A Biased Survey of Models and Methods for Verifying Cryptographic Protocols

Jean Goubault-Larrecq

Classical and Quantum Information Security — Dec 15, 2005
Why Cryptographic Protocols?
Cryptographic Protocols and Attacks
Dolev-Yao Models
The Original Dolev-Yao Model (1983)
Dolev-Yao and First-Order Logic
Equational Theories
Other Applications
Spi-Calculus and Friends: Observational Equivalence
The Idea of Process Algebra
Are Dolev-Yao Models too Weak?
Observational Equivalence, Bisimulation
Relation to Computational Security
Conclusion
What About Quantum Protocols?
Cryptographic Protocols

Cryptography:
Cryptographic Protocols

Cryptography: Protocols:

We may seek various properties:

(only a sample!
and pretty informal definitions for now, too)
Cryptography: Protocols:

We may seek various properties:

**secrecy**: $M$ is secret if no adversary can emit $M$;
Cryptography:  

We may seek various properties:  

**secrecy, authenticity** (one form): the only process that can emit \( M \) is \( A \);
Why Cryptographic Protocols?
Dolev-Yao Models
Spi-Calculus and Friends: Observational Equivalence
Relation to Computational Security
Conclusion

Cryptographic Protocols

Cryptography: Protocols:

We may seek various properties:

**secrecy, authenticity, anonymity**: you cannot link $A$ with her message $M$, although $A$ and $M$ may be public;

(only a sample! and pretty informal definitions for now, too)

Jean Goubault-Larrecq
Proving Security
Cryptography: Protocols:

We may seek various properties:

**secrecy, authenticity, anonymity, fairness:** A cannot prove to C that she promised to sign with B before A and B indeed signed (together); etc.
Outline

Why Cryptographic Protocols?
- Cryptographic Protocols... and Attacks

Dolev-Yao Models
- The Original Dolev-Yao Model (1983)
- Dolev-Yao and First-Order Logic
- Equational Theories
- Other Applications

Spi-Calculus and Friends: Observational Equivalence
- The Idea of Process Algebra
- Are Dolev-Yao Models too Weak?
- Observational Equivalence, Bisimulation

Relation to Computational Security

Conclusion

What About Quantum Protocols?
Fact: Cryptography is Not Enough

Even if you use perfect encryption algorithms (unbreakable), it is not easy to establish secrecy or authentication:

(Assumption: Kab is a secret key, shared between A et B – no adversary knows it)

How do you guarantee this?
Fact: Cryptography is Not Enough

Even if you use perfect encryption algorithms (unbreakable), it is not easy to establish secrecy or authentication.

The Needham-Schroeder symmetric key protocol:

1. \( A \rightarrow S : A, B, N_a \)
2. \( S \rightarrow A : \{ N_a, B, K_{ab}, \{ K_{ab}, A \}_K_{bs} \}_K_{as} \)
3. \( A \rightarrow B : \{ K_{ab}, A \}_K_{bs} \)
4. \( B \rightarrow A : \{ N_b \}_K_{ab} \)
5. \( A \rightarrow B : \{ N_b + 1 \}_K_{ab} \)
Fact: Cryptography is Not Enough

Even if you use perfect encryption algorithms (unbreakable), it is not easy to establish **secrecy** or **authentication**. The **Needham-Schroeder** symmetric key protocol... and its attack:

\[
\text{C} \quad \text{A} \quad \text{B} \quad \text{S}
\]

- **C** writes \( \{K_{ab_0}, A\} K_{bs} \)
- **A** reads \( \{K_{ab_0}\} K_{bs} \)
- **A** writes \( \{N_b\} K_{ab_0} \)
- **B** reads \( \{N_b + 1\} K_{ab_0} \)
- **B** new \( N_b \)
- **B** writes \( \{N_b\} K_{ab_0} \)
- **S** reads \( \{N_b+1\} K_{ab_0} \)
Fact: Cryptography is Not Enough

Even if you use perfect encryption algorithms (unbreakable), it is not easy to establish secrecy or authentication.
A Word of Warning

This survey is *partial*. I could talk on models and methods for verifying protocols for *hours*. Instead, this talk concentrates on:

- **Logic-based models of security;**
  - no algebra, no probabilities, no Turing machines involved here.
A Word of Warning

This survey is *partial*. I could talk on models and methods for verifying protocols for *hours*. Instead, this talk concentrates on:

- **Logic-based models of security;**
  no algebra, no probabilities, no Turing machines involved here.

- with a stress on **automation** of proof search;
A Word of Warning

This survey is *partial*. I could talk on models and methods for verifying protocols for *hours*. Instead, this talk concentrates on:

- **Logic-based** models of security;
  - no algebra, no probabilities, no Turing machines involved here.
- with a stress on **automation** of proof search;
- in the **classical**, non-quantum setting;
Outline

Why Cryptographic Protocols?
Cryptographic Protocols and Attacks

Dolev-Yao Models
The Original Dolev-Yao Model (1983)
Dolev-Yao and First-Order Logic
Equational Theories
Other Applications

Spi-Calculus and Friends: Observational Equivalence
The Idea of Process Algebra
Are Dolev-Yao Models too Weak?
Observational Equivalence, Bisimulation
Relation to Computational Security
Conclusion

What About Quantum Protocols?
Fundamentals of Dolev-Yao, with a Modern View

- Handles reachability properties, e.g. secrecy:

$$\neg EF \text{ knows}(M_{\text{secret}})$$
Fundamentals of Dolev-Yao, with a Modern View

- Handles **reachability** properties, e.g. secrecy:

  \[ \neg \text{EF knows}(M_{\text{secret}}) \]

  Handles **safety** properties, e.g., authentication,

  \[ \neg \text{EF}(\exists M \cdot B_{\text{received}}(M) \land \neg G^{-1}A_{\text{sent}}(M)) \]

  by the standard history variable trick.
Fundamentals of Dolev-Yao, with a Modern View

- Handles **reachability** properties, e.g. secrecy:

  \[ \neg \text{EF knows}(M_{secret}) \]

- Assumes an **active** adversary, which can eavesdrop, forge messages, reroute communication, play with arbitrary many sessions (even in parallel) of several protocols.
Fundamentals of Dolev-Yao, with a Modern View

- Handles reachability properties, e.g. secrecy:

\[ \neg \text{EF knows}(M_{\text{secret}}) \]

- Assumes an active adversary, which can eavesdrop, forge messages, reroute communication, play with arbitrary many sessions (even in parallel) of several protocols.

This actually simplifies the model: each message sent is sent to the adversary, each message received was built by the adversary.
Why Cryptographic Protocols?
Dolev-Yao Models
Spi-Calculus and Friends: Observational Equivalence
Relation to Computational Security
Conclusion

Dolev-Yao Models


Fundamentals of Dolev-Yao, with a Modern View

- Handles reachability properties, e.g. secrecy:
  \[ \neg \text{EF knows}(M_{\text{secret}}) \]

- Assumes an active adversary, which can eavesdrop, forge messages, reroute communication, play with arbitrary many sessions (even in parallel) of several protocols.

