Formal Verification of Cryptographic Protocols in Spi-Calculus

Eijiro Sumii
Tohoku University
Caution

♦ Literature on spi-calculus is confusing
  – Inconsistent terminology
  – Some "results" found too weak or even wrong
♦ This talk is my own combination of various results on spi-calculus
Outline

♦ What is spi-calculus?
  – Syntax and operational semantics
♦ Example protocol
♦ Attack against the example protocol
♦ Formalizing secrecy by non-interference
♦ Proving secrecy by hedged bisimulations
♦ Conclusions
What is spi-calculus? [Abadi-Gordon 99]

♦ spi-calculus = \( \pi \)-calculus + (shared-key) perfect encryption primitives

The only equation is:
\[
\text{dec}(\text{enc}(\text{Msg}, \text{key}), \text{key}) = \text{Msg}
\]

Cf. Textbook RSA is malleable:
\[
\text{enc}(\text{Msg}_1, \text{pubkey}) \times \text{enc}(\text{Msg}_2, \text{pubkey}) = \text{enc}(\text{Msg}_1 \times \text{Msg}_2, \text{pubkey})
\]
Syntax

\[ M, N ::= \]
\[ x \] message
\[ \{ M_1, \ldots, M_n \}_N \] name
\[ P, Q, R ::= \]
\[ 0 \] ciphertext
\[ M(N).P \] process
\[ M(x).P \] inaction
\[ M \] sending
\[ P | Q \] receiving
\[ (\nu x)P \] parallel composition
\[ !P \] restriction
\[ \text{case } M \text{ of } \{ x_1, \ldots, x_n \}_N \text{ in } P \] replication
\[ [M = N]P \] decryption
\[ \text{matching} \]
Operational Semantics (1/2): Structural Equivalence

\[
\text{case } \{M_1, \ldots, M_n\}_N \text{ of } \{x_1, \ldots, x_n\}_N \text{ in } P \\
\equiv [M_1, \ldots, M_n/x_1, \ldots, x_n]P
\]

\[
[M = M]P \equiv P \quad !P \equiv P | !P
\]

\[
P \mid (\nu x)Q \equiv (\nu x)(P \mid Q) \quad \text{if } x \not\in \text{free}(P)
\]

\[
P \mid 0 \equiv P \quad P \mid Q \equiv Q \mid P \quad (P \mid Q) \mid R \equiv P \mid (Q \mid R)
\]

\[
P \equiv P' \\
\frac{P \mid Q \equiv P' \mid Q}{P \mid Q \equiv P' \mid Q}
\]

\[
(\nu x)P \equiv (\nu x)P'
\]

\[
P \equiv P \\
\frac{P \equiv Q}{Q \equiv P}
\]

\[
P \equiv Q \quad Q \equiv R \quad P \equiv R
\]
Operational Semantics (2/2): Reaction Relation

\[ \overline{x}(M).P \mid x(y).Q \rightarrow P \mid [M/y]Q \]

\[ \frac{P \equiv P' \quad P' \rightarrow Q' \quad Q' \equiv Q}{P \rightarrow Q} \]

\[ \frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \quad \frac{P \rightarrow P'}{(\nu x)P \rightarrow (\nu x)P'} \]
Outline

♦ What is spi-calculus?
  – Syntax and operational semantics
♦ Example protocol
♦ Attack against the example protocol
♦ Formalizing secrecy by non-interference
♦ Proving secrecy by hedged bisimulations
♦ Conclusions
Example: A Naive Protocol
(Wide Mouthed Frog Protocol)

1. A → S : \{K_{AB}\}_{K_{AS}}
2. S → B : \{K_{AB}\}_{K_{BS}}
3. B → A : \{M\}_{K_{AB}}

\[
P_A = (\nu K_{AB})_{c_{AS}}\langle\{K_{AB}\}_{K_{AS}}\rangle.
\]
\[
c_{AB}(n).\text{case } n \text{ of } \{m\}_{K_{AB}} \text{ in } 0
\]
\[
P_S = c_{AS}(x).\text{case } x \text{ of } \{y\}_{K_{AS}} \text{ in } c_{BS}\langle\{y\}_{K_{BS}}\rangle
\]
\[
P_B = c_{BS}(x).\text{case } x \text{ of } \{y\}_{K_{BS}} \text{ in } c_{AB}\langle\{M\}_y\rangle
\]

The whole system is:

\[
(\nu K_{AS})(\nu K_{BS})(P_A | P_S | P_B)
\]
How does the protocol run? 

(1/2)

\[ (\nu K_{AS})(\nu K_{BS})(P_A | P_S | P_B) \]

\[ \equiv (\nu K_{AS})(\nu K_{BS})(\nu K_{AB}) \]

\[ (c_{AS}(\{K_{AB}\}K_{AS}).c_{AB}(n).\text{case } n \text{ of } \{m\}K_{AB} \text{ in } 0 | \]

\[ c_{AS}(x).\text{case } x \text{ of } \{y\}K_{AS} \text{ in } \overline{c_{BS}(\{y\}K_{BS})} | \]

\[ c_{BS}(x).\text{case } x \text{ of } \{y\}K_{BS} \text{ in } \overline{c_{AB}(\{M\}y)} ) \]

\[ \rightarrow (\nu K_{AS})(\nu K_{BS})(\nu K_{AB}) \]

\[ (c_{AB}(n).\text{case } n \text{ of } \{m\}K_{AB} \text{ in } 0 | \]

\[ \text{case } \{K_{AB}\}K_{AS} \text{ of } \{y\}K_{AS} \text{ in } \overline{c_{BS}(\{y\}K_{BS})} | \]

\[ c_{BS}(x).\text{case } x \text{ of } \{y\}K_{BS} \text{ in } \overline{c_{AB}(\{M\}y)} ) \]

\[ \equiv (\nu K_{AS})(\nu K_{BS})(\nu K_{AB}) \]

\[ (c_{AB}(n).\text{case } n \text{ of } \{m\}K_{AB} \text{ in } 0 | \]

\[ \overline{c_{BS}(\{K_{AB}\}K_{BS})} | \]

\[ c_{BS}(x).\text{case } x \text{ of } \{y\}K_{BS} \text{ in } \overline{c_{AB}(\{M\}y)} ) \]
How does the protocol run? (2/2)

