A Computational Interpretation of
Dolev-Yao Adversaries

Jonathan Herzog
Massachusetts Institute of Technology
How to Analyze Cryptographic Protocols

Many well-known methods based on formal methods
- Strand Spaces, model checkers, theorem provers, etc.
- Most based on common set of assumptions (Dolev-Yao model)

However, formal approach not the only framework for protocol analysis
“Computational” approach to cryptography comes from complexity theory
- Reflects character of that field

Both have advantages; how do they relate?
This talk: relating the two models’ adversaries
Related Work

This work greatly inspired by Abadi and Rogaway, A. Jürgens (2001)
Other approaches to (directly) relating Dolev-Yao adversary and underlying encryption (Guttman, Thayer, Zuck 2002)
Incorporating poly-time indistinguishability into process calculus (Mitchell et al. 1998, 1999, 2001)
Dolev-Yao model as universally composable reactive machine (Backes, Pfitzmann, Waidner 2003)
Everything is a bit-string
  – Messages, keys, numbers, etc.
Everything is an algorithm
  – Turing machine, sometimes with oracles
  – Key generation, encryption operation, adversary, etc.
(Almost) all security properties are asymptotic probabilities
  – Behavior of system as key-lengths grow
All proofs are by reduction
  – “If some adversary can break by encryption scheme, then some hard problem must be easy.”
  – Factoring, discrete log, quadratic residuosity, etc.
Definition of “semantic security” for public-key encryption:

‘The probability is negligible (in the asymptotic case) that an efficient (PPT) adversary can tell if you encrypted message $m_1$ or message $m_2$ (under a randomly picked key) even if the adversary gets to pick $m_1$ and $m_2$.’

$$\forall \text{Adv}_{PPT}, \forall \text{polynomials } q, \exists \eta_0. \forall \eta \geq \eta_0 \Pr[ (pk, sk) \leftarrow \text{KeyGen} (1^n); m_1, m_2 \leftarrow \text{Adv} (1^n, pk); b \leftarrow \text{CoinFlip} (1, 2); c \leftarrow \text{Enc} (m_b, pk); g \leftarrow \text{Adv} (c, m_1, m_2, pk): b = g ] \leq \frac{1}{2} + \frac{1}{q(\eta)}$$
Semantic Security, cont.

Semantic security (and more!) achieved by common schemes
   – Weak definition of security
Possible to instantiate given any hard problem*
Implies randomization of encryption
   – Given plaintext/key can produce many possible ciphertexts

Important note: security against *all* efficient adversaries

*Trapdoor permutation
Computational Approach: Pros and Cons

Constructions, theorems very concrete and meaningful
Approach applicable to many type of problems
- Not just primitives, protocols
- Very rich body of work
Proofs very labor-intensive, however
- New proof for each problem
- No automation
Dolev-Yao Model

Much higher-level model
- Focused on protocol analysis
Messages are elements of abstract algebra
- Assumed to have several nice properties
Honest participants assumed to be communicating state machines
- Transmitting and receiving abstract messages
Generally no randomness
Adversary assumed from restricted class of state machines
Dolev-Yao Adversary

Presumed to start knowing some initial set of messages
- All public/predictable values

Intercepts every message sent by honest participant

Can perform *only* the following operations:
- Encrypt known message with public key
- Decrypt known message with known private key
- Separate a known pair
- Pair two known messages
- Generate new nonces, keys

Can perform finite number of such operations before sending known message to any honest participant
Dolev-Yao Adversary, Formalized

Adversary trace: sequence of (adversary) queries, (participant) responses

\[ R_0 \ Q_1 \ R_1 \ \ldots \ Q_{n-1} \ R_{n-1} \ Q_n \ R_n \]

Each query \( Q_i \) must be derivable from initial adversary knowledge \((S_0)\), all previous responses

If \( S \) is a set of messages, \( C[S] \) is the smallest set so that

- \( S \subseteq C[S] \),
- \( C[S] \) is closed under pairing, separation
- \( C[S] \) is closed under decryption with keys in \( C[S] \), encryption with any key*

\( Q_i \) is derivable at time \( i \) iff:

\[ Q_i \in C[S_0 \cup \{R_0, R_1, \ldots R_{i-1}\}] \]

*Public-key setting
Generalizing Dolev-Yao Adversary

Fundamental assumption of Dolev-Yao model:

If Dolev-Yao adversary "knows" only messages in set $S$, cannot make messages outside of $C[S]$

“How can you justify this?”

– How can we say that adversary has no other abilities?

Answer: adversary has no other abilities in computational model, either (if underlying encryption is sufficiently strong)

Point of this talk: show computational cryptography can limit computational adversary to Dolev-Yao assumption

But first, a technical detail
Message Encoding

Must first bring formal messages down into the world of bit-strings.

Use both $\eta$, security parameter, and arbitrary computational encryption scheme.

Let $\llbracket M \rrbracket_\eta$ be the encoding of $M$:

- Public-key analogue to encoding function of Abadi and Rogaway
- Nonces, keys, encoded as random strings
- Encoding of pairs straightforward
- Encoding of $\{ \llbracket M \rrbracket_K \}$ uses $\llbracket M \rrbracket_\eta$, $\llbracket K \rrbracket_\eta$, and arbitrary computational encryption scheme
  - May be randomized
- Function from messages to distributions on bit-strings
Translating the Dolev-Yao Assumption

First step of enforcing Dolev-Yao assumption: translating it into computational terms

Recall that we want:

If an adversary “knows” a set $S$ of messages, cannot make a message outside $C[S]$.