- Assumes perfect cryptography primitives:
  “The only equations that hold between terms (with non-negligible probability) are \( M = M \).”
Fundamentals of Dolev-Yao, with a Modern View

- Handles reachability properties, e.g. secrecy:
  \[ \neg \text{EF knows}(M_{\text{secret}}) \]

- Assumes an active adversary, which can eavesdrop, forge messages, reroute communication, play with arbitrary many sessions (even in parallel) of several protocols.

- Assumes perfect cryptography primitives:
  “The only equations that hold between terms (with non-negligible probability) are \( M = M \).”

A kind of idealization of [Dolev-Dwork-Naor, STOC’91]’s notion of non-malleable cryptography.
Fundamentals of Dolev-Yao, with a Modern View

- Handles reachability properties, e.g. secrecy:
  \[ \neg \text{EF knows}(M_{secret}) \]

- Assumes an active adversary, which can eavesdrop, forge messages, reroute communication, play with arbitrary many sessions (even in parallel) of several protocols.

- Assumes perfect cryptography primitives:
  “The only equations that hold between terms (with non-negligible probability) are \( M = M' \).”

- Models all capabilities of the adversary by a deduction system (see next slide).
Formalizing the Adversary’s Knowledge

Given set $E$ of messages eavesdropped by adversary, say that $M$ is **deducible** from $E$, in notation $E \vdash M$, iff:

$\begin{align*}
E, M & \vdash M \quad (Ax) \\
E \vdash M & \quad (CryptI) \\
E \vdash \{M\}_K & \quad (CryptE) \\
E \vdash (M_1, \ldots, M_n) & \quad (TupleI) \\
E \vdash M_i & \quad (TupleE_i), 1 \leq i \leq n
\end{align*}$
Formalizing the Adversary’s Knowledge

Given set $E$ of messages eavesdropped by adversary, say that $M$ is **deducible** from $E$, in notation $E \vdash M$, iff:

$E, M \vdash M$  

$E \vdash M$  

$E \vdash K$  

$E \vdash \{M\}_K$  

(CryptI)

$E \vdash \{M\}_K$  

$E \vdash K'$  

$E \vdash M$  

$E \vdash M$  

$(K' \text{ inverse of } K)$  

(CryptE)

$E \vdash M_1 \ldots E \vdash M_n$  

$E \vdash (M_1, \ldots, M_n)$  

(TupleI)

$E \vdash (M_1, \ldots, M_n)$  

$E \vdash M_i$  

(TupleE$_i$), $1 \leq i \leq n$

- **Sending** $M$ means adding $M$ to $E$;
Formalizing the Adversary’s Knowledge

Given set \( E \) of messages eavesdropped by adversary, say that \( M \) is **deducible** from \( E \), in notation \( E \vdash M \), iff:

\[
E, M \vdash M \quad (Ax)
\]

\[
E \vdash M \quad E \vdash K \quad \frac{}{E \vdash \{M\}_K} \quad \text{(CryptI)}
\]

\[
E \vdash \{M\}_K \quad E \vdash K' \quad \frac{}{E \vdash M} \quad \text{(CryptE)}
\]

\[
E \vdash M_1 \quad \ldots \quad E \vdash M_n \quad \frac{}{E \vdash (M_1, \ldots, M_n)} \quad \text{(TupleI)}
\]

\[
E \vdash (M_1, \ldots, M_n) \quad \frac{}{E \vdash M_i} \quad \text{(TupleE}_i) \quad 1 \leq i \leq n
\]

- **Sending** \( M \) means adding \( M \) to \( E \);
- **Receiving** a message \( M \) means that \( E \vdash M \).

Jean Goubault-Larrecq

Proving Security
Deciding Protocol Insecurity

**Thm** [Durgin-Lincoln-Mitchell-Scedrov, FMSP’99] Reachability (=Insecurity) in Dolev-Yao models is **undecidable**.

**Prf** By reduction from 2-counter machines.
Deciding Protocol Insecurity

**Thm** [Rusinowitch-Turuani, CSFW’01] Reachability in Dolev-Yao models with a **fixed number of sessions** is NP-complete. (Note: only finitely many nonces.)
Deciding Protocol Insecurity

**Thm** [Rusinowitch-Turuani, CSFW’01] Reachability in Dolev-Yao models with a *fixed number of sessions* is NP-complete. (Note: only finitely many nonces.)

**Prf** Guess an interleaving of all sessions, create a fresh constant for each nonce. Then solve constraints

\[ \vdash M_0 \] (first agent expects \( M_0 \), sends \( N_1 \))

\[ N_1 \vdash M_1 \] (second agent expects \( M_1 \), sends \( N_2 \))

\[ N_1, N_2 \vdash M_2 \]

\[ \vdots \]

\[ N_1, N_2, \ldots, N_k \vdash M_{\text{secret}} \] (adversary obtains secret)
Deciding Protocol Insecurity

**Thm** [Amadio-Charatonik, Concur’02] Under some (stringent, but necessary) conditions, with arbitrary many nonces, recursive agents but no forking, insecurity is decidable in EXPTIME.
Deciding Protocol Insecurity

**Thm** [Amadio-Charatonik, Concur’02] Under some (stringent, but necessary) conditions, with arbitrary many nonces, recursive agents but no forking, insecurity is decidable in EXPTIME.

**Prf** By encoding this into a (new) class of set constraints with renaming.
Deciding Protocol Insecurity

**Thm** [Amadio-Charatonik, Concur’02] Under some (stringent, but necessary) conditions, with *arbitrary many* nonces, recursive agents but no forking, insecurity is decidable in EXPTIME.

**Prf** By encoding this into a (new) class of set constraints with renaming.

**Thm** [Seidl-Verma, 05] With full parallel sessions, under the assumption of *single blind copying* and bounded number of nonces, insecurity is EXPTIME-complete.
Deciding Protocol Insecurity

**Thm** [Amadio-Charatonik, Concur’02] Under some (stringent, but necessary) conditions, with arbitrary many nonces, recursive agents but no forking, insecurity is decidable in EXPTIME.

**Prf** By encoding this into a (new) class of set constraints with renaming.

**Thm** [Seidl-Verma, 05] With full parallel sessions, under the assumption of single blind copying and bounded number of nonces, insecurity is EXPTIME-complete.

**Prf** By encoding into a decidable subclass of first-order logic (introduced in [Comon-Lundh-Cortier, RTA’03]).
Outline

Why Cryptographic Protocols?
  Cryptographic Protocols... and Attacks

Dolev-Yao Models
  The Original Dolev-Yao Model (1983)
  Dolev-Yao and First-Order Logic
  Equational Theories
  Other Applications

Spi-Calculus and Friends: Observational Equivalence
  The Idea of Process Algebra
  Are Dolev-Yao Models too Weak?
  Observational Equivalence, Bisimulation
  Relation to Computational Security
  Conclusion

What About Quantum Protocols?
Dolev-Yao in Full Parallel Multi-Session Mode

Pioneered by [Weidenbach, CADE’99] and [Blanchet, CSFW’01].

▷ Use a unary predicate $\text{knows}$ instead of a binary relation $\vdash$:

$\text{knows}(M) \iff \text{"the adversary is able to build } M \text{"} \text{ (at whatever stage of the protocol).}$
Dolev-Yao in Full Parallel Multi-Session Mode

Pioneered by [Weidenbach, CADE’99] and [Blanchet, CSFW’01].

