Inductive Proofs of Computational Security

Anupam Datta (CMU)
Joint work with Arnab Roy, Ante Derek, John Mitchell
(Stanford)
Outline

- Network Protocols
  - Partipator Model
  - Adversary Model

- Cryptographic Security
  - Cryptographic Primitives
  - Security Definitions

- Formal Proofs
  - Computational PCL: Syntax, Semantics, Proof System
Protocols

- Distributed Programs
  - Protocol is a fixed set of ‘roles’ written as programs
  - A ‘thread’ is an instance of a role being executed by a principal
  - A single principal can execute multiple threads

- Actions in a role
  - Communication: send m; recv m;
  - Pairing, Unpairing: m := pair m₀, m₁; match m as m₀, m₁;
  - Encryption, Decryption: m’ := enc m, k; m’ := dec m, k;
  - Nonce generation: new m;
  - Pattern matching: match m as m’; …
Kerberos V5

Network Protocols
- Participant Model
- Adversary Model
Active Computational Adversary
Abstraction: Protocol Execution Model

Adversary: Prob. Poly Time

Protocol Randomness

A

B

C

Adversary Randomness

Result:

Set of computational traces:

One for each adversary and protocol randomness

Network Protocols
  ▸ Partipator Model
  ▸ Adversary Model

Adversary Randomness:
• Random coin flips for the PPT algorithm

Protocol Randomness:
• Key generation
• Randomness for encryption, signatures, …
Basic concepts

- **Computational complexity**
  - Adversary runs in probabilistic polynomial time
    - Polynomial in security parameter
    - Key lengths also polynomial in security parameter

- **Acceptable advantage of adversary**
  - A negligible function $\nu(x): N \rightarrow R$ is a function that asymptotically decreases faster than the reciprocal of any polynomial in $x$, i.e.,
    \[
    \forall p. \exists N. \forall n > N. \nu(n) < \frac{1}{p(n)}
    \]
Example: Security of signatures

Existential Unforgeability under Chosen Message Attack

A signature scheme is CMA secure if

\[ \forall \text{Prob-Polytime } A. \text{ Advantage (A, } \eta \text{) is a negligible function of } \eta \]
Computational PCL

- Proof system for direct reasoning
  - \( \text{Verify} (X, \text{sig}_Y(m), Y) \land \text{Honest} (Y) \Rightarrow \text{Sign} (Y, m) \)
  - No explicit use of probabilities and computational complexity
  - No explicit arguments about actions of attackers

- Semantics capture idea that properties hold with high probability against PPT attackers
  - Explicit use of probabilities and computational complexity
  - Probabilistic polynomial time attackers
  - Soundness proofs one time

- Soundness implies result equivalent to security proof by cryptographic reductions
Proof System

DH0  \( \text{DHGood}(X, a, x) \), for \( a \) of any atomic type, except nonce, viz. name or key
DH1  \( \text{New}(Y, n) \land n \neq x \supset \text{DHGood}(X, n, x) \)
DH2  \( \text{[receive } m;]_X \text{ DHGood}(X, m, x) \)
DH3  \( \text{[} m := \text{expg } x; \text{]}_X \text{ DHGood}(X, m, x) \)
DH4  \( \text{DHGood}(X, m_0, x) \land \text{DHGood}(X, m_1, x) \text{ [} m := m_0.m_1; \text{]}_X \text{ DHGood}(X, m, x) \)
DH5  \( \text{DHGood}(X, m, x) \text{ [} m' := \text{symenc } m, k; \text{]}_X \text{ DHGood}(X, m', x) \)
DH6  \( \text{DHGood}(X, m, x) \text{ [} m' := \text{hash } m; \text{]}_X \text{ DHGood}(X, m', x) \)
Applications

- We proved the following protocols secure in the complexity theoretic model:
  - Kerberos V5 with Symmetric Key initialization
    - Secrecy proofs first time in literature
  - Kerberos V5 with Public Key initialization
    - Secrecy proofs first time in literature
  - IKEv2
    - Proofs first time in literature
- We found an attack on the first phase of Kerberos V5 with Diffie Hellman initialization, proposed an easy fix and proved the resulting protocol secure.
Why our way?

