A Computational Interpretation of
Dolev-Yao Adversaries

Jonathan Herzog
Massachusetts Institute of Technology

How to Analyze Cryptographic Protoco

Many well-known methods based on formal methods

- Strand Spaces, model checkers, theorem provers, et
- Most based on common set of assumptions (Dolev-Ya

However, formal approach not the only framework for 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
Jürjens (2001)

Other approaches to (directly) relating Dolev-Yao adver
underlying encryption (Guttman, Thayer, Zuck 2002)

Incorporating poly-time indistinguishability into process
(Mitchell et al. 1998, 1999, 2001)

Dolev-Yao model as universally composable reactive mac
(Backes, Pfitzmann, Waidner 2003)

Computational Cryptography: Worldvie

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 probabilit

- 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.

Computational Cryptography: Example

Definition of "semantic security" for public-key encryption

'The probability is negligible (in the asymptotic ca
that an efficient (PPT) adversary can tell if you encr
message m_1 or message m_2 (under a randomly picke
even if the adversary gets to pick m_1 and m_2.'

$$\forall \texttt{Adv}_{PPT}, \forall \text{ polynomials } q, \exists \eta_0 . \forall \eta \geq \eta_0$$

$$\begin{aligned}\Pr[\quad & (pk, sk) \leftarrow \texttt{KeyGen}(1^\eta); \\ & m_1, m_2 \leftarrow \texttt{Adv}(1^\eta, pk); \\ & b \leftarrow \texttt{CoinFlip}(1,2); \\ & c \leftarrow \texttt{Enc}(m_b, pk); \\ & g \leftarrow \texttt{Adv}(c, m_1, m_2, pk) : \\ & b = g \qquad\qquad] \leq \frac{1}{2} + \frac{1}{q(\eta)}\end{aligned}$$

Semantic Security, cont.

Semantic security (and more!) achieved by common sch

– Weak definition of security

Possible to instantiate given any hard problem*

Implies randomization of encryption

– Given plaintext/key can produce many possible ciph

Important note: security against *all* efficient adversaries

*Trapdoor permutation

Computational Approach: Pros and Co

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
- Transmitting and receiving abstract messages

Generally no randomness

Adversary assumed from restricted class of state machin

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 sen
known message to any honest participant

Dolev-Yao Adversary, Formalized

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

$$\mathtt{R}_0 \quad \mathtt{Q}_1 \quad \mathtt{R}_1 \quad \ldots \quad \mathtt{Q}_{n-1} \quad \mathtt{R}_{n-1} \quad \mathtt{Q}_n \quad \mathtt{R}_n$$

Each query $\mathtt{Q}_i$ must be *derivable* from initial adversary k
(S_0), all previous responses

If S is a set of messages, $C[S]$ is the smallest set so tha

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

$\mathtt{Q}_i$ is derivable at time i iff:

$$\mathtt{Q}_i \in C\left[S_0 \cup \{\mathtt{R}_0, \mathtt{R}_1, \ldots \mathtt{R}_{i-1}\}\right]$$

*Public-key setting

Generalizing Dolev-Yao Adversary

Fundamental assumption of Dolev-Yao model:

If Dolev-Yao adversary "knows" only messages in set S
make messages outside of $C[S]$

"How can you justify this?"

– How can we say that adversary has no other abilitie

Answer: adversary has no other abilities in computationa
model, either (if underlying encryption is sufficiently stro

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

But first, a technical detail

Message Encoding

Must first bring formal messages down into the world of b

Use both η, security parameter, and arbitrary computati
encryption scheme

Let $[\![M]\!]_\eta$ be the *encoding* of M

- Public-key analogue to encoding function of Abadi a
  Rogaway
- Nonces, keys, encoded as random strings
- Encoding of pairs straightforward
- Encoding of $\{|M|\}_K$ uses $[\![M]\!]_\eta$, $[\![K]\!]_\eta$ and arbitrar
  computational encryption scheme
  - May be randomized
- Function from messages to distributions on bit-strin

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 message outside $C[S]$

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

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

Implementing Ideal Encryption

Definition of ideal cryptography useless unless it can be

Need strong cryptography

A computational encryption scheme is *plaintext-aware*

1. It is semantically secure, and
2. The adversary must know the plaintext to every cipl creates itself.

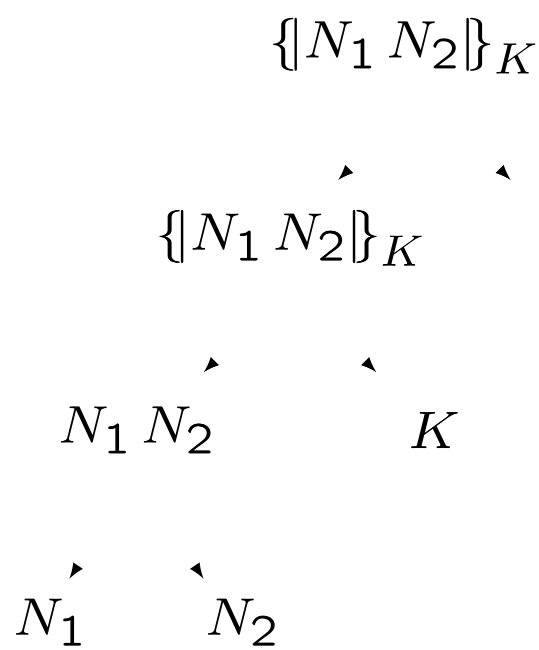
Thm: Let `Enc` be plaintext-aware. Then `Enc` is ideal.