\((\nu K_{AS})(\nu K_{BS})(\nu K_{AB})\)

\((c_{AB}(n).\text{case } n \text{ of } \{m\}^{K_{AB}} \text{ in } 0 |\)

\(c_{BS}\langle\{K_{AB}\}^{K_{BS}}\rangle)\)

\(c_{BS}(x).\text{case } x \text{ of } \{y\}^{K_{BS}} \text{ in } c_{AB}\langle\{M\}^{y}\rangle)\)

\(\rightarrow (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})\)

\((c_{AB}(n).\text{case } n \text{ of } \{m\}^{K_{AB}} \text{ in } 0 |\)

\text{case } \{K_{AB}\}^{K_{BS}} \text{ of } \{y\}^{K_{BS}} \text{ in } c_{AB}\langle\{M\}^{y}\rangle)\)

\(\equiv (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})\)

\((c_{AB}(n).\text{case } n \text{ of } \{m\}^{K_{AB}} \text{ in } 0 |\)

\(c_{AB}\langle\{M\}^{K_{AB}}\rangle)\)

\(\rightarrow (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})\)

\(\text{case } \{M\}^{K_{AB}} \text{ of } \{m\}^{K_{AB}} \text{ in } 0\)

\(\equiv (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})0\)
How does the protocol run? (2/2)

\[
(\nu K_{AS})(\nu K_{BS})(\nu K_{AB})
\]
\[
(\text{c}_{AB}(n).\text{case } n \text{ of } \{ m \}\text{ in } 0 |
\]
\[
\text{c}_{BS}\langle\{ K_{AB}\} K_{BS}\rangle |
\]
\[
\text{c}_{BS}(x).\text{case } x \text{ of } \{ y \}\text{ in } \text{c}_{AB}\langle\{ M \}\text{ in } y\rangle)
\]
\[
\rightarrow (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})
\]
\[
(\text{c}_{AB}(n).\text{case } n \text{ of } \{ m \}\text{ in } 0 |
\]
\[
\text{case } \{ K_{AB}\} K_{BS} \text{ of } \{ y \} K_{BS} \text{ in } \text{c}_{AB}\langle\{ M \}\text{ in } y\rangle)
\]
\[
\equiv (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})
\]
\[
(\text{c}_{AB}(n).\text{case } n \text{ of } \{ m \}\text{ in } 0 |
\]
\[
\text{c}_{AB}\langle\{ M \}\text{ in } K_{AB}\rangle)
\]
\[
\rightarrow (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})
\]
\[
\text{case } \{ M \} K_{AB} \text{ of } \{ m \} K_{AB} \text{ in } 0
\]
\[
\equiv (\nu K_{AS})(\nu K_{BS})(\nu K_{AB}) 0
\]
Outline

♦ What is spi-calculus?
  – Syntax and operational semantics
♦ Example protocol
♦ Attack against the example protocol
♦ Formalizing secrecy by non-interference
♦ Proving secrecy by hedged bisimulations
♦ Conclusions
Parallel runs of the protocol (1/2)

1. $A \rightarrow S : \{K_{AB}\} K_{AS}$
2. $S \rightarrow B : \{K_{AB}\} K_{BS}$
3. $B \rightarrow A : \{M\} K_{AB}$

1'. $B \rightarrow S : \{K_{BE}\} K_{BS}$
2'. $S \rightarrow E : \{K_{BE}\} K_{ES}$
3'. $E \rightarrow B : \{M'\} K_{BE}$
Parallel runs of the protocol (2/2)

\[ P_A = (\nu K_{AB})c_{AS}\langle\{K_{AB}\}K_{AS}\rangle. \]
\[ c_{AB}(n).\text{case } n \text{ of } \{m\}K_{AB} \text{ in } 0 \]

\[ P_S = c_{AS}(x).\text{case } x \text{ of } \{y\}K_{AS} \text{ in } c_{BS}\langle\{y\}K_{BS}\rangle \]
\[ \text{in } \quad c_{BS}(x').\text{case } x' \text{ of } \{y'\}K_{BS} \text{ in } c_{ES}\langle\{y'\}K_{ES}\rangle \]

\[ P_B = c_{BS}(x).\text{case } x \text{ of } \{y\}K_{BS} \text{ in } c_{AB}\langle\{M\}?y\rangle \]
\[ \text{in } \quad (\nu K_{BE})c_{BS}\langle\{K_{BE}\}K_{BS}\rangle. \]
\[ c_{BE}(n').\text{case } n' \text{ of } \{m'\}K_{BE} \text{ in } 0 \]

\[ P_E = c_{ES}(x').\text{case } x' \text{ of } \{y'\}K_{ES} \text{ in } c_{BE}\langle\{M'\}?y'\rangle \]
Exercise (?)

♦ Write down the reduction of 

\((\forall K_A)(\forall K_B)(\forall K_E)(P_A \mid P_S \mid P_B \mid P_E)\).
What if E is evil in fact?