- Use a unary predicate `knows` instead of a binary relation $\vdash$: 
  $\text{knows}(M) \iff \text{“the adversary is able to build } M\text{” (at whatever stage of the protocol).}$

- Abstract nonces $N_a$ as **Skolem functions** $N_a(X, Y, Z)$ of the protocol-dependent variables $X, Y, Z$ (identities, past nonces, etc.)
Dolev-Yao in Full Parallel Multi-Session Mode

Pioneered by [Weidenbach, CADE’99] and [Blanchet, CSFW’01].

- Use a **unary predicate** `knows` instead of a binary relation `⊢`:
  `knows(M) ⇔ “the adversary is able to build M”` (at whatever stage of the protocol).

- Abstract nonces `N_a` as **Skolem functions** `N_a(X, Y, Z)` of the protocol-dependent variables `X, Y, Z` (identities, past nonces, etc.)

- Encode everything into sets of **Horn clauses**.
Dolev-Yao in Full Parallel Multi-Session Mode

Pioneered by [Weidenbach, CADE’99] and [Blanchet, CSFW’01].

- Use a unary predicate $\text{knows}$ instead of a binary relation $\vdash$:
  $\text{knows}(M) \Leftrightarrow \text{"the adversary is able to build } M\text{" (at whatever stage of the protocol).}$

- Abstract nonces $N_a$ as Skolem functions $N_a(X, Y, Z)$ of the protocol-dependent variables $X, Y, Z$ (identities, past nonces, etc.)

- Encode everything into sets of Horn clauses.

  Yes, Horn clauses are undecidable, so what?
Dolev-Yao in Full Parallel Multi-Session Mode

Pioneered by [Weidenbach, CADE’99] and [Blanchet, CSFW’01].

- Use a unary predicate $\text{knows}$ instead of a binary relation $\vdash$:
  $$\text{knows}(M) \iff \text{“the adversary is able to build } M \text{” (at whatever stage of the protocol).}$$

- Abstract nonces $N_a$ as Skolem functions $N_a(X, Y, Z)$ of the protocol-dependent variables $X, Y, Z$ (identities, past nonces, etc.)

- Encode everything into sets of Horn clauses.

  Yes, Horn clauses are undecidable, so what?

Let’s examine our old friend, the symmetric key Needham-Schroeder protocol...
A Horn clause (pure Prolog) model

1. Abilities of the adversary.

\[
\text{knows}(\{M\}_K) \iff \text{knows}(M), \text{knows}(K) \quad (C \text{ can encrypt})
\]
\[
\text{knows}(M) \iff \text{knows}(\{M\}_{k_{(\text{sym},X)}},
\text{knows}(k_{(\text{sym},X)}) \quad \ldots \text{and decrypt [symmetric keys]})
\]
\[
\text{knows}([]) \quad (C \text{ can build}
\]
\[
\text{knows}(M_1 :: M_2) \iff \text{knows}(M_1), \text{knows}(M_2) \quad \text{any list of known messages}
\]
\[
\text{knows}(M_1) \iff \text{knows}(M_1 :: M_2) \quad (C \text{ can read heads})
\]
\[
\text{knows}(M_2) \iff \text{knows}(M_1 :: M_2) \quad (C \text{ can read tails})
\]
\[
\text{knows}(\text{suc}(M)) \iff \text{knows}(M) \quad (C \text{ can add}
\]
\[
\text{knows}(M) \iff \text{knows}(\text{suc}(M)) \quad \text{and subtract one})
\]
2. Protocol clauses—current sessions (à la Blanchet)

1. \( A \rightarrow S : A, B, N_a \) \( \text{knows}([a, b, na([a, b])]) \)

<table>
<thead>
<tr>
<th>Step</th>
<th>Protocol Clause</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>( A \rightarrow S : A, B, N_a )</td>
<td>( \text{knows}([N_a, B, k_{ab}, [K_{ab}, A]<em>{K</em>{bs}}]) ) ( \text{knows}([A, B, N_a]) ) ( (k_{ab} \equiv k(\text{sym}, \text{cur}(A, B, N_a))) )</td>
</tr>
<tr>
<td>2.</td>
<td>( S \rightarrow A : {N_a, B, K_{ab}, [K_{ab}, A]<em>{K</em>{bs}}} ) ( \text{knows}(M) ) ( \text{knows}([na([a, b]), b, K_{ab}, M]_{k(\text{sym}, [a, s])}) )</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>( A \rightarrow B : {K_{ab}, A}<em>{K</em>{bs}} )</td>
<td>( \text{a_key}(K_{ab}) ) ( \text{knows}([na([a, b]), b, K_{ab}, M]_{k(\text{sym}, [a, s])}) )</td>
</tr>
<tr>
<td>4.</td>
<td>( B \rightarrow A : {N_b}<em>{K</em>{ab}} )</td>
<td>( \text{knows}([K_{ab}, A]_{k(\text{sym}, [B, s])}) )</td>
</tr>
<tr>
<td>5.</td>
<td>( A \rightarrow B : {N_b + 1}<em>{K</em>{ab}} )</td>
<td>( \text{knows}([\text{suc}(N_b)]<em>{K</em>{ab}}) ) ( \text{knows}([N_b]<em>{K</em>{ab}}) )</td>
</tr>
</tbody>
</table>
3. Protocol clauses—old sessions

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$

$k_{sym}[s] = k_{sym}[s]$

$\left( k_{sym}[a] = k_{sym}[a] \right)$

$(k_{ab} \equiv k_{sym}(\text{prev}(A, B, N_a)))$
4. Initial Knowledge of the Adversary

\[
\begin{align*}
\text{agent}(a) & \quad \text{agent}(b) \\
\text{agent}(s) & \quad \text{agent}(i) \\
\text{knows}(X) & \iff \text{agent}(X) \\
\text{knows}(k(\text{pub}, X)) & \\
\text{knows}(k(\text{prv}, i)) & \\
\text{knows}(k(\text{sym}, \text{prev}(A, B, N_a))) & \quad (\text{old session keys are compromised})
\end{align*}
\]
5. Security queries

\[ \bot \iff \text{knows}(k(\text{sym}, \text{cur}(a, b, N_a))) \]

Can C build \( K_{ab} \) as created by S?

\[ \bot \iff \text{knows}(K_{ab}), \text{a\_key}(K_{ab}) \]

... as received by A?

\[ \bot \iff \text{knows}(\{\text{suc}(\text{nb}(K_{ab}, A, B))\}_K_{ab}), \text{knows}(K_{ab}) \]

... as received by B?
Important Remark: Security Proof = No Proof

A proof of ⊥ (false) is an attack.

... i.e., a way of running clauses 1.–5. which enables C to eventually know some sensitive data, here.
Important Remark: Security Proof = No Proof

A proof of $\perp$ (false) is an attack.

\[ \ldots \text{i.e., a way of running clauses 1.–5.} \]
which enables $C$ to eventually know some sensitive data, here.

Selinger’s Thesis: [Selinger, LACPV’01]

Security proof $\equiv$ no proof of $\perp$. 
Important Remark: Security Proof = No Proof

A proof of \( \bot \) (false) is an attack.

\[ \ldots \text{i.e., a way of running clauses 1.--5.} \]

which enables \( C \) to eventually know some sensitive data, here.

