The Needham-Schroeder public key protocol

\[ \{A, N_A\}_{pk_B} \]

\[ \{N_A, N_B\}_{pk_A} \]

\[ \{N_B\}_{pk_B} \]

- Nonce \( N_B \) sent in the second message:
  - is intended for \( A \) (identity received in the first message)
  - should be secret to any other party but \( A \)
- \( A \) and \( B \) should have matching conversations
The Needham-Schroeder public key protocol

- $N_B$ is secret if the adversary is passive
- $N_B$ is not secret if the adversary is active
- Matching conversations does not hold
Lowe's fix - Secure Version of NS

\[ \{A, N_A\}_{pk_B} \overset{\rightarrow}{\rightarrow} \{B, N_A, N_B\}_{pk_A} \overset{\leftarrow}{\leftarrow} \{N_B\}_{pk_B} \]

No more “logical” attacks; protocol secure
... or is it?

Implement the protocol with an (IND-CPA) secure encryption scheme “

\[
\text{Adv ("", A) (\xi) = Pr[(pk, sk) \xleftarrow{\text{K(\xi): A}} \in 1]} - \Pr[(pk, sk) \xleftarrow{\text{K(\xi): A}} \in 1]
\]
Another gap

- There exist IND-CPA secure encryption scheme and a deterministic polynomial time algorithm such that

\[ E(pk_A, (B, N_A N_B)) \quad \text{and} \quad E(pk_A, (C, N_A N_B)) \]
An attack against an implementation of NSL

- $N_B$ may not be secret even if encryption is IND-CPA
- Matching conversations does not hold
- ... use stronger encryption
IND-CCA security for multi-users

- Implement encryption with a scheme $(K,E,D)$ that is IND-CCA secure

$$E_{pk_1}(LR(b,\ldots))$$
$$E_{pk_2}(LR(b,\ldots))$$
$$D_{sk_1}(\cdot)$$
$$D_{sk_2}(\cdot)$$

$m_0, m_1$

$b$ is…
...back to NSL

• If NSL is implemented with an encryption scheme that is IND-CCA secure then:
  - $N_B$ is secret
  - Matching conversations holds
A gap

• Security of primitives is
  - *axiomatized* (in the symbolic approach)
  - *defined* (in the computational approach)

• Question:
  - Symbolically: not possible to calculate \(\{C,N_A,N_B\}_{pk_A}\) out of \(\{B,N_A,N_B\}_{pk_A}\)
  - Computationally: is it possible to enforce the above?
Computational soundness

• The goal is to find sufficient security conditions on the primitives used in the implementation such that a protocol secure in the symbolic setting is also secure in the computational setting...

• ...but what is a protocol, what does secure mean?
Protocols

- A sequence of message exchanges
- Messages constructed from constants, variables, and cryptographic operations

Send \{A,N_A\}_{p_{kB}}
Receive \{B,N_A,X\}_{p_{kA}}
Send \{X\}_{p_{kB}}

Receive \{A,Y\}_{p_{kB}}
Send \{B,Y,N_B\}_{p_{kA}}
Receive \{N_B\}_{p_{kB}}
Communication is over a network

THE
INTERNET
(Generic) Execution model
Symbolic execution model

- Messages exchanged during the execution are terms
- Cryptographic operations are operations on terms
- The adversary is a Dolev-Yao adversary who operates with a finite, well determined number of rules
Computational execution model

- Messages exchanged during the execution are bitstrings
- Cryptography implemented with actual (randomized) algorithms
- The adversary is an arbitrary randomized polynomial time algorithm
Back to the gap

- Security properties are statements about two very different executions
  - Non-deterministic executions (symbolically)
  - Randomized executions (computationally)
Computational soundness via black-box reactive simulation
The simulation approach
[Backes, Pfitzmann, Waidner]

CRYPTOGRAPHIC LIBRARY

Nonce generation, Encryption, Decryption, Signing, MACs, etc
The simulation approach
[Backes, Pfitzmann, Waidner]