♦ Assumption: attacker has full access to open channels (Dolev-Yao model)
♦ Result: not only M' but also M may leak!

1'\textsubscript{a}. \ B \rightarrow \ E(S) : \ \{K_{BE}\}_{K_{BS}}

2. \ E(S) \rightarrow \ B : \ \{K_{BE}\}_{K_{BS}}

1'\textsubscript{b}. \ E(B) \rightarrow S : \ \{K_{BE}\}_{K_{BS}}

2'. \ S \rightarrow E : \ \{K_{BE}\}_{K_{ES}}

3. \ B \rightarrow E(A) : \ \{M\}_{K_{BE}}
How does the attack work?

\[ P'_E = c'_{BS}(z).c_{BS}(z).c'_{BS}(z).c_{ES}(x').\text{case } x' \text{ of } \{ y' \}_{K_{ES}} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{ m \}_{y'} \text{ in DoEvil}_m \]

\[ P'_E \mid (\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid P_B) \]

\[ (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})(\nu K_{BE}) \]

\[ (c'_{BS}(z).c_{BS}(z).c'_{BS}(z).c_{ES}(x').\text{case } x' \text{ of } \{ y' \}_{K_{ES}} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{ m \}_{y'} \text{ in DoEvil}_m \mid 
\]

\[ c_{AS}\langle\{ K_{AB}\}_{K_{AS}}.c_{AB}(n).\text{case } n \text{ of } \{ m \}_{K_{AB}} \text{ in } 0 \mid c_{AS}(x).\text{case } x \text{ of } \{ y \}_{K_{AS}} \text{ in } c_{BS}\langle\{ y \}_{K_{BS}} \mid 
\]

\[ c'_{BS}(x').\text{case } x' \text{ of } \{ y' \}_{K_{BS}} \text{ in } c_{ES}\langle\{ y' \}_{K_{ES}} \mid 
\]

\[ c_{BS}(x).\text{case } x \text{ of } \{ y \}_{K_{BS}} \text{ in } c_{AB}\langle\{ M \}_{y} \mid 
\]

\[ c'_{BS}\langle\{ K_{BE}\}_{K_{BS}}.c_{BE}(n').\text{case } n' \text{ of } \{ m' \}_{K_{BE}} \text{ in } 0 \} \]
How does the attack work?

\[ P'_E = c'_BS(z).c_{BS}(z).c'_BS(z).\]
\[ c_{ES}(x').\text{case } x' \text{ of } \{ y' \} K_{ES} \text{ in} \]
\[ c_{AB}(n).\text{case } n \text{ of } \{ m \} y' \text{ in } \text{DoEvil}_m \]

\[ P'_E | (\nu K_{AS})(\nu K_{BS})(P_A | P_S | P_B) \]
\[ (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})(\nu K_{BE}) \]
\[ (c_{BS}\langle \{ K_{BE} \} K_{BS} \rangle).c_{BS}\langle \{ K_{BE} \} K_{BS} \rangle. \]
\[ c_{ES}(x').\text{case } x' \text{ of } \{ y' \} K_{ES} \text{ in} \]
\[ c_{AB}(n).\text{case } n \text{ of } \{ m \} y' \text{ in } \text{DoEvil}_m | \]
\[ c_{AS}\langle \{ K_{AB} \} K_{AS} \rangle. c_{AB}(n).\text{case } n \text{ of } \{ m \} K_{AB} \text{ in } 0 | \]
\[ c_{AS}(x).\text{case } x \text{ of } \{ y \} K_{AS} \text{ in } c_{BS}\langle \{ y \} K_{BS} \rangle | \]
\[ c'_BS(x').\text{case } x' \text{ of } \{ y' \} K_{BS} \text{ in } c_{ES}\langle \{ y' \} K_{ES} \rangle | \]
\[ c_{BS}(x).\text{case } x \text{ of } \{ y \} K_{BS} \text{ in } c_{AB}\langle \{ M \} y \rangle | \]
\[ c_{BE}(n').\text{case } n' \text{ of } \{ m' \} K_{BE} \text{ in } 0 \]
How does the attack work?

\[ P'_E = c'_{BS}(z).c_{BS}(z).c'_{BS}(z).c_{ES}(x').\text{case } x' \text{ of } \{ y' \} K_{ES} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{ m \} y' \text{ in DoEvil}_m \]

\[ P'_E | (\nu K_{AS})(\nu K_{BS})(P_A | P_S | P_B) \]
\[ (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})(\nu K_{BE}) \]
\[ (c_{BS}\langle\{ K_{BE} \} K_{BS} \rangle)c_{BS}\langle\{ K_{BE} \} K_{BS} \rangle).c'_{BS}\langle\{ K_{BE} \} K_{BS} \rangle).c_{ES}(x').\text{case } x' \text{ of } \{ y' \} K_{ES} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{ m \} y' \text{ in DoEvil}_m | \]
\[ c_{AS}\langle\{ K_{AB} \} K_{AS} \rangle)c_{AB}(n).\text{case } n \text{ of } \{ m \} K_{AB} \text{ in } 0 | \]
\[ c_{AS}(x).\text{case } x \text{ of } \{ y \} K_{AS} \text{ in } c_{BS}\langle\{ y \} K_{BS} \rangle | \]
\[ c'_{BS}(x').\text{case } x' \text{ of } \{ y' \} K_{BS} \text{ in } c_{ES}\langle\{ y' \} K_{ES} \rangle | \]
\[ c_{BS}(x).\text{case } x \text{ of } \{ y \} K_{BS} \text{ in } c_{AB}\langle\{ M \} y \rangle | \]
\[ c_{BE}(n').\text{case } n' \text{ of } \{ m' \} K_{BE} \text{ in } 0 \]
How does the attack work?