**Selinger’s Thesis:** [Selinger, LACPV’01]

Security proof \( \equiv \) no proof of \( \bot \).

Constructively, the non-existence of a proof will be witnessed by a model.

by completeness of first-order logic [Gödel, 1930].
Deciding Sets of Horn Clauses

- Blanchet calls a dedicated prover, either SPASS [Weidenbach, CADE’96], or his own, built into his tool ProVerif.
Deciding Sets of Horn Clauses

- Blanchet calls a dedicated prover, either SPASS [Weidenbach, CADE’96], or his own, built into his tool ProVerif.
  This may fail to terminate... except that [Blanchet-Podelski, FoSSaCs’03] tagging enforces termination of Blanchet’s selection strategy.
Deciding Sets of Horn Clauses

- Blanchet calls a dedicated prover, either SPASS [Weidenbach, CADE’96], or his own, built into his tool ProVerif.
- Or use upper approximations inside a decidable subclass of first-order logic:
  - Flat clauses [JGL, SECI’02] (DEXPTIME-complete);
  - $\mathcal{H}_1$ [Nielson-Nielson-Seidl, SAS’02; JGL, IPL’05] (DEXPTIME-complete, define regular tree languages).
Deciding Sets of Horn Clauses

- Blanchet calls a dedicated prover, either SPASS [Weidenbach, CADE’96], or his own, built into his tool ProVerif.
- Or use upper approximations inside a decidable subclass of first-order logic:
  - Flat clauses [JGL, SECI’02] (DEXPTIME-complete);
  - $\mathcal{H}_1$ [Nielson-Nielson-Seidl, SAS’02; JGL, IPL’05] (DEXPTIME-complete, define regular tree languages).

Note: the latter allows us to generate formal proofs of security in Coq automatically [JGL, JFLA’04], but this is another story.
How Verification Works, Using Upper Approximations

- Cryptographic protocols
  - Modélisation
  - Prolog programs = Horn clauses
    - h1 abstraction
      - Restricted Prolog programs ~ tree automata

Security guarantee
- No proof of false
  - Undecidable
  - Decidable

Jean Goubault-Larrecq
Proving Security
Outline

- Why Cryptographic Protocols?
  - Cryptographic Protocols... and Attacks
- Dolev-Yao Models
  - The Original Dolev-Yao Model (1983)
  - Dolev-Yao and First-Order Logic
- Equational Theories
- Other Applications
- Spi-Calculus and Friends: Observational Equivalence
  - The Idea of Process Algebra
  - Are Dolev-Yao Models too Weak?
  - Observational Equivalence, Bisimulation
- Relation to Computational Security
- Conclusion
- What About Quantum Protocols?
An Interesting Story: The Needham-Schroeder Public Key Protocol...

A's public key; A decrypts with her private key

B's public key; B decrypts using his private key

new Na
write {Na, A} Kb

read {Na, A} Kb^−1
new Nb
write {Na, Nb} Ka

A

B

read {=Na, Nb} Ka^−1
write {Nb} Kb

read {=Nb} Kb^−1
...Is Vulnerable to a Man-In-The-Middle Attack [Lowe, IPL’95]...

A starts talking with C... who diverts conversation

Here, B thinks he is talking with A
But B is actually talking with C.
...But Can Be Repaired Easily [Lowe, IPL’95]...
Security of Needham-Schroeder-Lowe

- You can then prove that Needham-Schroeder-Lowe is secure in the Dolev-Yao model.
Why Cryptographic Protocols?
Dolev-Yao Models
Spi-Calculus and Friends: Observational Equivalence
Relation to Computational Security
Conclusion

The Original Dolev-Yao Model (1983)
Dolev-Yao and First-Order Logic
Equational Theories
Other Applications

Security of Needham-Schroeder-Lowe

- You can then prove that Needham-Schroeder-Lowe is secure in the Dolev-Yao model.
- Now, implement this using El Gamal encryption:
  - Let $g$ be a generator of some cyclic group (say $\mathbb{Z}/p\mathbb{Z}^*$);
  - $A$ generates $x_A$ at random (nonce), publishes $g^{x_A}$;
  - $B$ generates $r$ at random, publishes $r$;
  - Let $K_a^{-1} = x_A$ ($A$’s private key), $K_a = g^{rx_A}$ ($A$’s public key);
You can then prove that Needham-Schroeder-Lowe is secure in the Dolev-Yao model.

Now, implement this using El Gamal encryption:
To encrypt $M$: $\{M\}_{K_a} = M \oplus K_a$ (≈ one-time pad).

About the most secure (classical) encryption scheme you can think of.
Security of Needham-Schroeder-Lowe

- You can then prove that Needham-Schroeder-Lowe is secure in the Dolev-Yao model.
- Now, implement this using El Gamal encryption:
  To encrypt $M$: $\{M\}_{Ka} = M \oplus Ka$ (≈ one-time pad).
  About the most secure (classical) encryption scheme you can think of.
- But you can replay Lowe’s attack, then [Joux, priv. comm., sep. 2002; Warinschi, CSFW’03]!

  $B$ sends
  $\{Na, Nb, B\}_{Ka}$
  $= (Na, Nb, B) \oplus Ka$
  Adversary xors with
  $(0, 0, B \oplus C)$
  $A$ expects
  $(Na, Nb, C) \oplus Ka$
  $= \{Na, Nb, C\}_{Ka}$
Security of Needham-Schroeder-Lowe

- You can then prove that Needham-Schroeder-Lowe is secure in the Dolev-Yao model.
- Now, implement this using El Gamal encryption:
  To encrypt $M$: $\{M\}_K = M \oplus K$ (≈ one-time pad).
  About the most secure (classical) encryption scheme you can think of.
- But you can replay Lowe’s attack.
- El Gamal encryption is malleable. Additional equations:
  \[
  \begin{align*}
  \{M\}_K &= M \oplus K \\
  M_1 \oplus M_2 &= M_2 \oplus M_1 \\
  M \oplus M &= 0 \\
  (M_1 \oplus M_2) \oplus M_3 &= M_1 \oplus (M_2 \oplus M_3) \\
  (M_1, M_2, M_3) \oplus (K_1, K_2, K_3) &= (M_1 \oplus K_1, M_2 \oplus K_2, M_3 \oplus K_3)
  \end{align*}
  \]
The Need for Equational Theories

We need to enrich the Dolev-Yao model with equations:

► See the Joux example: $\oplus$ is ACUI + homomorphic wrt. pairing;
The Need for Equational Theories

We need to enrich the Dolev-Yao model with equations:

- See the Joux example: $\oplus$ is ACUI + homomorphic wrt. pairing;
- Modeling Diffie-Hellman, i.e., Abelian group laws:

\[
(M_1 \times M_2) \times M_3 = M_1 \times (M_2 \times M_3) \quad M_1 \times M_2 = M_2 \times M_1 \\
M \times 1 = M \quad M \times M^{-1} = 1 \\
(XY)^{-1} = Y^{-1}X^{-1} \quad 1^{-1} = 1 \quad (X^{-1})^{-1} = X
\]

plus new adversary rules:

\[
\text{knows}(g^{X \times Y}) \iff \text{knows}(g^X), \text{knows}(Y) \\
\text{knows}(g^1) \\
\text{knows}(Y^{-1}) \iff \text{knows}(Y)
\]