Proof Overview:

1. Assume otherwise. Then adversary can create enc some $M \notin C[S]$.
2. Plaintext awareness implies adversary can create enc some atomic subterm not in $C[S]$.
3. Doing so violates semantic security

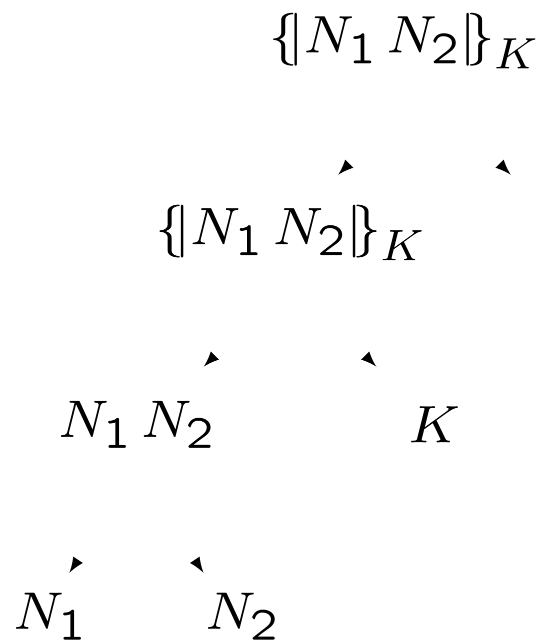
Proof Sketch (1/4)

Suppose adversary can create encoding of $M \notin C[S]$. 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.



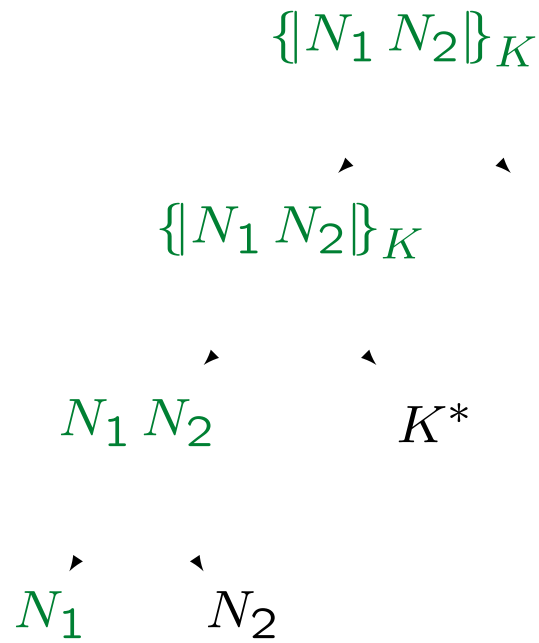
Proof Sketch (2/4)

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 $C[S]$.



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 $C\,[S]$.



Proof Sketch (4/4)

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

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 fro independent values

*It, or inverse, in parse tree of

Conclusions

Ideal encryption can be instantiated. This means...

Possible to limit computational adversary to attacks ava
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 throug 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 al

– $\texttt{KeyGen}$: Parameter $\longrightarrow$ PublicKey $\times$ PrivateKey
  - The (randomized) key generation algorithm
– $\texttt{Enc}$: PublicKey $\times$ String $\longrightarrow$ Ciphertext
  - The (randomized) encryption algorithm
– $\texttt{Dec}$: Ciphertext $\times$ PrivateKey $\longrightarrow$ String
  - The (deterministic) decryption algorithm

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

$$\texttt{Dec}(\texttt{Enc}(pk, m), sk) = m$$

Plaintext Awareness (Abr.)

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

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

Let $\texttt{Enc}$, adversary have access to RO

Encryption is *plaintext-aware* if for any adversary

$$c \leftarrow \texttt{Adv}(pk)$$

there exists an efficient *extractor* $\texttt{K}_{\texttt{Adv}}$ such that

$$\texttt{Dec}(sk, c) = \texttt{K}_{\texttt{Adv}}(c, pk, H)$$

where H is a list of the queries to RO made by $\texttt{Adv}$

Stronger Dolev-Yao Assumption

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

$$\ldots \forall\, M \notin C\,[S]\,, \ldots$$
$$\Pr[\ldots m \leftarrow \mathtt{Adv}(1^\eta, s) : m \in [\![M]\!]_\eta] \leq \frac{1}{q(\eta)}$$

(Probability of hitting fixed target is small.)

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

$$\ldots$$
$$\Pr[\ldots m \leftarrow \mathtt{Adv}(1^\eta, s) : \exists M \notin C\,[S] \text{ s. t. } m \in [\![M]\!]_\eta$$

(Probability of hitting any target is small.)