\[ P'_E = c'_BS(z).c_{BS}(z).c'_BS(z).c_{ES}(x').\text{case } x' \text{ of } \{y'\}_{K_{ES}} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{m\}_{y'} \text{ in DoEvil}_m \]

\[ P'_E \rightarrow^* \frac{(\nu K_{AS})(\nu K_{BS})(P_A | P_S | P_B)}{(\nu K_{AS})(\nu K_{BS})(\nu K_{AB})(\nu K_{BE})} \]

\[ (c'_BS(\{K_{BE}\}_{K_{BS}}).c_{ES}(x').\text{case } x' \text{ of } \{y'\}_{K_{ES}} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{m\}_{y'} \text{ in DoEvil}_m;| \]

\[ c_{AS}(\{K_{AB}\}_{K_{AS}}).c_{AB}(n).\text{case } n \text{ of } \{m\}_{K_{AB}} \text{ in } c_{BE}(n').\text{case } n' \text{ of } \{m'\}_{K_{BE}} \text{ in } 0) \]
How does the attack work?

\[ P'_E = c'_{BS}(z).\overline{c_{BS}(z)}.c'_{BS}(z).\]
\[ c_{ES}(x').\text{case } x' \text{ of } \{ y' \} _{K_{ES}} \text{ in }\]
\[ c_{AB}(n).\text{case } n \text{ of } \{ m \} _{y'} \text{ in } \text{DoEvil}_m \]

\[ P'_E \mid (\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid P_B) \]
\[ (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})(\nu K_{BE}) \]
\[ (c'_{BS}\langle\{K_{BE}\}_{K_{BS}}\rangle.\]
\[ c_{ES}(x').\text{case } x' \text{ of } \{ y' \} _{K_{ES}} \text{ in }\]
\[ c_{AB}(n).\text{case } n \text{ of } \{ m \} _{y'} \text{ in } \text{DoEvil}_m.\]
\[ \overline{c_{AS}\langle\{K_{AB}\}_{K_{AS}}\rangle}.c_{AB}(n).\text{case } n \text{ of } \{ m \} _{K_{AB}} \text{ in 0 }\]
\[ c_{AS}(x).\text{case } x \text{ of } \{ y \} _{K_{AS}} \text{ in } \overline{c_{BS}\langle\{y\} _{K_{BS}}\rangle}\]
\[ c'_{BS}(x').\text{case } x' \text{ of } \{ y' \} _{K_{BS}} \text{ in } \overline{c_{ES}\langle\{y'\} _{K_{ES}}\rangle}\]
\[ \overline{c_{AB}\langle\{M\} _{K_{BE}}\rangle}\]
\[ c_{BE}(n').\text{case } n' \text{ of } \{ m' \} _{K_{BE}} \text{ in 0} \]
How does the attack work?

\[ P'_E = c'_{BS}(z).c_{BS}(y).c'_{BS}(z). \\
    c_{ES}(x').\text{case } x' \text{ of } \{y\}'_{K_{ES}} \text{ in } \\
    c_{AB}(n).\text{case } n \text{ of } \{m\}'_{y'} \text{ in DoEvil}_m \]

\[ 
\begin{align*}
    &P'_E \mid (\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid P_B) \\
    \xrightarrow{*} & (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})(\nu K_{BE}) \\
    & c_{ES}(x').\text{case } x' \text{ of } \{y\}'_{K_{ES}} \text{ in } \\
    & c_{AB}(n).\text{case } n \text{ of } \{m\}'_{y'} \text{ in DoEvil}_m: \\
    & c_{AS} \langle \{k_{AB}\}_{K_{AS}} \rangle . c_{AB}(n).\text{case } n \text{ of } \{m\}_{K_{AB}} \text{ in } 0 \\
    & c_{AS}(x).\text{case } x \text{ of } \{y\}_{K_{AS}} \text{ in } c_{BS} \langle \{y\}_{K_{BS}} \rangle \\
    & \fbox{c_{ES} \langle \{k_{BE}\}_{K_{ES}} \rangle } \\
    & \fbox{c_{AB} \langle \{m\}_{K_{BE}} \rangle } \\
    & c_{BE}(n').\text{case } n' \text{ of } \{m\}'_{K_{BE}} \text{ in } 0 
\end{align*} \]
How does the attack work?

\[ P'_E = c'_{BS}(z).c_{BS}(z).c'_{BS}(z).c_{ES}(x').\text{case } x' \text{ of } \{y\}K_{ES} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{m\}y' \text{ in DoEvil}_m \]

\[ \rightarrow^* \quad (\nu K_A)(\nu K_{BS})(P_A \mid P_S \mid P_B)(\nu K_{AB})(\nu K_{BE}) \]

\[ c_{AB}(n).\text{case } n \text{ of } \{m\}K_{BE} \text{ in DoEvil}_m | \]

\[ c_{AS}\langle\{K_{AB}\}K_{AS}\rangle.c_{AB}(n).\text{case } n \text{ of } \{m\}K_{AB} \text{ in 0 |} \]

\[ c_{AS}(x).\text{case } x \text{ of } \{y\}K_{AS} \text{ in } c_{BS}\langle\{y\}K_{BS}\rangle | \]

\[ c_{AB}\langle\{M\}K_{BE}\rangle | \]

\[ c_{BE}(n').\text{case } n' \text{ of } \{m\}'K_{BE} \text{ in 0) } \]
How does the attack work?

