots \quad T \vdash_E M_k}{T \vdash_E f(M_1, \ldots, M_k)}\ f \in \Sigma$$

$$\frac{T \vdash M}{T \vdash M'}\ M =_E M'$$

Example : $E := \mathrm{dec}(\mathrm{enc}(x, y), y) = x$ and $T = \{\mathrm{enc}(secret, k),\ k\}$.

$$\frac{\dfrac{\dfrac{}{T \vdash \mathrm{enc}(secret, k)} \qquad \dfrac{}{T \vdash k}}{T \vdash \mathrm{dec}(\mathrm{enc}(secret, k), k)}\ f \in \Sigma}{T \vdash secret}\ \mathrm{dec}(\mathrm{enc}(x, y), y) = x$$

Rewriting system

For analyzing equational theories, we (try to) associate to E a finite convergent rewriting system $\mathcal{R}$ such that :

$$u =_E v \quad \text{iff} \quad u\downarrow = v\downarrow$$

Definition (Characterization of the deduction relation)

Let $t_1, \dots t_n$ and u be terms in normal form.

$$\{t_1, \dots t_n\} \vdash u \quad \text{iff} \quad \exists C \text{ s.t. } C[t_1, \dots, t_n] \to^* u$$

(Also called Cap Intruder problem [Narendran et al])

Some results with equational theories

	Security problem	
	Bounded number of sessions	**Unbounded number of sessions**
Commutative encryption	*co-NP-complete* [CKRT04]	**Ping-pong protocols :** *co-NP-complete* [Turuani04]
Exclusive Or	*Decidable* [CS03,CKRT03]	**One copy - No nonces :** *Decidable* [CLC03] **Two-way automata - No nonces :** *Decidable* [Verma03]
Abelian Groups	*Decidable* [Shmatikov04]	**Two-way automata - No nonces :** *Decidable* [Verma03]
Prefix encryption	*co-NP-complete* [CKRT03]	
Abelian Groups and Modular Exponentiation	**General case :** *Decidable* [Shmatikov04] **Restricted protocols :** *co-NP-complete* [CKRT03]	**AC properties of the Modular Exponentiation No nonces :** *Semi-Decision Procedure* [GLRV04]

And now are you ready to use any protocol verified with these techniques ?

Assuming :

- Analysis for an unbounded number of sessions
- With equational theories

Outline of the talk

Towards more cryptographic guarantees

1. Formal methods for protocols
   - Yesterday course
   - Adding equational theories
2. Cryptographic models
   - Encryption schemes
   - Security of encryption
   - Cryptographic models
   - Linking formal and cryptographic models
3. Passive case
   - Setting
   - Patterns
   - Soundness of indistinguishability
4. Active case
   - Setting
   - Trace mapping
   - A special case : computational secrecy
   - General computational indistinguishability

Specificity of cryptographic models

- Messages are bitstrings
- Real encryption algorithm
- Real signature algorithm
- General and powerful adversary

$\longrightarrow$ very little abstract model

Encryption : the old time

- Caesar encryption : $A \rightarrow E$, $B \rightarrow F$, $C \rightarrow G$, ...
- Cypher Disk (Léone Battista Alberti 1466)

Encryption : the old time

- Caesar encryption : $A \rightarrow E$, $B \rightarrow F$, $C \rightarrow G$, ...
- Cypher Disk (Léone Battista Alberti 1466)

$\rightarrow$ subject to statistical analysis (Analyzing letter frequencies)

Encryption : mechanized time

Automatic substitutions and permutations

Enigma

Encryption nowadays

$\rightarrow$ Based on algorithmically hard problems.

RSA Function $n = pq$, p et q primes.

e : public exponent

- $x \mapsto x^e \mod n$ easy (cubic)
- $y = x^e \mapsto x \mod n$ difficult
  $x = y^d$ où $d = e^{-1} \mod \phi(n)$

Encryption nowadays

$\rightarrow$ Based on algorithmically hard problems.

RSA Function $n = pq$, p et q primes.

e : public exponent

- $x \mapsto x^e \mod n$ easy (cubic)
- $y = x^e \mapsto x \mod n$ difficult
  $x = y^d$ où $d = e^{-1} \mod \phi(n)$

Diffie-Hellman Problem

- Given $A = g^a$ and $B = g^b$,
- Compute $\mathrm{DH}(A, B) = g^{ab}$

Encryption nowadays

$\rightarrow$ Based on algorithmically hard problems.

RSA Function $n = pq$, p et q primes.

e : public exponent

- $x \mapsto x^e \mod n$ easy (cubic)
- $y = x^e \mapsto x \mod n$ difficult
  $x = y^d$ où $d = e^{-1} \mod \phi(n)$

Diffie-Hellman Problem

- Given $A = g^a$ and $B = g^b$,
- Compute $\mathrm{DH}(A, B) = g^{ab}$

$\rightarrow$ Based on hardness of integer factorization.

Estimations for integer factorization

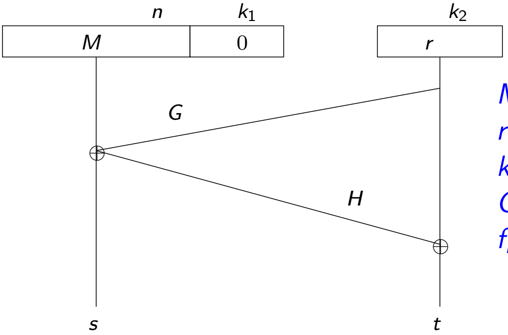
Module (bits)	Operations (in $\log_2$)
512	58
1024	80
2048	111
4096	149
8192	156

$\approx 2^{60}$ years

$\longrightarrow$ Lower bound for RSA and Diffie-Hellman.

How does an (asymmetric) encryption algorithm look like ?

Example : OAEP [Bellare Rogaway]



M : plaintext of length n
r : randomness of length
k_0
G, H : hash function
f_k : trapdoor function

$$E_K(x; r) = f_K(s||t)$$

What is a secure encryption scheme ?

What is a secure encryption scheme ?

Intuitively :

- An adversary
- should not know the underlying plaintext.

Security of asymmetric encryption

Public data :

- $c = E_{k_e}(m, r)$ cyphertext
- k_e encryption key

There exists a unique message m satisfying the relation (with possible several relevant r)
$\rightarrow$ An exhaustive search on m and r yields m !

Security of asymmetric encryption

Public data :

- $c = E_{k_e}(m, r)$ cyphertext
- k_e encryption key

There exists a unique message m satisfying the relation (with possible several relevant r)
$\rightarrow$ An exhaustive search on m and r yields m !

$\Rightarrow$ Unconditional secrecy is impossible, one has to rely on algorithmic assumptions.

How to define an attacker/adversary

We wish to model an attacker :

- as clever as possible
  $\rightarrow$ he/she should be able to perform any operation
- with a limited time.
  - E.g. we do not wish to consider attacks that require 2^{60} years
  - Otherwise, the adversary could enumerate all keys (exponential time in $2^{size(keys)}$)

How to define an attacker/adversary

We wish to model an attacker :

- as clever as possible
  $\rightarrow$ he/she should be able to perform any operation
- with a limited time.
  - E.g. we do not wish to consider attacks that require 2^{60} years
  - Otherwise, the adversary could enumerate all keys (exponential time in $2^{size(keys)}$)

Model : we consider any Turing machine

- that models any algorithm
- probabilistic : The adversary can generate keys and chose randomly his behavior
- polynomial in the size of the keys : which represents a reasonable execution time.

Security proof in a nutshell

Proof by reduction

1. Hypothesis : The algorithmic problem P is hard = there is no polynomial algorithm ($P = RSA, DL, DDH, CDH$...)

Security proof in a nutshell

Proof by reduction

1. Hypothesis : The algorithmic problem P is hard = there is no polynomial algorithm ($P = RSA, DL, DDH, CDH \ldots$)
2. Reduction :
   - If there exists a (polynomial) adversary A un adversaire (polynomial) breaking the encryption scheme,
   - Then one can build upon A for solving P in polynomial time.