(This is essentially the generic group model.)
The Need for Equational Theories

We need to enrich the Dolev-Yao model with equations:

- See the Joux example: $\oplus$ is ACUI + homomorphic wrt. pairing;
- Modeling Diffie-Hellman, i.e., Abelian group laws.
- Modeling RSA (exercise; hard, see e.g. [Kapur-Narendran-Wang, RTA’03]);
The Need for Equational Theories

We need to enrich the Dolev-Yao model with equations:

- See the Joux example: $\oplus$ is ACUI + homomorphic wrt. pairing;
- Modeling Diffie-Hellman, i.e., Abelian group laws.
- Modeling RSA (exercise; hard, see e.g. [Kapur-Narendran-Wang, RTA’03]);
- Representing ciphers (i.e., decryption never fails):

\[
\text{dec}(\{M\}_K, K) = M \quad \{\text{dec}(M, K)\}_K = M
\]
The Need for Equational Theories

We need to enrich the Dolev-Yao model with equations:

- See the Joux example: $\oplus$ is ACUI + homomorphic wrt. pairing;
- Modeling Diffie-Hellman, i.e., Abelian group laws.
- Modeling RSA (exercise; hard, see e.g. [Kapur-Narendran-Wang, RTA’03]);
- Representing ciphers (i.e., decryption never fails).
- Etc.
Formalizing the Adversary’s Knowledge... Modulo $T$

\[ E, M \vdash_T M \]

\[ E \vdash_T M \quad E \vdash_T K \]
\[ \quad \frac{}{E \vdash_T \{M\}_K} \quad (\text{CryptI}) \]

\[ E \vdash_T \{M\}_K \quad E \vdash_T K' \]
\[ \quad \frac{E' \vdash_T M}{E \vdash_T \{M\}_K} \quad (\text{CryptE}) \]

\[ E \vdash_T M_1 \quad \ldots \quad E \vdash_T M_n \]
\[ \quad \frac{}{E \vdash_T (M_1, \ldots, M_n)} \quad (\text{TupleI}) \]

\[ E \vdash_T (M_1, \ldots, M_n) \]
\[ \quad \frac{}{E \vdash_T M_i} \quad (\text{TupleE}_i), 1 \leq i \leq n \]

\[ E \vdash_T M \quad M \approx_T M' \]
\[ \quad \frac{}{E \vdash_T M'} \quad (T) \]
Finding Security Proofs Modulo $T$

- This is considerably harder, and a theme of active research. See [Kapur-Narendran-Wang, RTA’03], [Verma, RTA’03], [Cortier, RTA’03], [Verma, LPAR’03], [Chevalier-Küsters-Rusinowitch-Turuani, FST&TCS’03], [Verma, FST&TCS’04], etc.
Finding Security Proofs Modulo $T$

- This is considerably harder, and a theme of active research. See [Kapur-Narendran-Wang, RTA’03], [Verma, RTA’03], [Cortier, RTA’03], [Verma, LPAR’03], [Chevalier-Küsters-Rusinowitch-Turuani, FST&TCS’03], [Verma, FST&TCS’04], etc.
- When the number of sessions is bounded,
  - Insecurity with xor, or Abelian groups, is NP-complete [Chevalier-Rusinowitch-Turuani, LICS’03]; intruder deduction is in P.
  - With AC + homomorphism wrt. a hash, intruder deduction in P/NP (depending on coding) [Lafourcade-Treinen-et al.,’05].
  - With xor + homomorphism, or Abelian groups + homomorphism, intruder deduction in P [Delaune-Treinen-et al.,’06].
Outline

Why Cryptographic Protocols?
Cryptographic Protocols... and Attacks

Dolev-Yao Models
The Original Dolev-Yao Model (1983)
Dolev-Yao and First-Order Logic
Equational Theories

Other Applications

Spi-Calculus and Friends: Observational Equivalence
The Idea of Process Algebra
Are Dolev-Yao Models too Weak?
Observational Equivalence, Bisimulation
Relation to Computational Security

Conclusion

What About Quantum Protocols?
A Few Other Successes of Dolev-Yao Style Approaches

- [Pereira-Quisquater, CSFW’01] Group Diffie-Hellman is vulnerable for $n \geq 3$ participants, can be repaired for $n = 3$. (Uses modular exponentiation.)
A Few Other Successes of Dolev-Yao Style Approaches

- [Pereira, CSFW’04] There is no group key agreement protocol based solely on modular exponentiation for $n \geq 4$. 
A Few Other Successes of Dolev-Yao Style Approaches

- [Pereira, CSFW’04] There is no group key agreement protocol based solely on modular exponentiation for $n \geq 4$.
  (The sheer complexity of the protocol makes any by-hand analysis unfeasible.)
A Few Other Successes of Dolev-Yao Style Approaches

- [Pereira, CSFW’04] There is no group key agreement protocol based solely on modular exponentiation for $n \geq 4$.
A Few Other Successes of Dolev-Yao Style Approaches

- [Pereira, CSFW’04] There is no group key agreement protocol based solely on modular exponentiation for \( n \geq 4 \).
- Early inter-protocol flaws fixed in the Microsoft security architecture [Fournet, Microsoft Research,’04], using Blanchet’s tool ProVerif.
Outline

Why Cryptographic Protocols?
- Cryptographic Protocols... and Attacks

Dolev-Yao Models
- The Original Dolev-Yao Model (1983)
- Dolev-Yao and First-Order Logic
- Equational Theories
- Other Applications

Spi-Calculus and Friends: Observational Equivalence
- The Idea of Process Algebra
- Are Dolev-Yao Models too Weak?
- Observational Equivalence, Bisimulation
- Relation to Computational Security
- Conclusion

What About Quantum Protocols?
The Spi-Calculus

This is another modeling style. Program your protocols in some simple, core language. E.g., the spi-calculus [Abadi-Gordon, CCS’97].