\[ P'_E = c'_{BS}(z).c_{BS}(z).c'_{BS}(z).c_{ES}(x').\text{case } x' \text{ of } \{ y' \}_{K_{ES}} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{ m \}_{y'} \text{ in DoEvil}_m \]

\[ \xrightarrow{*} P'_E | (vK_{AS})(vK_{BS})(P_A | P_S | P_B) \]

\[ (vK_{AS})(vK_{BS})(vK_{AB})(vK_{BE}) \]

\[ (c_{AB}(n).\text{case } n \text{ of } \{ m \}_{K_{BE}} \text{ in DoEvil}_m | c_{AS}(\{ K_{AB} \}_{K_{AS}}).c_{AB}(n).\text{case } n \text{ of } \{ m \}_{K_{AB}} \text{ in } 0 | c_{AS}(x).\text{case } x \text{ of } \{ y \}_{K_{AS}} \text{ in } c_{BS}(\{ y \}_{K_{BS}}) | c_{AB}(\{ M \}_{K_{BE}}) | c_{BE}(n').\text{case } n' \text{ of } \{ m' \}_{K_{BE}} \text{ in } 0) \]
How does the attack work?

\[ P'_E = c'_{BS}(z).c_{BS}(z).c_{BS}(z).c_{ES}(x').c_{ES}(x').\text{case } x' \text{ of } \{y\}_{K_{ES}} \text{ in } \\
\quad c_{AB}(n).c_{AB}(n).\text{case } n \text{ of } \{m\}_{y'} \text{ in } \text{DoEvil}_m \]

\[ P'_E \mid (vK_{AS})(vK_{BS})(P_A \mid P_S \mid P_B) \rightarrow^* (vK_{AS})(vK_{BS})(vK_{AB})(vK_{BE}) \]

\[ \boxed{\text{DoEvil}_M} \]

\[ c_{AS}(\{K_{AB}\}_K_{AS}).c_{AB}(n).c_{AB}(n).\text{case } n \text{ of } \{m\}_{K_{AB}} \text{ in } 0 \]

\[ c_{AS}(x).c_{BS}(\{y\}_K_{AS} \mid c_{BS}(\{y\}_K_{BS}) | \\
c_{BE}(n').c_{BE}(n').c_{BE}(n').\text{case } n' \text{ of } \{m'\}_{K_{BE}} \text{ in } 0 \]
Outline

♦ What is spi-calculus?
  – Syntax and operational semantics
♦ Example protocol
♦ Attack against the example protocol
♦ Formalizing secrecy by non-interference
♦ Proving secrecy by hedged bisimulations
♦ Conclusions
Formalizing secrecy by non-interference

- "Definition": Process P keeps message x totally secret if \([M/x]P\) and \([N/x]P\) are "equivalent" for any M and N
  
  Cf. partial secrecy: \([M/x]P\) and \([N/x]P\) are equivalent for any M and N satisfying some condition (e.g., \(M \mod 2 = N \mod 2\))

- What equivalence should we take?  
  \(\Rightarrow\) (Strong) barbed equivalence
Definitions (1/2)

♦ Process $P$ immediately exhibits input barb $c$, written $P \downarrow c$, if

$$P \equiv (\nu x_1)...(\nu x_n)(c(y).Q | R)$$

for some $x_1, ..., x_n$ (distinct from $c$), $y$, $Q$ and $R$. Similar for output.

♦ A (strong) barbed simulation $S$ is a binary relation on processes such that $P S Q$ implies:
  - for each barb $\beta$, if $P \downarrow \beta$, then $Q \downarrow \beta$, and
  - if $P \rightarrow P'$, then $Q \rightarrow Q'$ and $P' S Q'$ for some $Q$

♦ $S$ is a barbed bisimulation if both $S$ and $S^{-1}$ are barbed simulations
Definitions (2/2)

♦ Barbed bisimilarity is the largest barbed bisimulation
  – Equals the union of all barbed bisimulations, which is also a barbed bisimulation

♦ Processes $P$ and $Q$ are barbed equivalent if $P \parallel R$ and $Q \parallel R$ are barbed bisimilar for every $R$
Example

\((\nu k)\overline{c}\langle\{x\}_k\rangle\) keeps \(x\) totally secret.

I.e., \((\nu k)\overline{c}\langle\{M\}_k\rangle\) and \((\nu k)\overline{c}\langle\{N\}_k\rangle\) are barbed equivalent for any \(M\) and \(N\).

Proof sketch: given \(M\) and \(N\), take

\[S = \{(P, Q) \mid P \equiv (\nu k)\left[\{M\}_k/y\right]R, Q \equiv (\nu k)\left[\{N\}_k/y\right]R, k \notin \text{free}(R)\}\]

and prove it to be a barbed bisimulation by case analysis (and induction) on the reduction rules.
Example

- \( P = (\nu k)(\overline{c}(\{x\}_k) \mid c(y).\text{case } y \text{ of } \{z\}_k \text{ in } \overline{c}(k)) \)

does not keep \( x \) totally secret. Indeed, \([M/x]P\) and \([N/x]P\) are not barbed equivalent for any \( M \neq N \).

Proof: given \( M \) and \( N \), take

\[ R = c(y).\overline{c}(y).c(k).\text{case } y \text{ of } \{m\}_k \text{ in } [m = M]\text{world}\langle\text{hello}\rangle \]

Cf. \( P = (\nu k)(\overline{k}(x) \mid k(y).\overline{c}(k)) \) does keep \( x \) secret
Many papers (including Abadi and Gordon's original work!) use may testing equivalence for defining secrecy by non-interference, but it is too weak.
Definitions

♦ Process $P$ may eventually exhibit barb $\beta$, written $P \downarrow \beta$, if $P \rightarrow \ldots \rightarrow P' \downarrow \beta$ for some $P'$.