Security proof in a nutshell

Proof by reduction

1. Hypothesis : The algorithmic problem P is hard = there is no polynomial algorithm ($P = RSA, DL, DDH, CDH \ldots$)
2. Reduction :
   - If there exists a (polynomial) adversary A un adversaire (polynomial) breaking the encryption scheme,
   - Then one can build upon A for solving P in polynomial time.
3. Conclusion : the encryption scheme is secure, there is no polynomial adversary.

What is a secure encryption scheme ?

- An adversary
- should not know the underlying plaintext.

$\longrightarrow$ several possible definitions of knowledge

One-Wayness (OW)

Basic security property : One-Wayness (OW)
without the inverse key, one cannot retrieve the underlying plaintext :

$$\Pr_{m,r}[c = E(m; r) \mid \mathcal{A}(c) = m]$$

is negligible.

One-Wayness (OW)

Basic security property : One-Wayness (OW)
without the inverse key, one cannot retrieve the underlying plaintext :

$$Pr_{m,r}[c = E(m;r) \mid \mathcal{A}(c) = m]$$

is negligible.

Negligibility : f is negligible if for any polynomial p, there exists η_0 s.t. for all $\eta \geq \eta_0$

$$f(\eta) \leq 1/p(\eta)$$

Not strong enough !

- The adversary may be able to compute half of the secret message.
- There is no guarantee in case that some partial information on the secret is known.

Not strong enough !

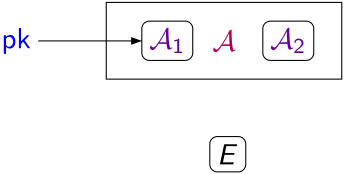
- The adversary may be able to compute half of the secret message.
- There is no guarantee in case that some partial information on the secret is known.

$\longrightarrow$ Introduction of a notion of indistinguishability. :
The adversary shall not guess even one bit of the underlying plaintext.

Indistinguabilité (IND)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

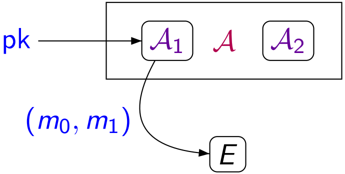
1. the adversary $\mathcal{A}_1$ is given the public key pk.



Indistinguabilité (IND)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

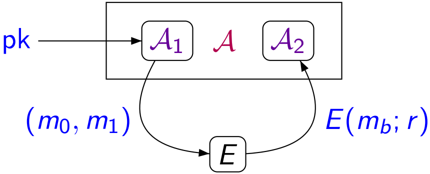
1. the adversary $\mathcal{A}_1$ is given the public key pk.
2. The adversary $\mathcal{A}_1$ chooses two messages m_0, m_1.



Indistinguabilité (IND)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

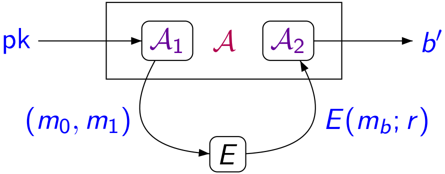
1. the adversary $\mathcal{A}_1$ is given the public key pk.
2. The adversary $\mathcal{A}_1$ chooses two messages m_0, m_1.
3. one bit $b = 0, 1$ is flipped and $c = E(m_b; r)$ is given to the adversary.



Indistinguabilité (IND)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

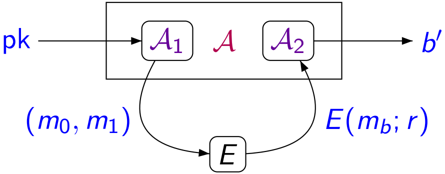
1. the adversary $\mathcal{A}_1$ is given the public key pk.
2. The adversary $\mathcal{A}_1$ chooses two messages m_0, m_1.
3. one bit $b = 0, 1$ is flipped and $c = E(m_b; r)$ is given to the adversary.
4. The adversary $\mathcal{A}_2$ outputs b'.



Indistinguabilité (IND)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

1. the adversary $\mathcal{A}_1$ is given the public key pk.
2. The adversary $\mathcal{A}_1$ chooses two messages m_0, m_1.
3. one bit $b = 0, 1$ is flipped and $c = E(m_b; r)$ is given to the adversary.
4. The adversary $\mathcal{A}_2$ outputs b'.



The probability $\Pr[b = b'] - \frac{1}{2}$ should be negligible.

Even stronger !

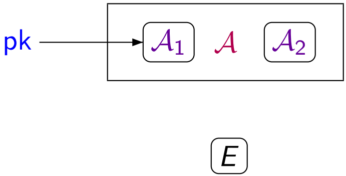
Non Malleability (NM)

Given a cyphertext $E(m; r)$, the adversary should not be able to create a cyphertext $E(m'; r')$ such that messages m and m' have a meaningful relation.

Definition of Non Malleability (NM)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

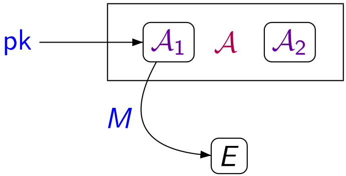
1. The adversary $\mathcal{A}_1$ is given the public key pk.



Definition of Non Malleability (NM)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

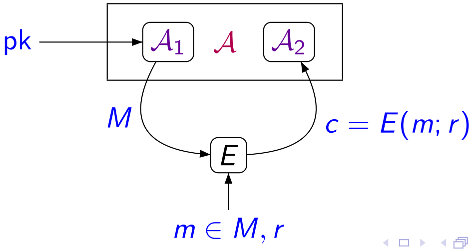
1. The adversary $\mathcal{A}_1$ is given the public key pk.
2. The adversary $\mathcal{A}_1$ chooses a set of messages M.



Definition of Non Malleability (NM)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

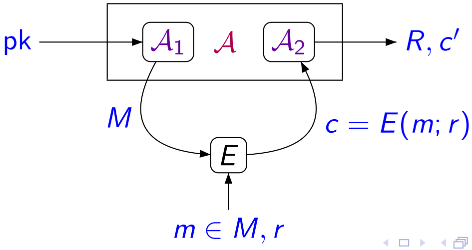
1. The adversary $\mathcal{A}_1$ is given the public key pk.
2. The adversary $\mathcal{A}_1$ chooses a set of messages M.
3. Two messages m and m^* are chosen at random in M and $c = E(m; r)$ is given to the adversary.



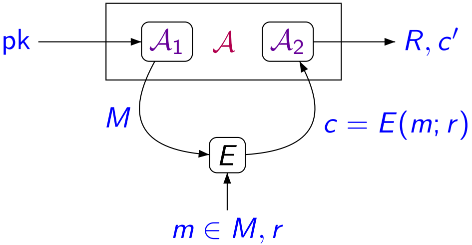
Definition of Non Malleability (NM)

Game Adversary : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

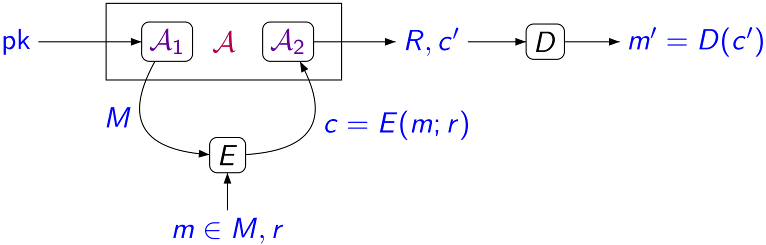
1. The adversary $\mathcal{A}_1$ is given the public key pk.
2. The adversary $\mathcal{A}_1$ chooses a set of messages M.
3. Two messages m and m^* are chosen at random in M and $c = E(m; r)$ is given to the adversary.
4. The adversary $\mathcal{A}_2$ outputs a binary relation R and a cyphertext c'.