- Expressions denote messages, as in Dolev-Yao:
  
  \[
  e, \ldots ::= X \quad \text{variables} \\
  \quad | \quad f(e_1, \ldots, e_n) \quad \text{application of constructor } f \\
  \quad | \quad \{ e_1 \} e_2 \quad \text{symmetric encryption} \\
  \quad | \quad [ e_1 ] e_2 \quad \text{asymmetric encryption}
  \]

- Processes: next slide.
Spi-Calculus Processes

- Processes are programs, or whole systems, $P, Q, R, \ldots ::=$

```
stop
\![X]P
P|Q
new X; P
out(e_1, e_2); P
in(e_1, X); P
let X = e in P
case e_1 of $f(X_1, \ldots, X_n) \Rightarrow P$ else Q
case e_1 of $\{X\}_{e_2} \Rightarrow P$ else Q
case e_1 of $[X]_{e_2} \Rightarrow P$ else Q
if $e_1 = e_2$ then $P$ else Q
f(e_1, \ldots, e_n)
```

- stop
- replication
- parallel composition
- fresh name creation
- writing to a channel
- reading from a channel
- local definition
- pattern-matching
- symmetric decryption
- asymmetric decryption
- equality test
- process call
Symmetric Key Needham-Schroeder in Spi

```
proc alice (to_s, from_s, 
    to_b, from_b, 
    A, B, Kas) =
  new Na;
  out (to_s, A, B, Na);
  in (from_s, 
      \{=Na, =B, Kab, M\} Kas);
  out (to_b, M);
  in (from_b, \{Nb\} Kab);
  out (to_b, \{s (Nb)\} Kab);

proc bob (to_a, from_a, Kbs) =
  in (from_a, \{Kab, A\} Kbs);
  new Nb;
  out (to_a, \{Nb\} Kab);
  in (from_a, \{=s (Nb)\} Kab);

proc server (from_a, to_a) =
  in (from_a, A, B, Na);
  new Kab;
  out (to_a, 
       \{Na, B, Kab, 
          \{Kab, A\} kxs(B)\} kxs(A));

proc main = new c_pub; new a; new b;
    new Kas; new Kbs;
  ![A] ![B]
    ( alice (c_pub, c_pub, 
            c_pub, c_pub, 
            A, B, kxs (A))
    | bob (c_pub, c_pub, 
          kxs (B)))
  | ![server (c_pub, c_pub) ];
```
You may translate spi-calculus processes to Horn clauses, losing some precision in passing. This is equivalent to a type system for security [Abadi-Blanchet, POPL’02]. (An undecidable one, by the way.)
Until Now, Nothing New Under the Sun

- You may translate spi-calculus processes to Horn clauses, losing some precision in passing. This is equivalent to a type system for security [Abadi-Blanchet, POPL’02]. (An undecidable one, by the way.)

- You may do the same by aiming at a decidable class. This is what [Nielson-Nielson-Seidl, SAS’02] do. The corresponding “type system” falls into $\mathcal{H}_1$, even in $\mathcal{H}_3$, a cubic-time class.
You may translate spi-calculus processes to Horn clauses, losing some precision in passing. This is equivalent to a type system for security [Abadi-Blanchet, POPL'02]. (An undecidable one, by the way.)

You may do the same by aiming at a decidable class. This is what [Nielson-Nielson-Seidl, SAS’02] do. The corresponding “type system” falls into $\mathcal{H}_1$, even in $\mathcal{H}_3$, a cubic-time class.

This all decides reachability properties, e.g., secrecy or authentication. In other words, this is Dolev-Yao in disguise.
Until Now, Nothing New Under the Sun

- You may translate spi-calculus processes to Horn clauses, losing some precision in passing. This is equivalent to a type system for security [Abadi-Blanchet, POPL’02]. (An undecidable one, by the way.)

- You may do the same by aiming at a decidable class. This is what [Nielson-Nielson-Seidl, SAS’02] do. The corresponding “type system” falls into $\mathcal{H}_1$, even in $\mathcal{H}_3$, a cubic-time class.

This all decides reachability properties, e.g., secrecy or authentication. In other words, this is Dolev-Yao in disguise. But one can do more with process algebra...
Outline

Why Cryptographic Protocols?
Cryptographic Protocols... and Attacks
Dolev-Yao Models
The Original Dolev-Yao Model (1983)
Dolev-Yao and First-Order Logic
Equational Theories
Other Applications

Spi-Calculus and Friends: Observational Equivalence
The Idea of Process Algebra
Are Dolev-Yao Models too Weak?
Observational Equivalence, Bisimulation
Relation to Computational Security
Conclusion

What About Quantum Protocols?
How Many Secret Bits?

Look at the \((\text{Tuple}l)\) rule:

\[
E \vdash M_1 \quad \ldots \quad E \vdash M_n
\]

\[
E \vdash (M_1, \ldots, M_n)
\]

This indicates that, to “know” a pair, you need to “know” each component.
How Many Secret Bits?

Look at the \((\text{Tuple}!)\) rule:

\[
\begin{align*}
E &\vdash M_1 \quad \ldots \quad E \vdash M_n \\
\hline
E &\vdash (M_1, \ldots, M_n)
\end{align*}
\]

This indicates that, to “know” a pair, you need to “know” each component.
In other words, a message is Dolev-Yao-secret if and only if the adversary cannot know all bits.
How Many Secret Bits?

Look at the (\textit{TupleI}) rule:

\[
E \vdash M_1 \quad \ldots \quad E \vdash M_n \\
\frac{}{E \vdash (M_1, \ldots, M_n)}
\]

This indicates that, to “know” a pair, you need to “know” each component.
In other words, a message is Dolev-Yao-secret if and only if the adversary cannot know all bits.
This looks ridiculous. In cryptography, \( M \) is secret if and only if the adversary cannot know more than a negligible proportion of the bits of \( M \).
Outline

Why Cryptographic Protocols?
- Cryptographic Protocols... and Attacks
Dolev-Yao Models
- The Original Dolev-Yao Model (1983)
- Dolev-Yao and First-Order Logic
- Equational Theories
- Other Applications

Spi-Calculus and Friends: Observational Equivalence
- The Idea of Process Algebra
- Are Dolev-Yao Models too Weak?
- Observational Equivalence, Bisimulation

Relation to Computational Security
Conclusion

What About Quantum Protocols?
May-Testing Equivalence

Say that two processes $P$, $Q$ are observationally equivalent $(P \simeq Q)$ iff

For every $Adv$, for every barb $\beta$, $(P|Adv) \Downarrow \beta$ iff $(Q|Adv) \Downarrow \beta$

More precisely, this is may-testing equivalence. A barb $\beta$ is a channel + a direction (input/output). $R \Downarrow \beta$ means that $R$ may eventually emit/receive on $\beta$. 
May-Testing Equivalence

Say that two processes $P$, $Q$ are observationally equivalent ($P \simeq Q$) iff

For every $Adv$, for every barb $\beta$, $(P | Adv) \Downarrow \beta$ iff $(Q | Adv) \Downarrow \beta$

More precisely, this is may-testing equivalence. A *barb* $\beta$ is a channel + a direction (input/output). $R \Downarrow \beta$ means that $R$ may eventually emit/receive on $\beta$.

In other words, $P \simeq Q$ iff no adversary $Adv$ can make any difference between $P$ and $Q$ by just looking at, and interfering with their input-output activity.
Strong Secrecy, Strong Authentication

Let $P[M]$ be a process with a distinguished occurrence of $M$. $M$ is strongly secret in $P$ iff

for every $M'$, $P[M] \simeq P[M']$.

In other words, no adversary can change behaviors as a result of observing a difference between $M$ and $M'$. 
Strong Secrecy, Strong Authentication

Let \( P[M] \) be a process with a distinguished occurrence of \( M \). \( M \) is strongly secret in \( P \) iff

\[
\text{for every } M', \ P[M] \simeq P[M'].
\]

In other words, no adversary can change behaviors as a result of observing a difference between \( M \) and \( M' \).

Let \( P[M, \text{in}(e, X); Q] \) be a process with a distinguished occurrence of a message \( M \) and a distinguished occurrence of a sub-process \( \text{in}(e, X); Q \). \( X \) is strongly authentically \( M \) iff

\[
P[M, \text{in}(e, X); Q] \simeq P[M, \text{in}(e, X); Q[X := M]]
\]

(The second process throws \( X \) away and “magically” gets \( M \).)
How Many Secret Bits (Again)?