♦ Processes $P$ and $Q$ are may testing equivalent if

\[(P | R) \downarrow \beta \iff (Q | R) \downarrow \beta\]

for every $R$ and $\beta$. 
So what's wrong?

♦ Surprisingly,

\[ P = (\nu d)(d\langle d \rangle \mid d() . \bar{c} \rangle) \]
and

\[ Q = (\nu d)(d\langle \rangle \mid d() . \bar{c} \rangle \mid d() . 0) \]

are may testing equivalent.

♦ As a result, processes like

if \( x > 0 \) then \( P \) else \( Q \)

are regarded as keeping \( x \) totally secret (under may testing equivalence)

♦ But the leak is possible!
Outline

♦ What is spi-calculus?
  – Syntax and operational semantics
♦ Example protocol
♦ Attack against the example protocol
♦ Formalizing secrecy by non-interference
♦ Proving secrecy by hedged bisimulations
♦ Conclusions
Hedged Bisimulation: Motivation

Direct proof of barbed equivalence is difficult because of "arbitrary R"

⇒ Devise a proof technique without "arbitrary R"

♦ What can R do?

- Gain "knowledge" by receiving from a known channel
- Send to a known channel a message synthesized from the knowledge
Definitions (1/4)

♦ A hedge $H$ is a binary relation on messages.

♦ $H \vdash M \iff N$ (messages $M$ and $N$ can be synthesized from hedge $H$) is defined by induction:

\[
\begin{align*}
(M, N) & \in \mathcal{H} & \frac{\mathcal{H} \vdash M \iff N}{(M_1, N_1) \in \mathcal{H}} & \frac{\mathcal{H} \vdash M_1 \iff N_1 \quad \mathcal{H} \vdash M_2 \iff N_2}{\mathcal{H} \vdash \{M_1\}_{M_2} \iff \{N_1\}_{N_2}} \\
\mathcal{H} \vdash \{M_1\}_{M_2} \iff \{N_1\}_{N_2} & \frac{\mathcal{H} \vdash M_2 \iff N_2}{\mathcal{H} \vdash M_1 \iff N_1} & \frac{x \not\in \text{free}(\mathcal{H})}{\mathcal{H} \vdash x \iff x}
\end{align*}
\]
A hedged simulation is a set $X$ of triples $(P, Q, \mathcal{H})$ that satisfies:

1. For any $P \rightarrow P'$, there exists some $Q'$ such that $Q \rightarrow Q'$ and $(P', Q', \mathcal{H}) \in X$.
2. If for some $\mathcal{H} \vdash c \leftrightarrow d$,

   $$P \equiv (\nu x_1) \ldots (\nu x_m)(\overline{c}(M).P_1 \mid P_2)$$

   $$x_i \not\in \{c\} \cup \text{free}(\text{fst}(\mathcal{H})),$$

   then $Q \equiv (\nu y_1) \ldots (\nu y_n)(\overline{d}(N).Q_1 \mid Q_2)$

   $$y_i \not\in \{d\} \cup \text{free}(\text{snd}(\mathcal{H}))$$

   and $(P_1 \mid P_2, Q_1 \mid Q_2, \mathcal{H} \cup (M, N)) \in X$. 
Definitions (3/4)

3. If for some $\mathcal{H} \vdash c \leftrightarrow d$,
   \[ P \equiv (\nu x_1) \ldots (\nu x_m)(c(z).P_1 \mid P_2) \]
   \[ x_i \not\in \{c\} \cup \text{free}(\text{fst}(\mathcal{H})) \]
   then $Q \equiv (\nu y_1) \ldots (\nu y_n)(d(z).Q_1 \mid Q_2)$
   \[ y_i \not\in \{d\} \cup \text{free}(\text{snd}(\mathcal{H})) \]
   and for any $\mathcal{H} \vdash M \leftrightarrow N$,
   \[ ([M/z]P_1 \mid P_2, [N/z]Q_1 \mid Q_2, \mathcal{H}) \in X. \]

4. If $\mathcal{H} \vdash M_1 \leftrightarrow N_1$ and $\mathcal{H} \vdash M_2 \leftrightarrow N_2$,
   then $M_1 = M_2$ implies $N_1 = N_2$.

5. If $\mathcal{H} \vdash \{M_1\}_{M_2} \leftrightarrow N$ and $\mathcal{H} \vdash M_2 \leftrightarrow N_2$,
   then $N = \{N_1\}_{N_2}$ for some $N_1$. 
Definitions (4/4)

♦ A hedged simulation $X$ is a hedged bisimulation if $X^{-1}$ is also a hedged simulation, where $X^{-1}$ is defined as:

$$\{(Q, P, H^{-1}) \mid (P, Q, H) \in X\}$$

♦ Hedged bisimilarity is the largest hedged bisimulation (i.e., the union of all hedged bisimulations, which is also a hedged bisimulation)

♦ Notation: $P \sim_H Q \iff (P, Q, H)$ is in the hedged bisimilarity
Caution: $\alpha$-Conversion of Hedged Bisimulation

- Every $(P, Q, H) \in X$ is regarded as $\alpha$-equivalent to
  $$(\sigma P, Q, \{ (\sigma M, N) \mid (M, N) \in H \})$$
  for every $\text{dom}(\sigma) \supseteq \text{free}(P) \cup \text{free}(\text{fst}(H))$