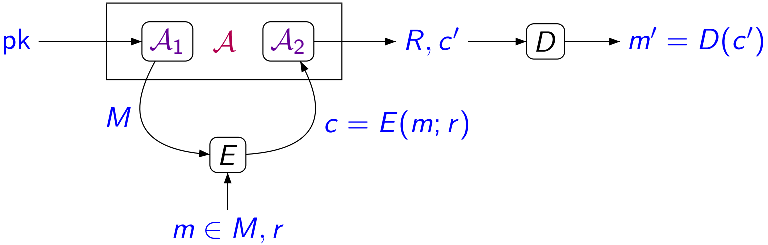
Definition of Non Malleability (NM)



Definition of Non Malleability (NM)



Definition of Non Malleability (NM)



The probability $\Pr[R(m, m')] - \Pr[R(m, m^*)]$ should be negligible.

Relations

Non Malleability

$\Downarrow$

Indistinguishability

$\Downarrow$

One-Wayness

Exercise (medium) : show the implications.

Adding even more security

The adversary has access to oracles :

→ Encryption of all messages of his choice
→ Decryption of all messages of his choice

Three standard levels of security :

- Chosen-Plaintext Attacks (CPA)

Adding even more security

The adversary has access to oracles :

→ Encryption of all messages of his choice
→ Decryption of all messages of his choice

Three standard levels of security :

- Chosen-Plaintext Attacks (CPA)
- Non adaptive Chosen-Ciphertext Attacks (CCA1)
  → access to the (decryption) oracle before the challenge.

Adding even more security

The adversary has access to oracles :

→ Encryption of all messages of his choice
→ Decryption of all messages of his choice

Three standard levels of security :

- Chosen-Plaintext Attacks (CPA)
- Non adaptive Chosen-Ciphertext Attacks (CCA1)
  → access to the (decryption) oracle before the challenge.
- Adaptive Chosen-Ciphertext Attacks (CCA2)
  → unlimited access to the (decryption) oracle (except for the challenge)

Relations

$$\begin{tikzcd}
\text{NM-CPA} \arrow[d] & \text{NM-CCA1} \arrow[l] \arrow[d] & \text{NM-CCA2} \arrow[l] \arrow[d, leftrightarrow] \\
\text{IND-CPA} \arrow[d] & \text{IND-CCA1} \arrow[l] & \text{IND-CCA2} \arrow[l] \\
\text{OW-CPA} & &
\end{tikzcd}$$

Outline of the talk

Towards more cryptographic guarantees

1. Formal methods for protocols
   - Yesterday course
   - Adding equational theories
2. Cryptographic models
   - Encryption schemes
   - Security of encryption
   - Cryptographic models
   - Linking formal and cryptographic models
3. Passive case
   - Setting
   - Patterns
   - Soundness of indistinguishability
4. Active case
   - Setting
   - Trace mapping
   - A special case : computational secrecy
   - General computational indistinguishability

Cryptographic models

Encryption is only one component of cryptographic models

- Cryptographic primitives : encryption, signatures, ...
- Protocol model
- Adversary
- Security notions

Setting for cryptographic protocols

Protocol :

- Message exchange program
- using cryptographic primitives

Adversary $\mathcal{A}$: any probabilistic polynomial Turing machine, *i.e.* any probabilistic polynomial program.

- polynomial : captures what is feasible
- probabilistic : the adversary may try to guess some information

Definition of secrecy preservation

$\rightarrow$ Several notions of secrecy :

One-Wayness : The probability for an adversary $\mathcal{A}$ to compute the secret s against a protocol $\mathcal{P}$ is negligible (smaller than any inverse of polynomial).

$$\forall p \text{ polynomial } \quad \exists \eta_0 \ \forall \eta \geq \eta_0 \quad \Pr^{\eta}_{m,r}[\mathcal{A}(\mathcal{P}_K) = s] \leq \frac{1}{p(\eta)}$$

η : security parameter = key length

Definition of secrecy preservation

$\rightarrow$ Several notions of secrecy :

One-Wayness : The probability for an adversary $\mathcal{A}$ to compute the secret s against a protocol $\mathcal{P}$ is negligible (smaller than any inverse of polynomial).

$$\forall p \text{ polynomial } \exists \eta_0\ \forall \eta \geq \eta_0 \quad \Pr^{\eta}_{m,r}[\mathcal{A}(\mathcal{P}_K) = s] \leq \frac{1}{p(\eta)}$$

η : security parameter $=$ key length

$\rightarrow$ Not enough ! (why ?)

Computational secrecy

Computational secrecy of s is defined through the following game :

- Two values n_0 and n_1 are randomly generated instead of s ;
- The adversary interacts with the protocol where s is replaced by n_b, $b \in \{0, 1\}$;
- We give the pair (n_0, n_1) to the adversary ;
- The adversary gives b',

The data s is secret if $\Pr[b = b'] - \frac{1}{2}$ is a negligible function.

A typical cryptographic proof

1. Assume that some algorithmic problem P is difficult (E.g. RSA or integer factorization or Discrete Log or CDH, DDH, ...)
2. Suppose that a (polynomial probabilistic) adversary $\mathcal{A}$ breaks the protocol security with non negligible probability

A typical cryptographic proof

1. Assume that some algorithmic problem P is difficult (E.g. RSA or integer factorization or Discrete Log or CDH, DDH, ...)
2. Suppose that a (polynomial probabilistic) adversary $\mathcal{A}$ breaks the protocol security with non negligible probability
3. Build out of $\mathcal{A}$ an adversary $\mathcal{B}$ that solves P.

A typical cryptographic proof

1. Assume that some algorithmic problem P is difficult (E.g. RSA or integer factorization or Discrete Log or CDH, DDH, ...)
2. Suppose that a (polynomial probabilistic) adversary $\mathcal{A}$ breaks the protocol security with non negligible probability
3. Build out of $\mathcal{A}$ an adversary $\mathcal{B}$ that solves P.
4. Conclude that the protocol is secure provided P is difficult.

Outline of the talk

Towards more cryptographic guarantees

1. Formal methods for protocols
   - Yesterday course
   - Adding equational theories
2. Cryptographic models
   - Encryption schemes
   - Security of encryption
   - Cryptographic models
   - Linking formal and cryptographic models
3. Passive case
   - Setting
   - Patterns
   - Soundness of indistinguishability
4. Active case
   - Setting
   - Trace mapping
   - A special case : computational secrecy
   - General computational indistinguishability

Formal and Cryptographic approaches

	Formal approach	Cryptographic approach
Messages	terms	bitstrings
Encryption	idealized	algorithm
Adversary	idealized	any polynomial algorithm
Secrecy property	reachability-based property	indistinguishability
Guarantees	unclear	strong
Protocol	may be complex	usually simpler

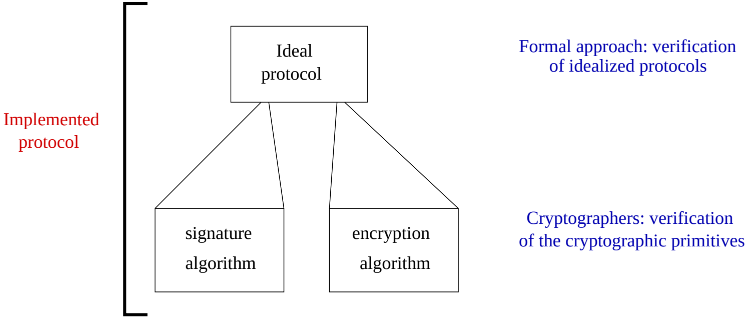
Formal and Cryptographic approaches

	Formal approach	Cryptographic approach
Messages	terms	bitstrings
Encryption	idealized	algorithm
Adversary	idealized	any polynomial algorithm
Secrecy property	reachability-based property	indistinguishability
Guarantees	unclear	strong
Protocol	may be complex	usually simpler
Proof	automatic	by hand, tedious and error-prone

Link between the two approaches ?