**SYMBOLIC CRYPTOGRAPHIC LIBRARY**

Internally the library operates with terms and enforces Dolev-Yao behaviours

**COMPUTATIONAL CRYPTOGRAPHIC LIBRARY**

Internally the library operates with bitstrings and actual cryptographic algorithms
THEOREM: If cryptographic primitives are secure in the computational cryptographic library, then there exists a simulator such that no probabilistic polynomial time environment can distinguish between the two worlds.
CRYPTOGRAPHIC LIBRARY
Protocol execution with a cryptographic library

CRYPTOGRAPHIC LIBRARY
Protocol execution with a cryptographic library

CRYPTOGRAPHIC LIBRARY

ENVIRONMENT
SYMBOLIC CRYPTOGRAPHIC LIBRARY

SIMULATOR

ENVIRONMENT
Soundness with a cryptographic library

- Security of protocols can be analyzed in a world where cryptography is idealized in the Dolev-Yao style
Computational soundness via trace mapping
Trace mapping

[Micciancio, Warinschi]
The trace mapping approach

Symbolic execution of a protocol

Real execution of a protocol
A bit more precisely

• The adversary may be able to corrupt parties

• The adversary may send any message it wants to a session and receives the answer calculated by the session
Execution traces

Execution trace:

Formally

\[ F_0 \xrightarrow{m_0} F_1 \xrightarrow{m_1} F_2 \xrightarrow{m_2} F_3 \xrightarrow{m_3} \ldots \]

\[ F_i : \text{Local variables of sessions} \to \text{Values} \]
Symbolic executions

- Messages, values etc... are terms

\[ F_i : \text{Local variables of sessions} \rightarrow \text{Terms} \]

- Adversary can only send messages that he can compute according to the Dolev Yao rules

- Nondeterministic executions

- For protocol " and adversary A, write \( Tr_s(\ldots, A) \) for the trace determined by A
Computational executions

- Messages, values etc... are bitstrings
  \[ G_i : \text{Local variables of sessions} \rightarrow \text{Bitstrings} \]

- Adversary can only send any polynomial-time computable message

- Executions are randomized

- \[ \text{Tr}_c \left( (R_A),A(R_A) \right) \] is the execution trace determined by adversary A, randomness \( R_A \) and \( R_A \)
Computational soundness result

\[ F_0 \xrightarrow{m_0} F_1 \xrightarrow{m_1} F_2 \xrightarrow{m_2} F_3 \xrightarrow{m_3} \ldots \]

\[ f_c \quad f_c \quad f_c \quad f_c \quad f_c \quad f_c \quad f_c \quad f_c \quad f_c \quad \ldots \]

\[ G_0 \xrightarrow{m_0} G_1 \xrightarrow{m_1} G_2 \xrightarrow{m_2} G_3 \xrightarrow{m_3} \ldots \]

- **“Mapping lemma”:** With overwhelming probability the computational trace is the image of a Dolev-Yao trace through an appropriate mapping \( f_c \).

- **Interpretation:** The real adversary only performs Dolev Yao operations!!!
Let $\Pi$ be a protocol and $A$ a computational adversary. If $\Pi$ is implemented with secure primitives then almost all of the computational traces of $\Pi$ are images of symbolic Dolev-Yao traces.

$$\text{Prob}[ < B, < f_c : \text{Tr}_c(\Pi, A(R_A)) = f_c(\text{Tr}_s(\Pi, B)) ]$$

is overwhelming
Proof idea

1. Fix an adversary $A$

2. Any concrete execution can be mapped to a symbolic execution

3. Show that this symbolic execution is that of a Dolev-Yao adversary (with overwhelming probability)

... or otherwise one can use $A$ to break the underlying primitives
Step 2: From concrete executions...
Send $\{A,N_A\}_{pk_B}$
Receive $\{B,N_A,X\}_{pk_A}$
Send $\{X\}_{pk_B}$
Receive $\{N_B\}_{pk_B}$

public executions

Send $\{A,NA\}_{pk_B}$
Receive $\{A,Y\}_{pk_B}$
Send $\{B,Y,N_B\}_{pk_A}$
Receive $\{N_B\}_{pk_B}$
Step 3: The symbolic trace is Dolev-Yao