- Why logical methods?
  - Proofs are rigorous but shorter than semantic proofs
  - Carry the same meaning as the semantic proofs
  - Potentially automatable

- Why complexity theoretic model?
  - Protocols are built using cryptographic primitives
  - Cryptographers prove their constructions correct with respect to the complexity theoretic model
Inductive Trace Properties for Computational Security
Secrecy Notion: Real or Random Game

Adversary: Prob. Poly Time

Protocol Execution Generating Nonce $s_0$

Generate $s_1$
Choose $b \leftarrow \{0,1\}$

$Adv(A, \eta) = Pr[b' = b] - \frac{1}{2}$
IND-CCA Game

(Key Gen Algo $K$, Encryption Algo $E$, Decryption Algo $D$)
Fix security parameter $\eta$

Adversary

$m_0, m_1$ → $\text{Challenger}$

Choose $k \leftarrow K(\eta)$
Choose $b \leftarrow \{0, 1\}$

$c$ (*)

$D_k(c)$

$\text{Adv} (A, \eta) = \Pr[b' = b] - \frac{1}{2}$

An encryption scheme is IND-CCA secure if
$\forall$ Prob-Polytime $A$.
$\text{Adv} (A, \eta)$ is a negligible function of $\eta$

(*) $c$’s should be different from any encryption response
n-IND-CCA Game

(Key Gen Algo K, Encryption Algo E, Decryption Algo D)
Fix security parameter $\eta$

Adversary

| $i, m_0, m_1$ |
| $E_{k_i}(m_b)$ |
| $i, c$ (*) |

Challenger

| Choose $k_1, k_2, \ldots, k_n \leftarrow K(\eta)$ |
| Choose $b \leftarrow \{0,1\}$ |
| $D_{k_i}(c)$ |

$(\ast)$: c's should be different from any encryption response

Adv $(A, \eta) = \Pr[b' = b] - \frac{1}{2}$

An encryption scheme is n-IND-CCA secure if
$\forall$ Prob-Polytime A. Adv $(A, \eta)$ is a negligible function of $\eta$

[BBM00] shows that an encryption scheme is
n-IND-CCA secure $\iff$ IND-CCA secure.
Secrecy Notion: Indistinguishability

- **Secrecy Property:**
  - Indistinguishability for the nonce holds if
    \[ \forall \text{Prob-Polytime } A. \]
    \[ \text{Adv} (A, \eta) \text{ is a negligible function of } \eta \]

- **We want to prove:**
  - If the encryption scheme is IND-CCA secure then indistinguishability for the nonce holds if it is protected by a set of keys.

- **Proof Strategy:**
  - Reduction! – if an adversary can break protocol then there is an adversary which can break CCA (contrapositive)
Reduction

Show that:
If for nonce indist game Adv (A, \(\eta\)) is non-negligible
Then for Simulator S, Adv(S, \(\eta\)) against n-IND-CCA game is non-negligible

Protocol
Adversary:
Prob. Poly Time

Simulate Protocol Execution Generating Nonce \(s_0\)

Generate \(s_1\) Choose \(d \leftarrow \{0,1\}\)

Choose \(k_1, \ldots, k_n \leftarrow K(\eta)\)
Choose \(b \leftarrow \{0, 1\}\)
Protocol example

Adversary: Prob. Poly Time

A
new s
m := pair a, s
e := enc m, k₁
send e

B
receive m'
l := dec m', k₁
t := pair l, c
r := enc t, k₂
send r

C
receive e'
j := dec e’, k₃

Generate s₁
Choose b ← {0,1}

S_b
b'
Adv (A, η) for nonce indist game = Adv(S, η) against n-IND-CCA game
Secretive Protocols

- A trace is a secretive trace with respect to nonce s and set of keys K if the following properties hold for every thread belonging to honest principals:
  - The thread which generates s, ensures that s is encrypted with a key k in K in any message sent out.
  - Whenever a thread decrypts a message with a key k in K and parses the decryption, it ensures that the results are re-encrypted with some key k' in K in any message sent out.

- A protocol is secretive if it overwhelmingly produces secretive traces.

- An inductive property over actions of honest parties
  - Formalization in Computational Protocol Composition Logic.
Theorem:

If

- the protocol is “secretive”
- the nonce-generator is honest
- the key-holders are honest

Then

- the key generated from the nonce satisfies indistinguishability

Do an inductive proof - for each protocol

Proof is by reduction to a multi-party IND-CCA game – one time soundness proof
Proof System to Establish “Secretive” Protocol – “Good” terms