Composition of the two approaches

Automatic cryptographically sound proofs



Passive Case

A first result : seminal result from M. Abadi and Ph. Rogaway

J. of Cryptology, 2002

How to symbolically abstract computational indistinguishability of distributions ?

Setting

- Messages are represented by terms

$$
\begin{array}{lllll}
T & ::= & & & \text{terms} \\
 & | & x & & \text{variable} \\
 & | & a & & \text{name/nonce} \\
 & | & f(T_1, \ldots, T_k) & & \text{application of symbol } f \in \mathcal{F}
\end{array}
$$

In the initial result of Abadi and Rogaway, $\mathcal{F} = \{\mathsf{enc}, \langle\ ,\rangle\}$

- Each functional symbol has a concrete implementation
$\Rightarrow$ a sequence of messages

$$n,\ \mathsf{enc}(n, k),\ \mathsf{enc}(\langle n\ , n\rangle, k)$$

generates a distribution : uniform distribution for nonces and application of the functions (symmetric encryption and pairing).

Distinguishing distributions

The two distributions $[\![\psi]\!]$ and $[\![\psi']\!]$ are indistinguishable, $[\![\psi]\!] \approx [\![\psi']\!]$, if

$$\mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi]\!]; \mathcal{A}(\eta, \widehat{\psi}) = 1\right] - \mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi']\!]; A(\eta, \widehat{\psi}) = 1\right]$$

is a negligible function of η.

Distinguishing distributions

The two distributions $[\![\psi]\!]$ and $[\![\psi']\!]$ are indistinguishable, $[\![\psi]\!] \approx [\![\psi']\!]$, if

$$\mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi]\!]; \mathcal{A}(\eta, \widehat{\psi}) = 1\right] - \mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi']\!]; \mathcal{A}(\eta, \widehat{\psi}) = 1\right]$$

is a negligible function of η.

Examples

$$\phi_1 = n_0, n_1, \mathrm{enc}(n_0, k) \qquad \phi_2 = n_0, n_1, \mathrm{enc}(n_1, k)$$

Distinguishing distributions

The two distributions $[\![\psi]\!]$ and $[\![\psi']\!]$ are indistinguishable, $[\![\psi]\!] \approx [\![\psi']\!]$, if

$$\mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi]\!]; \mathcal{A}(\eta, \widehat{\psi}) = 1\right] - \mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi']\!]; A(\eta, \widehat{\psi}) = 1\right]$$

is a negligible function of η.

Examples

$$\phi_1 = n_0, n_1, \mathsf{enc}(n_0, k) \approx \phi_2 = n_0, n_1, \mathsf{enc}(n_1, k)$$

$$\phi_3 = n_0, n_1, \mathsf{enc}(n_0, k), k \qquad \phi_4 = n_0, n_1, \mathsf{enc}(n_1, k), k$$

Distinguishing distributions

The two distributions $[\![\psi]\!]$ and $[\![\psi']\!]$ are indistinguishable, $[\![\psi]\!] \approx [\![\psi']\!]$, if

$$\mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi]\!]; \mathcal{A}(\eta, \widehat{\psi}) = 1\right] - \mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi']\!]; \mathcal{A}(\eta, \widehat{\psi}) = 1\right]$$

is a negligible function of η.

Examples

$$\phi_1 = n_0, n_1, \mathrm{enc}(n_0, k) \approx \phi_2 = n_0, n_1, \mathrm{enc}(n_1, k)$$

$$\phi_3 = n_0, n_1, \mathrm{enc}(n_0, k), k \not\approx \phi_4 = n_0, n_1, \mathrm{enc}(n_1, k), k$$

$$\phi_5 = \mathrm{enc}(n_0, k), k \qquad \phi_6 = \mathrm{enc}(n_0, k'), k$$

Distinguishing distributions

The two distributions $[\![\psi]\!]$ and $[\![\psi']\!]$ are indistinguishable, $[\![\psi]\!] \approx [\![\psi']\!]$, if

$$\mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi]\!]; \mathcal{A}(\eta, \widehat{\psi}) = 1\right] - \mathbb{P}\left[\widehat{\psi} \leftarrow [\![\psi']\!]; A(\eta, \widehat{\psi}) = 1\right]$$

is a negligible function of η.

Examples

$$\phi_1 = n_0, n_1, \mathsf{enc}(n_0, k) \approx \phi_2 = n_0, n_1, \mathsf{enc}(n_1, k)$$

$$\phi_3 = n_0, n_1, \mathsf{enc}(n_0, k), k \not\approx \phi_4 = n_0, n_1, \mathsf{enc}(n_1, k), k$$

$$\phi_5 = \mathsf{enc}(n_0, k), k \not\approx \phi_6 = \mathsf{enc}(n_0, k'), k$$

Patterns : Definition of what is visible to an intruder

Given a sequence $S = M_1, M_2, \ldots, M_k$, we define

$$\mathsf{Pat}(S) = \{\mathsf{Pat}^S(M_1), \mathsf{Pat}^S(M_2), \ldots, \mathsf{Pat}^S(M_k)\} \text{ with}$$

Patterns : Definition of what is visible to an intruder

Given a sequence $S = M_1, M_2, \ldots, M_k$, we define

$$\mathrm{Pat}(S) = \{\mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2), \ldots, \mathrm{Pat}^S(M_k)\} \text{ with}$$

$$\mathrm{Pat}^S(a) = \begin{cases} a & \text{if } S \vdash a \\ \square & \text{otherwise} \end{cases}$$

Patterns : Definition of what is visible to an intruder

Given a sequence $S = M_1, M_2, \ldots, M_k$, we define

$$\mathrm{Pat}(S) = \{\mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2), \ldots, \mathrm{Pat}^S(M_k)\} \text{ with}$$

$$\begin{aligned}
\mathrm{Pat}^S(a) &= \begin{cases} a & \text{if } S \vdash a \\ \square & \text{otherwise} \end{cases} \\
\mathrm{Pat}^S(\langle M_1, M_2 \rangle) &= \langle \mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2) \rangle
\end{aligned}$$

Patterns : Definition of what is visible to an intruder

Given a sequence $S = M_1, M_2, \ldots, M_k$, we define

$$\mathrm{Pat}(S) = \{\mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2), \ldots, \mathrm{Pat}^S(M_k)\} \text{ with}$$

$$\mathrm{Pat}^S(a) = \begin{cases} a & \text{if } S \vdash a \\ \square & \text{otherwise} \end{cases}$$

$$\mathrm{Pat}^S(\langle M_1, M_2 \rangle) = \langle \mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2) \rangle$$

$$\mathrm{Pat}^S(\{M\}_k) = \begin{cases} \{\mathrm{Pat}^S(M)\}_k & \text{if } S \vdash k \\ \square & \text{otherwise} \end{cases}$$

Reminder : deduction system

Standard "Dolev Yao" deduction system, seen Part I of this course.

$$\frac{T \vdash u \quad T \vdash v}{T \vdash \langle u\,,v\rangle} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathrm{enc}(u,v)}$$

$$\frac{}{T \vdash u}\; u \in T \qquad \frac{T \vdash \langle u\,,v\rangle}{T \vdash u} \qquad \frac{T \vdash \langle u\,,v\rangle}{T \vdash v}$$

$$\frac{T \vdash \mathrm{enc}(u,v) \quad T \vdash v}{T \vdash u}$$

Examples

$$\phi_1 = \begin{array}{c} n_0, n_1, \mathrm{enc}(n_0, k) \\ n_0, n_1, \square \end{array} \approx \phi_2 = \begin{array}{c} n_0, n_1, \mathrm{enc}(n_1, k) \\ n_0, n_1, \square \end{array}$$