One may argue that $M$ is strongly secret iff no predicate of $M$ can be used by $\text{Adv}$ to make a decision.
How Many Secret Bits (Again)?

One may argue that $M$ is strongly secret iff no predicate of $M$ can be used by $Adv$ to make a decision. In other words, the adversary knows strictly less than 1 bit of $M$. (Similarly for authentication.)
How Many Secret Bits (Again)?

One may argue that $M$ is strongly secret iff no predicate of $M$ can be used by $Adv$ to make a decision.

In other words, the adversary knows strictly less than 1 bit of $M$.

(Similarly for authentication.)

Paradoxically, this is sometimes too much. E.g., in a password-based authentication system, the adversary will know whether the typed password is valid or not.
One may argue that $M$ is strongly secret iff no predicate of $M$ can be used by $Adv$ to make a decision. In other words, the adversary knows strictly less than 1 bit of $M$. (Similarly for authentication.)

Paradoxically, this is sometimes too much. E.g., in a password-based authentication system, the adversary will know whether the typed password is valid or not.

Funnily, we threw Dolev-Yao out the door, and it will come back through the window.
What Are Bisimulations?

- A classic way to establish observational equivalence.
- Usually takes the form of:

  A bisimulation is a relation $\approx$ between processes such that

  \[
  \begin{align*}
  P & \approx P' \\
  Q & \approx Q'
  \end{align*}
  \begin{align*}
  P & \Rightarrow P' \\
  Q & \Rightarrow Q'
  \end{align*}
  \begin{align*}
  P & \approx P' \\
  Q & \approx Q'
  \end{align*}

  Usually, $\approx$ implies $\simeq$; with some luck, the converse holds.
What Are Bisimulations?

- A classic way to establish observational equivalence.
- Usually takes the form of:

A bisimulation is a relation $\approx$ between processes such that:

\[
\begin{align*}
P &\approx P' \\
Q &\approx Q'
\end{align*}
\]

Usually, $\approx$ implies $\simeq$; with some luck, the converse holds.

- Complex in the case of the spi-calculus: encryption, creation of fresh names complicate the picture.
What Are Bisimulations?

- A classic way to establish observational equivalence.
- Usually takes the form of:
  
  A bisimulation is a relation $\approx$ between processes such that

  \[
  P \approx P' \quad P \approx P' \\
  Q \approx Q' \quad Q \approx Q'
  \]

  Usually, $\approx$ implies $\equiv$; with some luck, the converse holds.

- Any non-trivial notion of observational equivalence is undecidable [Hüttel, INFINITY’02], even for finite-control spi processes. In particular, framed bisimulation [Abadi-Gordon, NJC’98], also [Boreale-de-Nicola-Pugliese, LICS’99].
Hedged Bisimulation
[Borgström-Nestmann,AMAST’04]

Refinement of [Boreale-de-Nicola-Pugliese, LICS’99], characterizes observational equivalence. On terms:

- **Hedges** $th$ are finite sets of pairs $(M_1, M_2)$ in each world that are thought to be indistinguishable.

\[
(M_1, M_2) \in th \quad th \vdash M_1 \approx M_2 \quad th \vdash K_1 \approx K_2
\]

\[
\vdash \{M_1\}_{K_1} \approx \{M_2\}_{K_2}
\]
Hedged Bisimulation
[Borgström-Nestmann, AMAST’04]

Refinement of [Boreale-de-Nicola-Pugliese, LICS’99], characterizes observational equivalence. On terms:

- **Hedges** $th$ are finite sets of pairs $(M_1, M_2)$ in each world that are thought to be indistinguishable.

$$
(M_1, M_2) \in th \quad th \vdash M_1 \approx M_2 \\
\{ M_1 \}_{K_1} \approx \{ M_2 \}_{K_2}
$$

- A hedge is **consistent** iff relates names with names, encryptions with encryptions, is a partial bijection, and

$$
K_1 \not\in \text{dom } th \quad K_2 \not\in \text{codom } th
$$
Hedged Bisimulation
[Borgström-Nestmann, AMAST’04]

Refinement of [Boreale-de-Nicola-Pugliese, LICS’99], characterizes observational equivalence. On terms:

- It was observed in [JGL-Lasota-Nowak-Zhang, CSL’04] that it is equivalent to throw $th$ away, and define $\vdash \_ \approx \_$ directly, subject to the proviso that:

  \[
  \begin{align*}
  & \vdash M_1 \approx M_2 \quad \vdash K_1 \approx K_2 \\
  \therefore & \vdash \{M_1\}_{K_1} \approx \{M_2\}_{K_2}
  \end{align*}
  \]

  \[
  \begin{align*}
  & \vdash M_1 \approx M_2 \quad \vdash K_1 \approx K_2 \\
  \therefore & \vdash \text{dec}(M_1, K_1) \approx \text{dec}(M_2, K_2)
  \end{align*}
  \]

where $\text{dec}({M\}_K, K) = M$, $\text{dec}(M, K) = \bot$ otherwise.
Hedged Bisimulation

[Borgström-Nestmann, AMAST’04]

Refinement of [Boreale-de-Nicola-Pugliese, LICS’99], characterizes observational equivalence. On terms:

- It was observed in [JGL-Lasota-Nowak-Zhang, CSL’04] that it is equivalent to throw $th$ away, and define $\vdash _ \approx$ directly, subject to the proviso that:

\[
\begin{align*}
    & th \vdash M_1 \approx M_2 & th \vdash K_1 \approx K_2 \\
    \Rightarrow & th \vdash \{M_1\}_{K_1} \approx \{M_2\}_{K_2} & \quad th \vdash M_1 \approx M_2 & th \vdash K_1 \approx K_2 \\
    & \quad th \vdash \text{dec}(M_1, K_1) \approx \text{dec}(M_2, K_2)
\end{align*}
\]

where $\text{dec}(\{M\}_{K}, K) = M$, $\text{dec}(M, K) = \bot$ otherwise. Dolev-Yao strikes again... in binary form.
Hedged Bisimulation Per Se

Definition omitted: rather horrible, see [Borgström-Nestmann, AMAST’04], Definition 15.

- Quantifies universally over extensions of $h \vdash \_ \approx \_$ on receiving message;
- Quantifies existentially over extensions relating sent messages $M_1$ and $M_2$ on message sends.
Hedged Bisimulation Per Se

- Definition omitted: rather horrible, see [Borgström-Nestmann, AMAST’04], Definition 15.
  - Quantifies universally over extensions of $h \vdash _- \approx _-$ on receiving message;
  - Quantifies existentially over extensions relating sent messages $M_1$ and $M_2$ on message sends.