- Proof of construction of good terms is carried out inductively over actions of honest principals

\[
\begin{align*}
G_0 & \quad \text{Good}(X, a, s, \mathcal{K}), \text{ if } a \text{ is of an atomic type different from nonce or key} \\
G_1 & \quad \text{New}(Y, n) \land n \neq s \supset \text{Good}(X, n, s, \mathcal{K}) \\
G_2 & \quad [\text{receive } m;]_X \text{Good}(X, m, s, \mathcal{K}) \\
G_3 & \quad \text{Good}(X, m, s, \mathcal{K}) [a]_X \text{Good}(X, m, s, \mathcal{K}), \text{ for all actions } a \\
G_4 & \quad \text{Good}(X, m, s, \mathcal{K}) [\text{match } m \text{ as } m';]_X \text{Good}(X, m', s, \mathcal{K}) \\
G_5 & \quad \text{Good}(X, m_0, s, \mathcal{K}) \land \text{Good}(X, m_1, s, \mathcal{K}) [m := m_0.m_1;]_X \text{Good}(X, m, s, \mathcal{K}) \\
G_6 & \quad \text{Good}(X, m, s, \mathcal{K}) [\text{match } m \text{ as } m_0.m_1;]_X \text{Good}(X, m_0, s, \mathcal{K}) \land \text{Good}(X, m_1, s, \mathcal{K}) \\
G_7 & \quad \text{Good}(X, m, s, \mathcal{K}) \lor k \in \mathcal{K} [m' := \text{symenc } m, k;]_X \text{Good}(X, m', s, \mathcal{K}) \\
G_8 & \quad \text{Good}(X, m, s, \mathcal{K}) \land k \notin \mathcal{K} [m' := \text{symdec } m, k;]_X \text{Good}(X, m', s, \mathcal{K})
\end{align*}
\]
Proof System to Establish “Secretive” Protocol – Induction

A protocol is “secretive” if all honest participants send out only “good” terms.

\[ \forall \text{roles } \rho \text{ in protocol } Q. \]
\[ \forall \text{segments } P \text{ in role } \rho. \]

\[ \text{SendGood}(X, s, K) [P]_x \Phi \Rightarrow \text{SendGood}(X, s, K) \]
\[ Q \vdash \Phi \Rightarrow \text{Secretive}(s, K) \]
Example

- Let \( n \) be the putative secret and \( K = \{k_1, k_2, \ldots\} \)
- We want to prove that protocol satisfies \( \text{Secretive}(n, K) \)
- Consider the following fragment of the protocol:

```
recv e;
t := dec e, k;
match t as A.n';
p := enc n', k;
send p;
```
Case: $k \notin K$

recv e;

Good(e, n, K)

Good(e, n, K) \land k \notin K

t := dec e, k;

Good(t, n, K)

Good(t, n, K)

Axiom G2

Good(t, n, K)

Axiom G8

Good(t, n, K)

Axiom G8

match t as A.n’;

Good(A, n, K) \land Good(n’, n, K)

Good(A, n, K) \land Good(n’, n, K)

Axiom G4

Good(n’, n, K)

p := enc n’, k;

Good(p, n, K)

Good(p, n, K)

Axiom G6

send p;

Good(p, n, K)

Axiom G6
Case: $k \in K$

- $\text{recv } e;$
- $t := \text{dec } e, k;$
- match $t$ as $A.n'$;
- $k \in K$
- $p := \text{enc } n', k;$
- $\text{Good}(p, n, K)$
- $\text{Good}(p, n, K)$
- send $p;$
- Axiom G7
Good Keys: A weaker notion [DDMW06]

- Key is “good” for a certain purpose
- Intuition: Exchanged key is good for encrypting messages if no attacker can win an appropriate game played with that key.
Relating “Secretive” Protocols to “Good” Keys

- **Theorem:**
  - If
    - the protocol is “secretive”
    - the nonce-generator is honest
    - the nonce may be used as a key
    - the key-holders are honest
  - Then
    - the key generated from the nonce is a “good” key

Proof is by reduction to a multi-party IND-CCA game – one time soundness proof

Do an inductive proof - for each protocol
Many interesting protocols establish a hierarchy of keys. For example – Kerberos, IEEE 802.11i

Keys at level i may be used to encrypt keys of level j < i
# Some Results

<table>
<thead>
<tr>
<th>Language</th>
<th>Crypto Assumption</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret not used as a key</td>
<td>IND-CCA</td>
<td>Secrecy: Indist for level-1</td>
</tr>
<tr>
<td>Secret used as a symmetric key</td>
<td>IND-CCA</td>
<td>Secrecy: GoodKey for level-1</td>
</tr>
<tr>
<td>Secret not used as a key</td>
<td>IND-CCA</td>
<td>Secrecy: Indist for key DAGs</td>
</tr>
<tr>
<td>Secret used as a symmetric key</td>
<td>IND-CCA</td>
<td>Secrecy: GoodKey for key DAGs</td>
</tr>
<tr>
<td>Auth of msg encrypted with the secret</td>
<td>IND-CPA+INT-CTXT</td>
<td>Authentication for key DAGs</td>
</tr>
</tbody>
</table>
Kerberos V5 results