Examples

$$\phi_1 = \begin{array}{c} n_0, n_1, \mathrm{enc}(n_0, k) \\ n_0, n_1, \square \end{array} \approx \phi_2 = \begin{array}{c} n_0, n_1, \mathrm{enc}(n_1, k) \\ n_0, n_1, \square \end{array}$$

$$\phi_3 = \begin{array}{c} n_0, n_1, \mathrm{enc}(n_0, k), k \\ n_0, n_1, \mathrm{enc}(n_0, k), k \end{array} \not\approx \phi_4 = \begin{array}{c} n_0, n_1, \mathrm{enc}(n_1, k), k \\ n_0, n_1, \mathrm{enc}(n_1, k), k \end{array}$$

Examples

$$\begin{array}{rccrc}
\phi_1 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_0, k) \\ n_0, n_1, \square \end{array} & \approx & \phi_2 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_1, k) \\ n_0, n_1, \square \end{array} \\[2em]
\phi_3 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_0, k), k \\ n_0, n_1, \mathrm{enc}(n_0, k), k \end{array} & \not\approx & \phi_4 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_1, k), k \\ n_0, n_1, \mathrm{enc}(n_1, k), k \end{array} \\[2em]
\phi_5 = & \begin{array}{c} \mathrm{enc}(n_0, k), k \\ \mathrm{enc}(n_0, k), k \end{array} & \not\approx & \phi_6 = & \begin{array}{c} \mathrm{enc}(n_0, k'), k \\ \square, k \end{array}
\end{array}$$

Examples

$$\begin{array}{llcll}
\phi_1 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_0, k) \\ n_0, n_1, \square \end{array} & \approx & \phi_2 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_1, k) \\ n_0, n_1, \square \end{array} \\[2em]
\phi_3 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_0, k), k \\ n_0, n_1, \mathrm{enc}(n_0, k), k \end{array} & \not\approx & \phi_4 = & \begin{array}{c} n_0, n_1, \mathrm{enc}(n_1, k), k \\ n_0, n_1, \mathrm{enc}(n_1, k), k \end{array} \\[2em]
\phi_5 = & \begin{array}{c} \mathrm{enc}(n_0, k), k \\ \mathrm{enc}(n_0, k), k \end{array} & \not\approx & \phi_6 = & \begin{array}{c} \mathrm{enc}(n_0, k'), k \\ \square, k \end{array}
\end{array}$$

Definition

Two patterns are equivalent, denoted by $\equiv$ if they are equal up-to bijective renaming.

Soundness of indistinguishability

Theorem (Abadi-Rogaway)

Equivalence of patterns implies computational indistinguishability

$$\mathrm{Pat}(S_1) \equiv \mathrm{Pat}(S_2) \quad \Rightarrow \quad [\![S_1]\!] \approx [\![S_2]\!]$$

Soundness of indistinguishability

Theorem (Abadi-Rogaway)

Equivalence of patterns implies computational indistinguishability

$$\mathsf{Pat}(S_1) \equiv \mathsf{Pat}(S_2) \quad \Rightarrow \quad [\![S_1]\!] \approx [\![S_2]\!]$$

Provided that :

- Encryption is
  - IND-CPA

Soundness of indistinguishability

Theorem (Abadi-Rogaway)

Equivalence of patterns implies computational indistinguishability

$$\mathrm{Pat}(S_1) \equiv \mathrm{Pat}(S_2) \quad \Rightarrow \quad [\![S_1]\!] \approx [\![S_2]\!]$$

Provided that :

- Encryption is
  - IND-CPA
  - message length-concealing
    $\mathrm{Pat}(\mathrm{enc}(n,k)) = \square = \mathrm{Pat}(\mathrm{enc}(\langle n\ , n, n, n\rangle, k))$

Soundness of indistinguishability

Theorem (Abadi-Rogaway)

Equivalence of patterns implies computational indistinguishability

$$\mathrm{Pat}(S_1) \equiv \mathrm{Pat}(S_2) \quad \Rightarrow \quad [\![S_1]\!] \approx [\![S_2]\!]$$

Provided that :

- Encryption is
  - IND-CPA
  - message length-concealing
    $\mathrm{Pat}(\mathrm{enc}(n, k)) = \Box = \mathrm{Pat}(\mathrm{enc}(\langle n\ , n, n, n\rangle, k))$
  - which key-concealing

$$\mathrm{Pat}(\mathrm{enc}(n, k), \mathrm{enc}(n', k)) = \Box = \mathrm{Pat}(\mathrm{Pat}(\mathrm{enc}(n, k), \mathrm{enc}(n', k')))$$

Soundness of indistinguishability

Theorem (Abadi-Rogaway)

Equivalence of patterns implies computational indistinguishability

$$\mathrm{Pat}(S_1) \equiv \mathrm{Pat}(S_2) \quad \Rightarrow \quad [\![S_1]\!] \approx [\![S_2]\!]$$

Provided that :

- Encryption is
  - IND-CPA
  - message length-concealing
    $\mathrm{Pat}(\mathrm{enc}(n, k)) = \Box = \mathrm{Pat}(\mathrm{enc}(\langle n\ , n, n, n\rangle, k))$
  - which key-concealing
    $\mathrm{Pat}(\mathrm{enc}(n, k), \mathrm{enc}(n', k)) = \Box = \mathrm{Pat}(\mathrm{Pat}(\mathrm{enc}(n, k), \mathrm{enc}(n', k')))$
- S_1, S_2 contain no key cycles
  Examples : $\mathrm{enc}(k, k)$ or $\mathrm{enc}(k_1, k_2), \mathrm{enc}(k_2, k_1)$

Proof of soundness of indistinguishability

Lemma (Main lemma)

$$[\![S]\!] \approx [\![\mathrm{Pat}(S)]\!]$$

We can then easily deduce the main theorem.

Proof of soundness of indistinguishability

Lemma (Main lemma)

$$[\![S]\!] \approx [\![\mathrm{Pat}(S)]\!]$$

We can then easily deduce the main theorem.

Indeed, assume $\mathrm{Pat}(S_1) \equiv \mathrm{Pat}(S_2)$.

1. By the lemma, we have $[\![S_1]\!] \approx [\![\mathrm{Pat}(S_1)]\!]$ and $[\![S_2]\!] \approx [\![\mathrm{Pat}(S_2)]\!]$.
2. Then $\mathrm{Pat}(S_1) \equiv \mathrm{Pat}(S_2)$ implies $[\![\mathrm{Pat}(S_1)]\!] \approx [\![\mathrm{Pat}(S_2)]\!]$.

Proof of the main lemma $[\![S]\!] \approx [\![\mathrm{Pat}(S)]\!]$

Main steps :

Renaming Let $K_1, \ldots, k_n$ be the hidden (non deducible) keys of S and $J_1, \ldots, J_l$ be the visible (deducible) keys of S.

Since S contain no key cycles,
we may assume that K_j does not encrypt k_i whenever $i < j$.

Proof of the main lemma $[\![S]\!] \approx [\![\mathrm{Pat}(S)]\!]$

Main steps :

Renaming Let $K_1, \ldots, k_n$ be the hidden (non deducible) keys of S and $J_1, \ldots, J_l$ be the visible (deducible) keys of S.

Since S contain no key cycles,
we may assume that K_j does not encrypt k_i whenever $i < j$.