Given an adversary that produces traces that are not Dolev-Yao, use that adversary to break the security of the basic primitive(s)
Select n0, n1 random nonces (n0, n1)

If n = n0 then output 0
else output 1

\( E_{pk(a)}(LR(b,..)) \)
\( E_{pk(b)}(LR(b,..)) \)
\( D_{k_1}(.) \)
\( D_{k_2}(.) \)
Trace mapping lemma

• Let $\Pi$ be a protocol and $A$ a computational adversary. If $\Pi$ is implemented with secure primitives then almost all of the computational traces of $\Pi$ are images of symbolic Dolev-Yao traces.

\[ \text{Prob}[\exists B, \exists f_c : \text{Tr}_c(\Pi(R_\Pi), A(R_A)) = f_c(\text{Tr}_s(\Pi, B))] \] is overwhelming
Computational soundness for trace properties
A security property is a predicate on the set of possible traces

E.g.: Matching conversations: every session of user B (with A) that finishes successfully has a matching session of user A
Security Properties - symbolically

- Protocol $\Pi$ satisfies security property $P_s$ ($\Pi \models P_s$) iff $(;A) Tr_s (\Pi, A) \in P_s$
Security Properties - symbolically

- Protocol $\pi$ satisfies security property $P_s$ ($\pi \models_s P_s$) iff $(;A) Tr_s (\pi,A) \in P_s$
Security Properties - computationally

• Protocol \( \Pi \) satisfies computationally property \( P_c \):

\[ 
\Pi \models \text{iff } \Pr \left[ \text{Tr}_c \left( \Pi, (R, A(R_A)) \right) \in P_c \right]
\]

is overwhelming
Security Properties - computationally

- Protocol \( \Pi \) satisfies computationally property \( P_c \):
  \[
  \Pi \models_{c} P_c \iff \Pr \left( \text{Tr}_c(\Pi(R), A(R_A)) \in P_c \right)
  \]
  is overwhelming
Translation of trace properties

Let $P_s$ be a symbolic security property and let $P_c = \pi(P_s) = \bigcup f(P_s)$ (the union is after all appropriate mappings $f$). If the mapping lemma holds then:

**THEOREM:** Let $\Pi$ be a protocol. Then:

\[ \Pi \models s P_s, \quad \Pi \models c P_c \]
Proof

Let \( \Pi \) be a protocol and \( A \) a computational adversary. Pick \( R_\Pi \) and \( R_A \). Then (with overwhelming probability):

\[
\exists f \quad (\exists B) \quad \text{Tr}_s(\Pi, B) > f \quad \text{Tr}_c(\Pi, A(R_A))
\]
Soundness for secrecy properties
Soundness for secrecy
[Cortier, Warinschi]

- For the Needham Schroder Lowe protocol:

  NSL $\vdash_s \text{Secret}(N_B)$
  
  For any session $t$ of $B$ with an honest party $A$, the nonce $n^t$ that instantiates $N_B$ in session $t$ is never sent by the adversary in clear over the network.

  Send $\{A, NA\}_{pk_B}$
  Receive $\{B, NA, X\}_{pk_A}$
  Send $\{X\}_{pk_B}$
  Receive $\{A, Y\}_{pk_B}$
  Send $\{B, Y, N_B\}_{pk_A}$
  Receive $\{N_B\}_{pk_B}$
Soundness for secrecy

• The mapping lemma implies a notion of computational secrecy:

• (With overwhelming probability) the adversary cannot output any of the nonces that instantiate variable $N_B$ in sessions of B with honest A

• ...but this security notion - onewayness - is cryptographically unsatisfying
Computational secrecy

• Computational secrecy for nonce N in session t: prior to the execution select $n_0$, $n_1$. Run the protocol with $n_b$ as value for $N_B$ in session t. Give $n_0, n_1$ to the adversary and ask him to guess $b$.