If Client C completes the protocol with Kerberos Authentication Server K, Ticket Granting Server T and Application Server S then information available to C can be sufficient to guarantee:

<table>
<thead>
<tr>
<th>Type</th>
<th>Honesty Assumption</th>
<th>Guarantee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticity</td>
<td>C, K</td>
<td>A message containing a valid ticket granting ticket was indeed sent by K intended for (C, T), with overwhelming probability.</td>
</tr>
<tr>
<td>Authenticity</td>
<td>C, K, T</td>
<td>A message containing a valid server ticket was indeed sent by T intended for (C, S), with overwhelming probability.</td>
</tr>
<tr>
<td>Secrecy</td>
<td>C, K, T</td>
<td>AKey is a good key for C, K and T.</td>
</tr>
<tr>
<td>Secrecy</td>
<td>C, K, T, S</td>
<td>SKey is a good key for C, K, T and S.</td>
</tr>
</tbody>
</table>

- Similar results are proved from the perspective of K, T and S as well
- Theorems proved in [ESORICS2007]
Diffie Hellman
Fix group $G$ satisfying certain cryptographic properties

$$g^{xy}$$ is secret to a passive adversary
Kerberos with DHINIT

KAS

Client

Cert\_C, sig\_C, C, n\_1

Cert\_K, sig\_K, [Akey\_C]\_K\_TK, [Akey\_n\_1\_T]\_k

TGS

Client

[Akey\_C]\_K\_TK, [C]\_Akey, n\_2

[Skey\_C]\_K\_ST, [Skey\_n\_1\_T]\_AKey

Server

Client

[Skey\_C]\_K\_ST, [C.t]\_SKey

[t]\_SKey
Is the KAS authenticated after the first phase?

1. Cert$_C$, Sig$_C$ ("Auth", Hash($C$, $T$, $n_1$), $m_1$, $g^x$), $C$, $T$, $n_1$

2. Cert$_I$, Sig$_I$ ("Auth", Hash($I$, $T$, $n_1$), $m_1$, $g^x$), $I$, $T$, $n_1$

3. Cert$_K$, Sig$_K$ ("DHKey", $g^y$, $m_1$), TGT($I$), Enc-Akey

4. Cert$_K$, Sig$_K$ ("DHKey", $g^y$, $m_1$), TGT($I$), Enc-Akey
Decisional Diffie Hellman Assumption

Fix security parameter $\eta$
$G(\eta), g \leftarrow G$

$\textbf{Adversary}$
$g^x, g^y, g^z$

$\textbf{Challenger}$
$x, y, w \leftarrow [1, |G|]$
$b \leftarrow \{0, 1\}$
If $b = 0$
then $z = w$
else $z = xy$

$\textbf{Adv (A, }\eta\text{)} = \Pr[b' = b] - \frac{1}{2}$

The DDH assumption holds if
\[ \forall \text{Prob-Polytime A.} \]
$\text{Adv (A, }\eta\text{)}$ is a negligible function of $\eta$
Reduction

Show that:
If for key indist game Adv (A, η) is non-negligible
Then for Simulator S, Adv(S, η) against DDH game is non-negligible
Protocol example

Adversary: Prob. Poly Time

A
new a
Ga := exp g, a
send ga

B
receive Ga'
new b
Gb := exp g, b
k₀ := exp Ga', b
r := pair B, Gb
send r

C
receive e'
t := pair C, e'
send t

Generate k₁
Choose b ← {0,1}

k₀

b'
Reduction

Adv (A, η) for DH-key indist game = Adv(S, η) against DDH game
DHStrongSecretive Property

- A trace is a \textit{DHStrongSecretive} trace with respect to \((x, y)\) if the following properties hold for every thread belonging to honest principals if,
  - the thread which generates \(x\) ensures that it appears only exponentiated as \(g^x\) in any message sent out. Similarly for \(y\).
  - the generators of \(x, y\) only use each other’s DH exponentials to generate the key.

- A protocol is \textit{DHStrongSecretive} if it overwhelmingly produces DHStrongSecretive traces.