Intermediate patterns We define a sequence $\mathrm{Pat}_o(S), \ldots, \ldots \mathrm{Pat}_n(S)$ such that $\mathrm{Pat}_o(S) = \mathrm{Pat}(S)$ and $\mathrm{Pat}_n(S) = S$

Proof of the main lemma $[\![S]\!] \approx [\![\mathrm{Pat}(S)]\!]$

Main steps :

Renaming Let $K_1, \ldots, k_n$ be the hidden (non deducible) keys of S and $J_1, \ldots, J_l$ be the visible (deducible) keys of S.

Since S contain no key cycles,
we may assume that K_j does not encrypt k_i whenever $i < j$.

Intermediate patterns We define a sequence
$\mathrm{Pat}_o(S), \ldots, \ldots \mathrm{Pat}_n(S)$ such that
$\mathrm{Pat}_o(S) = \mathrm{Pat}(S)$ and $\mathrm{Pat}_n(S) = S$

Hybrid argument If $[\![S]\!] \not\approx [\![\mathrm{Pat}(S)]\!]$ then there exists i such that
$[\![\mathrm{Pat}_i(S)]\!] \not\approx [\![\mathrm{Pat}_{i+1}(S)]\!]$.

Proof of the main lemma $[\![S]\!] \approx [\![\mathrm{Pat}(S)]\!]$

Main steps :

Renaming Let $K_1, \ldots, k_n$ be the hidden (non deducible) keys of S and $J_1, \ldots, J_l$ be the visible (deducible) keys of S.

Since S contain no key cycles,
we may assume that K_j does not encrypt k_i whenever $i < j$.

Intermediate patterns We define a sequence $\mathrm{Pat}_o(S), \ldots, \ldots \mathrm{Pat}_n(S)$ such that $\mathrm{Pat}_o(S) = \mathrm{Pat}(S)$ and $\mathrm{Pat}_n(S) = S$

Hybrid argument If $[\![S]\!] \not\approx [\![\mathrm{Pat}(S)]\!]$ then there exists i such that $[\![\mathrm{Pat}_i(S)]\!] \not\approx [\![\mathrm{Pat}_{i+1}(S)]\!]$.

Security of encryption $[\![\mathrm{Pat}_i(S)]\!] \not\approx [\![\mathrm{Pat}_{i+1}(S)]\!]$ contradicts the security of encryption.

Intermediate patterns

Let $K_1, \ldots, k_n$ be the hidden (non deducible) keys of S and $J_1, \ldots, J_l$ be the visible (deducible) keys of S such that K_j does not encrypt k_i whenever $i < j$.

Definition (Intermediate patterns)

$$\mathrm{Pat}_i(S) = \mathrm{Pat}_{S \cup \{K_1, \ldots, K_i\}}(S)$$

$\mathrm{Pat}_i(S)$: what is visible to an intruder, with the extra knowledge $K_1, \ldots, K_i$.

Example of intermediate patterns

Visible keys : J_1, J_2
Hidden keys : K_1, K_2

$$S = \mathrm{Pat}_2(S) = \quad \mathrm{enc}(\langle \mathrm{enc}(J_1, K_2)\ , J_1\rangle, K_1), \quad \mathrm{enc}(J_1, J_2), \quad J_2$$

Example of intermediate patterns

Visible keys : J_1, J_2
Hidden keys : K_1, K_2

$$S = \mathrm{Pat}_2(S) = \quad \mathrm{enc}(\langle \mathrm{enc}(J_1, K_2)\ , J_1\rangle, K_1), \quad \mathrm{enc}(J_1, J_2), \quad J_2$$

$$\mathrm{Pat}_1(S) = \quad \mathrm{enc}(\langle \Box\ , J_1\rangle, K_1), \quad \mathrm{enc}(J_1, J_2), \quad J_2$$

Example of intermediate patterns

Visible keys : J_1, J_2
Hidden keys : K_1, K_2

$$\begin{array}{rcccc}
S = \mathsf{Pat}_2(S) = & \mathsf{enc}(\langle \mathsf{enc}(J_1, K_2)\ , J_1\rangle, K_1), & \mathsf{enc}(J_1, J_2), & J_2 \\
\mathsf{Pat}_1(S) = & \mathsf{enc}(\langle \square\ , J_1\rangle, K_1), & \mathsf{enc}(J_1, J_2), & J_2 \\
\mathsf{Pat}(S) = \mathsf{Pat}_0(S) = & \square, & \mathsf{enc}(J_1, J_2), & J_2
\end{array}$$

Hybrid argument

$$\mathrm{Pat}(S) = \mathrm{Pat}_0(S) \qquad \mathrm{Pat}_1(S) \quad \cdots \quad \mathrm{Pat}_{n-1}(S) \qquad \mathrm{Pat}_n(S) = S$$

Assume by contradiction that $[\![\mathrm{Pat}(S)]\!] \not\approx [\![S]\!]$.

Then, since the number n of intermediate steps is fixed, there must exist i such that

$$[\![\mathrm{Pat}_i(S)]\!] \not\approx [\![\mathrm{Pat}_{i+1}(S)]\!]$$

Exercises

Abstracting indistinguishability in various contexts

1. How to adapt the definition of patterns for encryption schemes that are not which key-concealing ?
2. How to adapt the definition of patterns for encryption schemes that are not message length-concealing ?
3. How to adapt the definition of patterns for asymmetric encryption schemes ?

Active Case

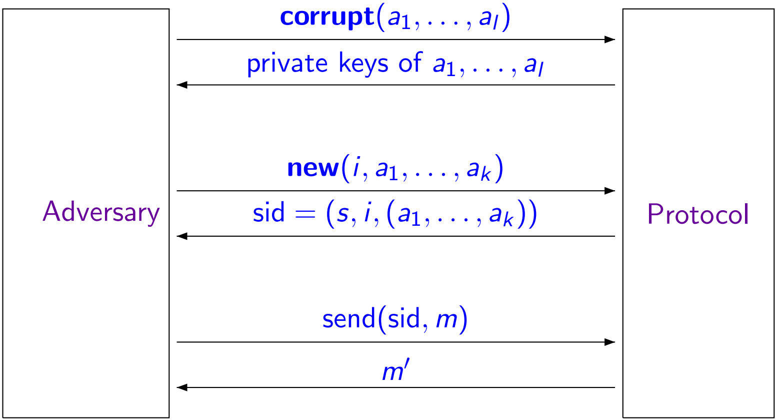
Can we extend the work to the active case ?

that is,

Are standard Dolev-Yao models sound w.r.t. to computational ones ?

A common setting

Same setting in formal and cryptographic models



Formal Intruder Deduction Rules

$$\frac{S\vdash m_1 \quad S\vdash m_2}{S\vdash \langle m_1, m_2\rangle} \qquad \frac{S\vdash \langle m_1, m_2\rangle}{S\vdash m_i}\, i \in \{1,2\}$$

$$\frac{S\vdash \mathrm{ek}(b) \quad S\vdash m}{S\vdash \{m\}_{\mathrm{ek}(b)}^{\mathrm{adv}(i)}}\, i \in \mathbb{N} \qquad \frac{S\vdash \{m\}_{\mathrm{ek}(b)}^{l} \quad S\vdash \mathrm{dk}(b)}{S\vdash m}$$

$$\frac{S\vdash \mathrm{sk}(b) \quad S\vdash m}{S\vdash [m]_{\mathrm{sk}(b)}^{\mathrm{adv}(i)}}\, i \in \mathbb{N} \qquad \frac{S\vdash [m]_{\mathrm{sk}(b)}^{l}}{S\vdash m}$$

Result : Soundness of trace properties

Theorem (extension of [Micciancio Warinschi TCC'04])

Every concrete trace is the image of a valid formal trace, except with negligible probability.

Result : Soundness of trace properties

Theorem (extension of [Micciancio Warinschi TCC'04])

Every concrete trace is the image of a valid formal trace, except with negligible probability.