• $\text{NSL}_c \text{Secret}(N)$ if N is computationally secret in any session of B with an honest party.
Soundness for secrecy

• For any protocol \(\Pi\) implemented with secure primitives (digital signatures, public key encryption, nonces)

\[\Pi \models_{s} \text{Secret}(N), \quad \Pi \models_{c} \text{Secret}(N)\]

• The proof relies on the computational adversary to only perform Dolev-Yao operations
Soundness for hash functions
Hash functions
[Cortier, Kremer, Küsters, Warinschi]

- The trace mapping lemma holds if hash functions are implemented by random oracles
  - Hash values can be interpreted as symbolic terms by observing the communication with the random oracle
- ... soundness holds for trace properties
- How about secrecy?
Soundness for secrecy does not hold anymore

- Consider a protocol where A sends to B the message $h(N_A)$, where $N_A$ is a random nonce. Then
  - $\Pi \models_{s} \text{Secret}(N_A)$ is true
  - $\Pi \models_{c} \text{Secret}(N_A)$ is not true

  Since given $h(n_b)$, $n_0$, $n_1$ the adversary can easily recover $b$
...but it can be recovered

- Define the pattern that the adversary can observe when given N. In particular:
  - $\text{pattern}_N(\{N\}_pk) = \Box_{pk}$
  - $\text{pattern}_N(\text{h}(N)) = \text{h}(N)$
  - $\text{pattern}_N(\text{h}(N')) = \text{h}(\Box)$
Stronger notion of secrecy

• Stronger notion of secrecy for nonces:

" $\Pi_s S\text{Secret}(N)$ if for any instantiation $n^t$ of nonce $N$ and for any adversary $A$, $n^t$ does not occur in pattern $n^t(\text{Tr}_s(\Pi, A))$

• Computational soundness for secrecy holds:

" $\Pi_s S\text{Secret}(N)$, " $\Pi_c \text{Secret}(N)$
Additional results
Non-interactive zero-knowledge
[Backes,Unruh]

• Consider a specification language for protocols where non-interactive ZK statements can be used

• Identify the requirements needed to ensure that a mapping lemma holds

  (unpredictable non-interactive multi-theorem adaptive extraction zero-knowledge argument of knowledge with deterministic verification and extraction)

  - Extractability
  - Non-malleability
  - Unpredictability
Computational soundness for a process calculus
[Cortier, Comon-Lundh]

• Protocols written in a subset of applied π-calculus
  - Use symmetric key-encryption
• Define symbolic and computational executions for processes
• Soundness of observational equivalence: processes indistinguishable, symbolically, are indistinguishable by a computational attacker.
Commitment schemes

[Galindo, Garcia, van Rosum]

• Soundness for non-malleable commitments
• Commitments are similar to encryption
Some observations
Extractability

• Needed for interpreting uniquely each bitstring as a term

• Is ensured by either cryptographic security (e.g. integrity of encryption, collision resistance for hashes, extractability for ZK, message revealing signatures), extra randomization, and/or tagging of messages with types
Executability (simulatability)

- Needed to ensure that the execution of the protocol can be simulated for the adversary
- Identify appropriate restrictions on the protocols to ensure execution is possible (at the very least “normal” executability but possibly more)
Non-malleability

- Usually symbolic axiomatization implies non-malleability
- The Lowe-type attack on the NS implementation with IND-CPA scheme is permitted by non-malleability
- Seems to be a (the) useful property (soundness for non-malleable commitments and ZK)
Some future directions

• Compositional soundness results

• Convincing applications

• Relevance to actual implementations
Thank you.