- Despite undecidability, you can extract sound reasoning principles, see [Boreale-Gorla, JTIT’02], and next slide. Implemented by Boreale and Gorla.
Sound Reasoning Principles

\[
P \equiv Q \\
\frac{}{(\sigma, \sigma) \vdash P \approx Q} \\
\frac{\vdash C[P + \text{if } M = M \text{ then } Q] \approx C[P + Q]}{M \text{ not a name bound in } \text{new } n; C[\bullet] \\
\frac{\vdash \text{new } n; C[P + \text{if } n = M \text{ then } Q] \approx \text{new } n; C[P]}{\text{etc.}}}
\]
Bisimulations and Biproceses

- An old idea, well-known to category-theorists [Joyal-Nielsen-Winskel, I&C’96]:

A bisimulation between two transitions systems $\delta_1$ on state set $Q_1$ and $\delta_2$ on $Q_2$ is a transition system $\delta$ on $Q_1 \times Q_2$ such that $\pi_1(\delta) = \delta_1$ and $\pi_2(\delta) = \delta_2$. 
Bisimulations and Biprocesses

- An old idea, well-known to category-theorists [Joyal-Nielsen-Winskel, I&C’96]:
  A bisimulation between two transitions systems $\delta_1$ on state set $Q_1$ and $\delta_2$ on $Q_2$ is a transition system $\delta$ on $Q_1 \times Q_2$ such that $\pi_1(\delta) = \delta_1$ and $\pi_2(\delta) = \delta_2$.

- Can give concrete representations of $\delta$ when $\delta_1$ and $\delta_2$ arise from processes with same control (only messages change), see [Abadi-Blanchet-Fournet, LICS’05], idea from [Simonet, POPL’03?].
Bisimulations and Biproceses

- An old idea, well-known to category-theorists [Joyal-Nielsen-Winskel, I&C’96]:
  A bisimulation between two transitions systems $\delta_1$ on state set $Q_1$ and $\delta_2$ on $Q_2$ is a transition system $\delta$ on $Q_1 \times Q_2$ such that $\pi_1(\delta) = \delta_1$ and $\pi_2(\delta) = \delta_2$.

- Can give concrete representations of $\delta$ when $\delta_1$ and $\delta_2$ arise from processes with same control (only messages change), see [Abadi-Blanchet-Fournet, LICS’05], idea from [Simonet, POPL’03?]
  E.g., share $\mathbf{out}(e_1, e_2)$; and $\mathbf{out}(e'_1, e'_2)$; as $\mathbf{outC}(\mathbf{diff}(e_1, e'_1), \mathbf{diff}(e_2, e'_2))$
Security against Off-Line Guessing Attacks

Hmm, we probably won’t have time for this. See [Baudet, SSP’05].
Idea [Lowe] is that some protocols are secure, although secret $M$ is encrypted with a weak secret $K$ (e.g., a password). You find $K$ by enumeration, but cannot test whether $K$ is the right one if you cannot recognize the right $M$.
Uses equational theories in an essential way.
Computational Notions of Security

- Familiar to cryptographers.
- **Reduce** security of protocol to security of basic cryptographic primitives, quantifying the probabilistic advantage that this gives to the adversary.

This seems very different from what we did above. But...
Relating Computational and Formal Security

We can in fact relate computational security with formal methods.

- A very trendy topic these days.
We can in fact relate computational security with formal methods.

- A very trendy topic these days.
- Pioneering papers: [Abadi-Rogaway, IFIP-TCS’00] (using patterns—with a Dolev-Yao twist!), [Warinschi, CSFW’03] (comp. proof of Needham-Schroeder-Lowe), [Micciancio-Warinschi, JCS’04], etc.
Relating Computational and Formal Security

We can in fact relate computational security with formal methods.

- A very trendy topic these days.
- Pioneering papers: [Abadi-Rogaway, IFIP-TCS’00] (using patterns—with a Dolev-Yao twist!), [Warinschi, CSFW’03] (comp. proof of Needham-Schroeder-Lowe), [Micciancio-Warinschi, JCS’04], etc.
- with a Las Vegas model of computation, purely reduces to Dolev-Yao [Degano-Zunino, FoSSaCs’04; Baudet, JALC’05], under mild assumptions;
- FormaCrypt project: Blanchet, Pointcheval, Baudet, JGL, Cortier, Abadi, Kremer, Warinschi, Lubicz, just started.
Relating Computational and Formal Security

We can in fact relate computational security with formal methods.

- A very trendy topic these days.
- Pioneering papers: [Abadi-Rogaway, IFIP-TCS’00] (using patterns—with a Dolev-Yao twist!), [Warinschi, CSFW’03] (comp. proof of Needham-Schroeder-Lowe), [Micciancio-Warinschi, JCS’04], etc.
- with a Las Vegas model of computation, purely reduces to Dolev-Yao [Degano-Zunino, FoSSaCs’04; Baudet, JALC’05], under mild assumptions;
- FormaCrypt project: Blanchet, Pointcheval, Baudet, JGL, Cortier, Abadi, Kremer, Warinschi, Lubicz, just started.
- Ask me, or Mathieu Baudet, his 72-slide presentation on
Outline

Why Cryptographic Protocols?
Cryptographic Protocols... and Attacks
Dolev-Yao Models
The Original Dolev-Yao Model (1983)
Dolev-Yao and First-Order Logic
Equational Theories
Other Applications
Spi-Calculus and Friends: Observational Equivalence
The Idea of Process Algebra
Are Dolev-Yao Models too Weak?
Observational Equivalence, Bisimulation
Relation to Computational Security
Conclusion
What About Quantum Protocols?
What Can We Do Here?

Two ideas I have started but not really developed yet:

- Use linear logic. $\text{knows}(M_1) \& \text{knows}(M_2)$ means that adversary knows $\alpha|M_1\rangle + \beta|M_2\rangle$. 
What Can We Do Here?

Two ideas I have started but not really developed yet:

- **Use linear logic.** $\text{knows}(M_1) \& \text{knows}(M_2)$ means that adversary knows $\alpha | M_1 \rangle + \beta | M_2 \rangle$.
  
  Good match with the rules of logic: adversary can then, say, use $\text{knows}(M_1)$ to pursue an attack, but will lose $\text{knows}(M_2)$ ($\approx$ measurement). (Ask me the draft... )
What Can We Do Here?

Two ideas I have started but not really developed yet:

- Use linear logic. $\text{knows}(M_1) \& \text{knows}(M_2)$ means that adversary knows $\alpha |M_1\rangle + \beta |M_2\rangle$.

  Good match with the rules of logic: adversary can then, say, use $\text{knows}(M_1)$ to pursue an attack, but will lose $\text{knows}(M_2)$ ($\approx$ measurement). (Ask me the draft...)

Unfortunately, since probabilities are lost (as in methods above), no hope of proving [BB84] or [Ekert91], say.
What Can We Do Here?

Two ideas I have started but not really developed yet:

- Use **linear logic**. \( \text{knows}(M_1) \& \text{knows}(M_2) \) means that adversary knows \( \alpha | M_1 \rangle + \beta | M_2 \rangle \).

  Good match with the rules of logic: adversary can then, say, use \( \text{knows}(M_1) \) to pursue an attack, but will lose \( \text{knows}(M_2) \) (\( \approx \) measurement). (Ask me the draft...)

  Unfortunately, since probabilities are lost (as in methods above), no hope of proving [BB84] or [Ekert91], say.

- Use classical logic but with terms modulo the theory of **density matrices**. I.e., pursue the equational theory theme. Less exciting, but might work better.