Corollary :

Let Π *be protocol,* P^s *an arbitrary predicate on formal traces and* P^c *its corresponding predicate on concrete traces.*

Then $\Pi \models^s P^s$ *implies* $\Pi \models^c P^c$.

Result : Soundness of trace properties

Theorem (extension of [Micciancio Warinschi TCC'04])

Every concrete trace is the image of a valid formal trace, except with negligible probability.

Corollary :

Let Π be protocol, P^s an arbitrary predicate on formal traces and P^c its corresponding predicate on concrete traces.

Then $\Pi \models^s P^s$ implies $\Pi \models^c P^c$.

Applications : authentication, secrecy, ...

Hypotheses on the Implementation

- encryption : IND-CCA2
  $\longrightarrow$ the adversary cannot distinguish between $\{n_0\}_k$ and $\{n_1\}_k$ even if he has access to encryption and decryption oracles.
- signature : randomized and existentially unforgeable under chosen-message attack *i.e.* one can not produce a valid pair (m, σ)
- parsing :
  - each bit-string has a label which indicates his type (identity, nonce, key, signature, ...)
  - one can retrieve the (public) encryption key from an encrypted message.
  - one can retrieve the signed message from the signature

skip the proof

Proof idea

Proof technique : Reducing the protocol security to the robustness of the primitives (which itself reduces to hardness of algorithmic problem like integer factorization).

$$\mathcal{A} \text{ breaks } \mathcal{P} \Rightarrow \mathcal{A}' \text{ breaks } \{\ \} \text{ or sign}$$

Proof idea

Proof technique : Reducing the protocol security to the robustness of the primitives (which itself reduces to hardness of algorithmic problem like integer factorization).

$$\mathcal{A} \text{ breaks } \mathcal{P} \Rightarrow \mathcal{A}' \text{ breaks } \{\ \} \text{ or sign}$$

Example : If a computational (concrete) adversary $\mathcal{A}$ is able to compute $\{n_a\}_{K_a}$ out of $\{< A, n_a >\}_{K_a}$,
Then we can build an adversary $\mathcal{A}'$ that breaks the encryption $\{\ \}_{K_a}$.

Proof idea

Key result : every concrete trace is the image of a valid formal trace, except with negligible probability.

$$
\begin{array}{ccccccc}
 & \mathit{init}(1,a,b) & \rightarrow & \{a,n_a\}_{K_b} & & \{n_a\}_{K_b}\text{non valid !} \\
 & \uparrow & & \downarrow & & \uparrow \\
\mathcal{A}: & \mathit{init}(1,a,b) & & m_1 & \rightarrow & \mathit{send}(m_2)
\end{array}
$$

Proof idea

Key result : every concrete trace is the image of a valid formal trace, except with negligible probability.

$$
\begin{array}{lcccc}
 & init(1,a,b) & \rightarrow \quad \{a, n_a\}_{K_b} & & \{n_a\}_{K_b}\text{non valid !} \\
 & \uparrow & \downarrow & & \uparrow \\
\mathcal{A}: & init(1,a,b) & m_1 & \rightarrow & send(m_2)
\end{array}
$$

Using the adversary $\mathcal{A}$, we build an adversary $\mathcal{A}'$ that breaks encryption.

$$
\mathcal{A}' : \quad (\langle a, n_a^0\rangle, \langle a, n_a^1\rangle) \rightarrow \begin{array}{c}\text{encryption}\\ \text{oracle}\end{array} \rightarrow \{a, n_a^\alpha\}_{K_b}
$$

Proof idea

Key result : every concrete trace is the image of a valid formal trace, except with negligible probability.

$$
\begin{array}{lcccccc}
 & init(1,a,b) & \rightarrow & \{a, n_a\}_{K_b} & & \{n_a\}_{K_b}\text{non valid !} \\
 & \uparrow & & \downarrow & & \uparrow \\
\mathcal{A}: & init(1,a,b) & & m_1 & \rightarrow & send(m_2)
\end{array}
$$

Using the adversary $\mathcal{A}$, we build an adversary $\mathcal{A}'$ that breaks encryption.

$$
\mathcal{A}' : \quad (\langle a, n_a^0\rangle, \langle a, n_a^1\rangle) \rightarrow \begin{array}{c}\text{encryption}\\ \text{oracle}\end{array} \rightarrow \{a, n_a^\alpha\}_{K_b}
$$

$$
\rightarrow \mathcal{A} \rightarrow \{n_a^\alpha\}_{K_b}
$$

Proof idea

Key result : every concrete trace is the image of a valid formal trace, except with negligible probability.

$$
\begin{array}{lccccc}
 & init(1,a,b) & \rightarrow & \{a, n_a\}_{K_b} & & \{n_a\}_{K_b}\text{non valid !} \\
 & \uparrow & & \downarrow & & \uparrow \\
\mathcal{A}: & init(1,a,b) & & m_1 & \rightarrow & send(m_2)
\end{array}
$$

Using the adversary $\mathcal{A}$, we build an adversary $\mathcal{A}'$ that breaks encryption.

$$
\mathcal{A}' : \quad (\langle a, n_a^0\rangle, \langle a, n_a^1\rangle) \rightarrow \text{encryption oracle} \rightarrow \{a, n_a^\alpha\}_{K_b}
$$

$$
\rightarrow \mathcal{A} \rightarrow \{n_a^\alpha\}_{K_b} \rightarrow \text{decryption oracle} \rightarrow n_a^\alpha \rightarrow \alpha
$$

Trace properties vs observational equivalence

Fact 1 : Computational security properties are often stated as indistinguishability games rather than trace properties.
Example : secrecy, ideal functionalities, ...

Trace properties vs observational equivalence

Fact 1 : Computational security properties are often stated as indistinguishability games rather than trace properties.
Example : secrecy, ideal functionalities, ...

Fact 2 : Some security properties cannot be expressed as trace properties.
Example : Privacy properties of e-voting protocols

$$P(A, a) \| P(B, b) \sim_o P(A, b) \| P(B, a)$$

Correspondence of computational secrecy

Theorem

Symbolic secrecy implies computational secrecy.

- For protocols with only public key encryption, signatures and nonces
- Provided the public key encryption and the signature algorithms verify strong existing cryptographic properties (IND-CCA2, existentially unforgeable),

The previous result does not work in general

Example

$A \rightarrow B : h(s)$

s is inaccessible but not indistinguishable to an attacker :

$h(n_b), n_0, n_1 \rightarrow b$

The previous result does not work in general

Example

$A \rightarrow B : h(s)$

s is inaccessible but not indistinguishable to an attacker :
$h(n_b), n_0, n_1 \rightarrow b$

Results :

1. Design of a new formal secrecy property
2. Proof of its soundness and its faithfulness w.r.t. indistinguishability in our new setting :
   - pairing
   - asymmetric encryption
   - hashes (random oracle model)
3. NP-completeness of the secrecy property

Patterns : extension to hashes

Given $S = \{M_1, M_2, \ldots, M_k\}$ and some addintional knowledge T, we define

$\mathrm{Pat}_T(S) = \{\mathrm{Pat}^{S\cup\{T\}}(M_1), \mathrm{Pat}^{S\cup\{T\}}(M_2), \ldots, \mathrm{Pat}^{S\cup\{T\}}(M_k)\}$ with

Patterns : extension to hashes

Given $S = \{M_1, M_2, \ldots, M_k\}$ and some addintional knowledge T, we define

$$\mathrm{Pat}_T(S) = \{\mathrm{Pat}^{S\cup\{T\}}(M_1), \mathrm{Pat}^{S\cup\{T\}}(M_2), \ldots, \mathrm{Pat}^{S\cup\{T\}}(M_k)\} \text{ with}$$

$$\mathrm{Pat}^S(a) = \begin{cases} a & \text{if } S \vdash a \\ \Box & \text{otherwise} \end{cases}$$