- An inductive property over actions of honest parties
  - Formalization in Computational Protocol Composition Logic.
Relating “DHStrongSecretive” Protocols to Computational Secrecy

- **Theorem:**
  - If
    - the protocol is (x, y)-DHStrongSecretive
    - the x, y generators are honest
  - Then
    - the key generated from $g^{xy}$ satisfies key indistinguishability

Proof is by reduction to a DDH game – one time soundness proof
## Some Results

<table>
<thead>
<tr>
<th>Language</th>
<th>Crypto Assumption</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret not used as a key</td>
<td>DDH</td>
<td>Secrecy: Indist</td>
</tr>
<tr>
<td>Secret used as a symmetric key</td>
<td>DDH+IND-CPA/CCA</td>
<td>Secrecy: GoodKey for DHStrongSecretive</td>
</tr>
<tr>
<td>Secret used as a symmetric key</td>
<td>DDH+INT-CTX</td>
<td>Authentication for DHStrongSecretive</td>
</tr>
<tr>
<td>Secret used as a symmetric key</td>
<td>CDH+RO+INT-CTX</td>
<td>Authentication for DHSecretive</td>
</tr>
<tr>
<td>Secret used to protect other secrets</td>
<td>DDH+IND-CCA</td>
<td>Secrecy of keys protected by DHKey</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>... so on</td>
</tr>
</tbody>
</table>
Axioms to prove DH-“safety”

DH0  $\text{DHGood}(X, a, x)$, for $a$ of any atomic type, except nonce, viz. name or key
DH1  $\text{New}(Y, n) \land n \neq x \supset \text{DHGood}(X, n, x)$
DH2  $[\text{receive } m;]_X \text{DHGood}(X, m, x)$
DH3  $[m := \text{expg } x;]_X \text{DHGood}(X, m, x)$
DH4  $\text{DHGood}(X, m_0, x) \land \text{DHGood}(X, m_1, x) [m := m_0.m_1;]_X \text{DHGood}(X, m, x)$
DH5  $\text{DHGood}(X, m, x) [m' := \text{symenc } m, k;]_X \text{DHGood}(X, m', x)$
DH6  $\text{DHGood}(X, m, x) [m' := \text{hash } m;]_X \text{DHGood}(X, m', x)$

Pre-condition  Action  Post-condition
Kerberos DHINIT Results

- If Client C completes the protocol with Kerberos Authentication Server K, Ticket Granting Server T and Application Server S then information available to C can be sufficient to guarantee:

<table>
<thead>
<tr>
<th>Type</th>
<th>Honesty Assumption</th>
<th>Guarantee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticity</td>
<td>C, K</td>
<td>A message containing a valid ticket granting ticket was indeed sent by K intended for (C, T), with overwhelming probability.</td>
</tr>
<tr>
<td>Authenticity</td>
<td>C, K, T</td>
<td>A message containing a valid server ticket was indeed sent by T intended for (C, S), with overwhelming probability.</td>
</tr>
<tr>
<td>Secrecy</td>
<td>C, K, T</td>
<td>AKey is a good key for C, K and T.</td>
</tr>
<tr>
<td>Secrecy</td>
<td>C, K, T, S</td>
<td>SKey is a good key for C, K, T and S.</td>
</tr>
</tbody>
</table>

- Similar results are proved from the perspective of K, T and S as well
- Theorems proved in [TGC2007]
IKEv2 Results

- IKEv2 is a protocol used to negotiate keys at the beginning of an IPsec session.
- If Initiator I completes the protocol with Responder R then I can infer the following guarantees:

<table>
<thead>
<tr>
<th>Type</th>
<th>Honesty Assumption</th>
<th>Guarantee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticity</td>
<td>I, R</td>
<td>Intended messages were indeed received and sent by R with overwhelming probability.</td>
</tr>
<tr>
<td>Secrecy</td>
<td>I, R</td>
<td>The exchanged keys are good keys for I and R.</td>
</tr>
</tbody>
</table>

- Similar results are proved from the perspective of R as well.
Conclusion
Summary of Results

- Sym-Key Kerberos
- Pub-Key Kerberos
- DH-Key Kerberos
- IKEv2

- Secretive
- DHSecretive

- Logics – Cryptographic Reduction implying Soundness

- Proofs by Direct Reasoning

- Inductive Trace Properties
PCL: Big Picture

High-level proof principles

PCL
• Syntax (Properties)
• Proof System (Proofs)

Symbolic Model
• PCL Semantics
  (Meaning of formulas)

Soundness
Theorem
(Induction)

Computational PCL
• Syntax $\pm \Delta$
• Proof System $\pm \Delta$

Soundness
Theorem
(Reduction)

Cryptographic Model
• PCL Semantics
  (Meaning of formulas)

Unbounded # concurrent sessions

[BPW, MW,...]

Polynomial # concurrent sessions
Thanks!

Questions?