An encryption operation is ideal if, when used in $[\cdot]_\eta$

$$\forall \text{Adv}_{PPT}, \forall \text{sets of messages } S, \forall M \notin C[S], \forall \text{polynomials } q, \exists \eta_0. \forall \eta \geq \eta_0, \Pr[s \leftarrow [S]_\eta; m \leftarrow \text{Adv}(1^\eta, s) : m \in [M]_{\eta^*}] \leq 2^{-\eta}.$$
Implementing Ideal Encryption

Definition of ideal cryptography useless unless it can be achieved
Need strong cryptography
A computational encryption scheme is plaintext-aware if
1. It is semantically secure, and
2. The adversary must know the plaintext to every ciphertext it creates itself.

Thm: Let Enc be plaintext-aware. Then Enc is ideal.

Proof Overview:
1. Assume otherwise. Then adversary can create encoding of some $M \notin C[S]$.
2. Plaintext awareness implies adversary can create encoding of some atomic subterm not in $C[S]$.
3. Doing so violates semantic security.
Proof Sketch (1/4)

Suppose adversary can create encoding of $M \not\in C[S]$. Look at interior node of $M$'s parse tree. If both children in $C[S]$, then node itself in $C[S]$. Why? Membership in $C[S]$ closed under encryption, pairing.
Proof Sketch (2/4)

If all paths from root to leaves go through $C[S]$, then root is in $C[S]$. Know this is not the case. Hence, at least one path does not go through $C[S]$. 

$\{N_1 N_2\}_K$
Proof Sketch (3/4)

Along this path: If adversary can create parent, can create both children. Easy to undo pair operation. If adversary can create ciphertext, can create plaintext. (Thanks to plaintext-awareness.) Hence, adversary can create some leaf not in $C[S]$. 

\[
\{N_1 N_2\}_K
\]

\[
N_1 \quad N_2 \quad K^*
\]

\[
N_1 \quad \quad \quad N_2
\]
Proof Sketch (4/4)

Adversary can create encoding of some atomic $M' \notin C[S]

$M'$ cannot be public information

- Must be nonce or private key

Two cases:

Case 1: $M'$ is component of* something in $C[S]$

- Example: $M' = N_1 \notin C[S]$, and $\{N_1\}_K \in C[S]$

- Example: $M' = K^{-1} \notin C[S]$, and $K \in C[S]$

- These are the only possibilities

- Hence, adversary able to break semantic security of encryption scheme

Case 2: $M'$ not component of something in $C[S]$

- Then adversary able to guess random nonce/key from independent values

*It, or inverse,
in parse tree of
Conclusions

Ideal encryption can be instantiated. This means...
Possible to limit computational adversary to attacks available to formal adversary. This means...
We have a rebuttal to Dolev-Yao naysayers

Details, full proofs to be found in my thesis:
http://theory.lcs.mit.edu/~jherzog
(Slides there also)
Email address in proceedings
Future work

Seek stronger result
  – Stronger statement of Dolev-Yao assumption
  – Weaker assumptions* on computational encryption
Result only covers case where adversary *tries* to create valid encoding
  – What if adversary intentionally sends garbage?
  – Honest participants might reveal information through error behavior
Better formalism of adversary-chosen nonces, keys
Extend result to other primitives
  – Signatures, hashing, symmetric encryption, etc.

*Plaintext awareness somewhat controversial, requires strong assumptions
Computational Encryption

Computational encryption schemes actually a triple of algorithms:

- \texttt{KeyGen} : Parameter $\rightarrow$ PublicKey $\times$ PrivateKey
  - The (randomized) key generation algorithm

- \texttt{Enc} : PublicKey $\times$ String $\rightarrow$ Ciphertext
  - The (randomized) encryption algorithm

- \texttt{Dec} : Ciphertext $\times$ PrivateKey $\rightarrow$ String
  - The (deterministic) decryption algorithm

Need that if $(pk, sk) \leftarrow \text{KeyGen}(1^n)$, then for all $m$, it is always true that

$$\text{Dec}(\text{Enc}(pk, m), sk) = m$$
Plaintext Awareness (Abr.)

Let $RO(\cdot)$ be the *random oracle*

- Provides randomly chosen function from $\{0, 1\}^*$ to
  (for some reasonable $n$)

Let $Enc$, adversary have access to $RO$

Encryption is *plaintext-aware* if for any adversary

$$c \leftarrow Adv(pk)$$

there exists an efficient *extractor* $K_{Adv}$ such that

$$Dec(sk, c) = K_{Adv}(c, pk, H)$$

where $H$ is a list of the queries to $RO$ made by $Adv$
Stronger Dolev-Yao Assumption

An encryption operation is *ideal* if, when used in $[[\cdot]]_\eta$

$$\ldots \forall M \notin C[S], \ldots$$
$$\Pr[\ldots m \leftarrow \text{Adv}(1^\eta, s) : m \in [[M]]_\eta] \leq \frac{1}{q(\eta)}$$
(Probability of hitting fixed target is small.)

An encryption operation is *perfect* (??) if, when used in

$$\ldots$$
$$\Pr[\ldots m \leftarrow \text{Adv}(1^\eta, s) : \exists M \notin C[S] \text{ s. t. } m \in [[M]]_\eta]$$
(Probability of hitting any target is small.)