Patterns : extension to hashes

Given $S = \{M_1, M_2, \ldots, M_k\}$ and some addintional knowledge T, we define

$$\mathrm{Pat}_T(S) = \{\mathrm{Pat}^{S\cup\{T\}}(M_1), \mathrm{Pat}^{S\cup\{T\}}(M_2), \ldots, \mathrm{Pat}^{S\cup\{T\}}(M_k)\} \text{ with}$$

$$\begin{aligned}
\mathrm{Pat}^S(a) &= \begin{cases} a & \text{if } S \vdash a \\ \square & \text{otherwise} \end{cases} \\
\mathrm{Pat}^S(\langle M_1, M_2\rangle) &= \langle \mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2)\rangle
\end{aligned}$$

Patterns : extension to hashes

Given $S = \{M_1, M_2, \ldots, M_k\}$ and some addintional knowledge T, we define

$$\mathrm{Pat}_T(S) = \{\mathrm{Pat}^{S\cup\{T\}}(M_1), \mathrm{Pat}^{S\cup\{T\}}(M_2), \ldots, \mathrm{Pat}^{S\cup\{T\}}(M_k)\} \text{ with}$$

$$\begin{array}{rcl}
\mathrm{Pat}^S(a) & = & \left\{\begin{array}{ll} a & \text{if } S \vdash a \\ \square & \text{otherwise} \end{array}\right. \\
\mathrm{Pat}^S(\langle M_1, M_2\rangle) & = & \langle \mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2)\rangle \\
\mathrm{Pat}^S(\{M\}^r_{\mathrm{ek}(a)}) & = & \left\{\begin{array}{ll} \{\mathrm{Pat}^S(M)\}^r_{\mathrm{ek}(a)} & \text{if } \; S \vdash \mathrm{dk}(a) \\ \{\mathrm{Pat}^S(M)\}^r_{\mathrm{ek}(a)} & \text{or if } r \in \mathrm{Rand}_{adv} \\ \square & \text{otherwise} \end{array}\right.
\end{array}$$

Patterns : extension to hashes

Given $S = \{M_1, M_2, \ldots, M_k\}$ and some addintional knowledge T, we define

$$\mathrm{Pat}_T(S) = \{\mathrm{Pat}^{S\cup\{T\}}(M_1), \mathrm{Pat}^{S\cup\{T\}}(M_2), \ldots, \mathrm{Pat}^{S\cup\{T\}}(M_k)\} \text{ with}$$

$$\begin{array}{rcl}
\mathrm{Pat}^S(a) & = & \left\{\begin{array}{ll} a & \text{if } S \vdash a \\ \square & \text{otherwise} \end{array}\right. \\
\mathrm{Pat}^S(\langle M_1, M_2\rangle) & = & \langle \mathrm{Pat}^S(M_1), \mathrm{Pat}^S(M_2)\rangle \\
\mathrm{Pat}^S(\{M\}^r_{\mathrm{ek}(a)}) & = & \left\{\begin{array}{ll} \{\mathrm{Pat}^S(M)\}^r_{\mathrm{ek}(a)} & \text{if } S \vdash \mathrm{dk}(a) \\ \{\mathrm{Pat}^S(M)\}^r_{\mathrm{ek}(a)} & \text{or if } r \in \mathrm{Rand}_{adv} \\ \square & \text{otherwise} \end{array}\right. \\
\mathrm{Pat}^S(h(M)) & = & \left\{\begin{array}{ll} h(\mathrm{Pat}^S(M)) & \text{if } S \vdash M \\ \square & \text{otherwise} \end{array}\right.
\end{array}$$

Examples

- $\phi_1 = \{h(\langle n_b, n' \rangle)\}$. Then $\mathsf{Pat}_{n_b}(\phi_1) = \{\Box\}$
  $\longrightarrow$ n_b is intuitively hidden by n'.

Examples

- $\phi_1 = \{h(\langle n_b, n' \rangle)\}$. Then $\mathrm{Pat}_{n_b}(\phi_1) = \{\Box\}$
  $\rightarrow$ n_b is intuitively hidden by n'.

- $\phi_2 = \{h(\langle n_b, \{n'\}^r_{ek(a)} \rangle), n'\}$. $\mathrm{Pat}_{n_b}(\phi_2) = \{\Box, n'\}$.
  $\rightarrow$ The encryption of n' does hide n_b.

Pattern-based secrecy definition

Π protocol
$X^j_{A_i}$ nonce variable occurring in some role A_i.
$\mathcal{M}$ set of sent messages
s session number

Definition

$X^j_{A_i}$ is secret in Π, written $\Pi \models^f \mathrm{Invisible}(i,j)$, if :

$$n^{a_i,j,s} \text{ does not occur in } \mathrm{Pat}_{n^{a_i,j,s}}(\mathcal{M}) \qquad \forall \mathcal{M} \in \mathrm{Exec}(\Pi)\ \forall s$$

Soundness and decidability of the secrecy property

Theorem

$$\Pi \models^f \mathsf{Invisible}^f(i,j) \quad \textit{iff} \quad \Pi \models^c \mathsf{Indist}(i,j)$$

Remark : Our formal secrecy definition is both sufficient and necessary for indistinguishability in the computational world.

Soundness and decidability of the secrecy property

Theorem

$$\Pi \models^f \mathsf{Invisible}^f(i,j) \quad \textit{iff} \quad \Pi \models^c \mathsf{Indist}(i,j)$$

Remark : Our formal secrecy definition is both sufficient and necessary for indistinguishability in the computational world.

Theorem

Deciding $\Pi \models^f \mathsf{Invisible}^f(i,j)$ *is NP-complete for a finite number of sessions.*

General computational indistinguishability

Observational equivalence is a sound abstraction of computational indistinguishability.

$$P \sim_o Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

- For simple processes
  (A fragment of applied pi-calculus that captures most security protocols)
- For symmetric encryption implemented using IND-CC2 schemes

General computational indistinguishability

Observational equivalence is a sound abstraction of computational indistinguishability.

$$P \sim_o Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

- For simple processes
  (A fragment of applied pi-calculus that captures most security protocols)
- For symmetric encryption implemented using IND-CC2 schemes

Limitation : No dishonest keys !

(currently solved by Guillaume Scerri)

Related Work

Abadi-Rogaway, followed by several extensions : passive case.

Backes-Pfitzmann

- very general results : symmetric and asymmetric encryption, pairing, signatures, MACs.
- less abstract model than classical Dolev-Yao models,

Laud : specialized decision procedure for symmetric encryption

Datta-Derek-Mitchell-Shmatikov-Turuani : symbolic deduction system for proofs in the concrete model (asymmetric encryption, no automatic procedure)

Blanchet : direct automation of the (game-based) cryptographic proofs in the concrete model → tool CryptoVerif

Conclusion

Formal methods form a powerful approach for analyzing security protocols

- Makes use of classical techniques in formal methods : term algebra, equational theories, clauses and resolution techniques, tree automata, etc.
  $\Rightarrow$ Many decision procedures
- Several automatic tools
  - For successfully detecting attacks on protocols (e.g. Casper, Avispa)
  - For proving security for an arbitrary number of sessions (e.g. ProVerif)
- Provides cryptographic guarantees under classical assumptions on the implementation of the primitives

Some current directions of research

- Enriching the symbolic model
  - Considering more equational theories (e.g. theories for e-voting protocols)
  - Adding more complex structures for data (list, XML, ...)
  - Considering recursive protocols (e.g. group protocol) where the number of message exchanges in a session is not fixed
  - Proving more complex security properties like equivalence-based properties (e.g. for anonymity or e-voting protocols)
- With cryptographic guarantees
  - Combining formal and cryptographic models for more complex primitives and security properties.
  - How far can we go ?
  - Is it possible to consider weaker cryptographic primitives ?