- Every $(P, Q, H) \in X$ is regarded as $\alpha$-equivalent to
  $$(P, \sigma Q, \{ (M, \sigma N) \mid (M, N) \in H \})$$
  for every $\text{dom}(\sigma) \supseteq \text{free}(Q) \cup \text{free}(\text{snd}(H))$

- Everything in the rest is considered "up to" this $\alpha$-equivalence
Example 1

♦ For any M and N,

\[(\nu k)\overline{c}\langle\{M\}_k\rangle.0 \sim \{(c,c)\} \ (\nu k)\overline{c}\langle\{N\}_k\rangle.0\]

Proof: take

\[X = \{((\nu k)\overline{c}\langle\{M\}_k\rangle.0, \\
(\nu k)\overline{c}\langle\{N\}_k\rangle.0, \\
\{(c, c)\})\} \cup \{(0, \\
0, \\
c, c, \{(M)_k, (N)_k\})\}\}

and check conditions 1-5.
Example 2

\((\nu k)(\nu n)\overline{c}\langle \{n\}_k \rangle. (\nu m)\overline{c}\langle m \rangle \sim \{(c, c)\}\)

\((\nu k)(\nu n)\overline{c}\langle \{n\}_k \rangle. \overline{c}\langle n \rangle\)

Proof: take

\[
X = \{((\nu k)(\nu n)\overline{c}\langle \{n\}_k \rangle. (\nu m)\overline{c}\langle m \rangle,
\quad (\nu k)(\nu n)\overline{c}\langle \{n\}_k \rangle. \overline{c}\langle n \rangle,
\quad \{(c, c)\})\}\}
\cup \{((\nu m)\overline{c}\langle m \rangle,
\quad \overline{c}\langle n \rangle,
\quad \{(c, c), \{(n)_k, \{n\}_k\}\})\}\}
\cup \{(0,
\quad 0,
\quad \{(c, c), \{(n)_k, \{n\}_k\}, (m, n)\})\}\}.
Example 3

\[ (\nu k)(\nu n)(\nu l)\bar{c}\langle\{\{n\}_k\}_l\rangle.(\nu m)\bar{c}\langle m \rangle \sim \{(c, c)\} \]
\[ (\nu k)(\nu n)\bar{c}\langle\{n\}_k\rangle.(\nu m)\bar{c}\langle m \rangle \]

Proof: take

\[ X = \{((\nu k)(\nu n)(\nu l)\bar{c}\langle\{\{n\}_k\}_l\rangle.(\nu m)\bar{c}\langle m \rangle, \]
\[ (\nu k)(\nu n)\bar{c}\langle\{n\}_k\rangle.(\nu m)\bar{c}\langle m \rangle, \]
\[ \{(c, c)\}) \}
\]
\[ \cup \{((\nu m)\bar{c}\langle m \rangle, \]
\[ (\nu m)\bar{c}\langle m \rangle, \]
\[ \{(c, c), (\{\{n\}_k\}_l, \{n\}_k)\}) \}
\]
\[ \cup \{(0, \]
\[ 0, \]
\[ \{(c, c), (\{\{n\}_k\}_l, \{n\}_k), (m, m)\}) \}.\]
Theorem

Hedged bisimilarity is sound w.r.t. barbed equivalence. I.e., if $P \sim_H Q$ for $H = \{ (x, x) \mid x \in \text{free}(P) \cup \text{free}(Q) \}$, then $P$ and $Q$ are barbed equivalent.

Proof sketch: take $S = \{ (P', Q') \mid P \sim_H Q,$

$P' \equiv (\nu x_1)...(\nu x_l) (P \mid [M_1,...,M_n/z_1,...,z_n]R),$ 

$Q' \equiv (\nu y_1)...(\nu y_m) (Q \mid [N_1,...,N_n/z_1,...,z_n]R),$ 

$H \vdash M_1 \leftrightarrow N_1, ..., H \vdash M_n \leftrightarrow N_n,$

$\text{free}(R)$ distinct from $\text{free}(P)$, $\text{free}(Q)$, and $\text{free}(H)$ $\}$ and prove it to be a barbed bisimulation by case analysis (and induction) on the reduction rules.
Real Example: Fixed Version of Previous Protocol

1. \( A \rightarrow S : \{K_{AB}, B\}K_{AS} \)
2. \( S \rightarrow B : \{K_{AB}, A\}K_{BS} \)
3. \( B \rightarrow A : \{M\}K_{AB} \)

1'. \( B \rightarrow S : \{K_{BE}, E\}K_{BS} \)
2'. \( S \rightarrow E : \{K_{BE}, B\}K_{ES} \)
3'. \( E \rightarrow B : \{M'\}K_{BE} \)
As Spi-Calculus Processes...

\[ P_A = (\nu K_{AB})c_{AS}\langle\{K_{AB}, B\}K_{AS}\rangle. \]
\[ c_{AB}(n).\text{case } n \text{ of } \{m\}K_{AB} \text{ in } 0 \]

\[ P_S = c_{AS}(x).\text{case } x \text{ of } \{y, b\}K_{AS} \text{ in } \]
\[ [b = B]c_{BS}\langle\{y\}K_{BS}\rangle \]
\[ | \quad c'_{BS}(x').\text{case } x' \text{ of } \{y', e\}K_{BS} \text{ in } \]
\[ [e = E]c_{ES}\langle\{y'\}K_{ES}\rangle \]

\[ P_B = c_{BS}(x).\text{case } x \text{ of } \{y, a\}K_{BS} \text{ in } \]
\[ [a = A]c_{AB}\langle\{z\}y\rangle \]
\[ | \quad (\nu K_{BE})c'_{BS}\langle\{K_{BE}, E\}K_{BS}\rangle. \]
\[ c_{BE}(n').\text{case } n' \text{ of } \{m'\}K_{BE} \text{ in } 0 \]
Exercise (?)

- Write down the reduction(s) of $P'_E \parallel (\forall K_{AS})(\forall K_{BS})(P_A \parallel P_S \parallel P_B)$ for the same attacker $P'_E$ as before, for the fixed version of $P_A$, $P_S$, and $P_B$. Pinpoint where the attack fails.
Claim

- \((\nu K_{AS})(\nu K_{BS})(P_A \parallel P_S \parallel P_B)\) keeps \(z\) totally secret. I.e.,
  \[P = (\nu K_{AS})(\nu K_{BS})(P_A \parallel P_S \parallel [M/z]P_B)\]
  and
  \[Q = (\nu K_{AS})(\nu K_{BS})(P_A \parallel P_S \parallel [N/z]P_B)\]
are barbed equivalent for any \(M\) and \(N\).
Proof Sketch

♦ Let $H = \{ (x, x) \mid x \in \text{free}(P) \cup \text{free}(Q) \}$

♦ We construct some hedged bisimulation $X \ni (P, Q, H)$
  – The $X$ is far from minimal, but this is fine as far as $X$ is a hedged bisimulation
    • It is a nightmare to write down minimal $X$ for real...
\[ P_A = (\nu K_{AB}) \overline{c_{AS}} \langle \{ K_{AB}, B \} \rangle_{K_{AS}} \cdot c_{AB}(n) \cdot \text{case } n \text{ of } \{ m \} \rangle_{K_{AB}} \text{ in } 0 \]

\[ P_S = c_{AS}(x) \cdot \text{case } x \text{ of } \{ y, b \} \rangle_{K_{AS}} \text{ in } [b = B] \overline{c_{BS}} \langle \{ y \} \rangle_{K_{BS}} \]

\[ | c'_{BS}(x') \cdot \text{case } x' \text{ of } \{ y', e \} \rangle_{K_{BS}} \text{ in } [e = E] \overline{c_{ES}} \langle \{ y' \} \rangle_{K_{ES}} \]

\[ P'_{S_0} \]

\[ P'_{S_1} \]

\[ P'_{S_2} \]

\[ P'_{S_3} \]
\[ P_B = c_{BS}(x) \cdot \text{case } x \text{ of } \{y, a\}_{K_{BS}} \text{ in } [a = A] \overline{c_{AB}}(\{z\}y) \]

\[ | (\nu K_{BE}) \overline{c_{BS}}(\{K_{BE}, E\}_{K_{BS}}) \cdot c_{BE}(n') \cdot \text{case } n' \text{ of } \{m\}_{K_{BE}} \text{ in } 0 \]
\[ X = \{ (P', Q', H') | \]

\[ P' \equiv (\nu c_1)...(\nu c_u) \]

\[ ( [M_1/n]PA_i | [M_2/x]PS_j | [M_3,A/x',e]P'S_k | [M_4,E,M/x,a,z]PB_l | [M_5/n']PB_m ), \]

\[ Q' \equiv (\nu d_1)...(\nu d_v) \]

\[ ( [N_1/n]PA_i | [N_2/x]PS_j | [N_3,A/x',e]P'S_k | [N_4,E,N/x,a,z]PB_l | [N_5/n']PB_m ), \]

\[ H' \subseteq H \cup \{ (\{K_{AB},B\}_{KAS}, \{K_{AB},B\}_{KAS}), (\{K_{AB}, A\}_{KBS}, \{K_{AB}, A\}_{KBS}), (\{M\}_{KAB}, \{N\}_{KAB}), (\{K_{BE}, E\}_{KBS}, \{K_{BE}, E\}_{KBS}), (\{K_{BE}, B\}_{KES}, \{K_{BE}, B\}_{KES} ) \}, \]

\[ H' \vdash M_w \leftrightarrow N_w \text{ for } w = 1, 2, 3, 4, 5, \]

\[ c_1, ..., c_u \notin \text{free(fst(H'))}, \]

\[ d_1, ..., d_v \notin \text{free(snd(H'))} \} \]
Exercise (?)

- Try to prove the total secrecy of $z$ in the original version of this protocol by means of hedged bisimulation. Explain how the "proof" fails.
Side Step II: Completeness of Hedged Bisimulation

Conjecture:
Hedged bisimilarity is complete with respect to barbed equivalence.
I.e., if $P$ and $Q$ are barbed equivalent, then $P \sim_H Q$ for
$$H = \{ (x, x) \mid x \in \text{free}(P) \cup \text{free}(Q) \}$$

- Proved for "structurally image finite" processes, but not for the general case (to my knowledge)
Outline

♦ What is spi-calculus?
  – Syntax and operational semantics
♦ Example protocol
♦ Attack against the example protocol
♦ Formalizing secrecy by non-interference
♦ Proving secrecy by hedged bisimulations
♦ Conclusions
Other Topics in Spi-Calculus

♦ Other bisimulations [Abadi-Gordon 98]
  [Boreale-DeNicola-Pugliese 99]
  [Elkjær-Höhle-Hüttel-Overgård 99]
  – More complex and "less complete"

♦ Secrecy by typing [Abadi 97]
  [Abadi-Blanchet 01]

♦ Authenticity by typing [Gordon-Jeffery 01]
  [Gordon-Jeffery 02] [Blanchet 02]

Cf. http://www.soe.ucsc.edu/~abadi/
http://www.di.ens.fr/~blanchet/
http://netlib.bell-labs.com/who/ajeffrey/ etc.