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

\[ M, N ::= \\
   x \\
   \{M_1, \ldots, M_n\}_N \\
 P, Q, R ::= \\
   0 \\
   \overline{M}(N).P \\
   M(x).P \\
   P | Q \\
 (\nu x)P \\
 !P \\
 \text{case } M \text{ of } \{x_1, \ldots, x_n\}_N \text{ in } P \\
 [M = N]P \]

message  
name  
ciphertext  
process  
inaction  
sending  
receiving  
parallel composition  
restriction  
replication  
decryption  
matching
Operational Semantics (1/2): Structural Equivalence

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

\[
[M = M]P \equiv P \quad \neg P \equiv P \quad \neg \neg P
\]

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

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

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

\[
\frac{P \equiv P'}{(\nu x)P \equiv (\nu x)P'}
\]

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

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

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

\[ P \equiv P' \quad P' \rightarrow Q' \quad Q' \equiv Q \]

\[ P \rightarrow Q \]

\[ P \rightarrow P' \]

\[ P \mid Q \rightarrow P' \mid Q \]

\[ (\nu x)P \rightarrow (\nu x)P' \]
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. $A \rightarrow S : \{K_{AB}\}K_{AS}$
2. $S \rightarrow B : \{K_{AB}\}K_{BS}$
3. $B \rightarrow A : \{M\}K_{AB}$

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

$$P_S = c_{AS}(x).\text{case } x \text{ of } \{y\}K_{AS} \text{ in } \overline{c_{BS}}\langle\{y\}K_{BS}\rangle$$

$$P_B = c_{BS}(x).\text{case } x \text{ of } \{y\}K_{BS} \text{ in } \overline{c_{AB}}\langle\{M\}y\rangle$$

The whole system is:

$$(\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid P_B)$$
How does the protocol run?

\((\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid P_B)\)

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

\(\left(\overline{c_{AS}}\left\langle\{K_{AB}\}K_{AS}\right\rangle , c_{AB}(n)\right)\text{. case } n \text{ of } \{m\}K_{AB} \text{ in } 0\) | 
\(\overline{c_{AS}(x)\text{. case } x \text{ of } \{y\}K_{AS} \text{ in } \overline{c_{BS}(\{y\}K_{BS})}}\) |
\(c_{BS}(x)\text{. case } x \text{ of } \{y\}K_{BS} \text{ in } \overline{c_{AB}(\{M\}y)}\)

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

\(\left(\overline{c_{AB}(n)\text{. case } n \text{ of } \{m\}K_{AB} \text{ in } 0}\right) | 
\text{ case } \{K_{AB}\}K_{AS} \text{ of } \{y\}K_{AS} \text{ in } \overline{c_{BS}(\{y\}K_{BS})}| \n\overline{c_{BS}(x)\text{. case } x \text{ of } \{y\}K_{BS} \text{ in } \overline{c_{AB}(\{M\}y)}\})\)

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

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

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

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

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

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

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

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

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

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

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

\[
(\nu K_{AS})(\nu K_{BS})(\nu K_{AB})0
\]

\[
(\nu K_{AS})(\nu K_{BS})(\nu K_{AB})0
\]

\[
(\nu K_{AS})(\nu K_{BS})(\nu K_{AB})0
\]

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

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

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

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

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

\[
(\langle c_{AB}(n)\rangle).\text{case } n \text{ of } \{m\}_{K_{AB}} \text{ in } 0 | \\
\text{ case } \{K_{AB}\}_{K_{BS}} \text{ of } \{y\}_{K_{BS}} \text{ in } c_{AB}\langle \{M\}_y \rangle
\]

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

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

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

\[
\text{ case } \{M\}_{K_{AB}} \text{ of } \{m\}_{K_{AB}} \text{ in } 0
\]

\[\equiv (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})0
\]
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)

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

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

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

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

♦ Write down the reduction of

$$(\nu K_{AS})(\nu K_{BS})(\nu K_{ES})(P_A \mid P_S \mid P_B \mid P_E).$$
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'a. $B \rightarrow E(S) : \{K_{BE}\}K_{BS}$
2. $E(S) \rightarrow B : \{K_{BE}\}K_{BS}$
1'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?

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

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

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

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

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

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

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

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

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

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

\[ \rightarrow \]

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

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

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

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

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

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

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

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

\[ P'_E = c'_{BS}(z).c_{BS}^{-1}(z).c'_BS(z) \cdot c_{ES}(x').\text{case } x' \text{ of } \{y'\}_{K_{ES}} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{m\}_{y'} \text{ in DoEvil}_m \]

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

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

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

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

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

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

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

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

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

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

\[ \rightarrow^* \]

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

\[ c_{BS}(\{K_{BE}\}_K_{BS}).c_{ES}(x').\text{case } x' \text{ of } \{ y' \}_K_{ES} \text{ in } c_{AB}(n).\text{case } n \text{ of } \{ m \}_y \text{ in DoEvil}_n | \]

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

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

\[ c'_{BS}(x').\text{case } x' \text{ of } \{ y' \}_K_{BS} \text{ in } \overline{c_{ES}(\{y'\}_K_{ES})} | \]

\[ \overline{c_{AB}(\{M\}_K_{BE})} | \]

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

\[
P'_E = c'_{BS}(z).c_{BS}(z).c'_{BS}(z).
\]

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

\[
c_{AB}(n).\text{case } n \text{ of } \{m\}'_{y'} \text{ in } \text{DoEvil}_m
\]

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

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

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

\[
c_{AB}(n).\text{case } n \text{ of } \{m\}'_{y'} \text{ in } \text{DoEvil}_m
\]

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

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

\[
\overline{c}_{ES}\langle\{K_{BE}\}_{K_{ES}}\rangle
\]

\[
\overline{c}_{AB}\langle\{M\}_{K_{BE}}\rangle
\]

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

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

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

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

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

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

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

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

\[
P'_E \rightarrow^* (\nu K_{AS})(\nu K_{BS})(P_A | P_S | P_B) \\
    (\nu K_{AS})(\nu K_{BS})(\nu K_{AB})(\nu K_{BE}) \\
    (\text{DoEvil}_M) \\
    c_{AS}(\{K_{AB}\}_K_{AS}).c_{AB}(n).\text{case } n \text{ of } \{ m \}_{K_{AB}} \text{ in } 0 \\
    c_{AS}(x).\text{case } x \text{ of } \{ y \}_K_{AS} \text{ in } \overline{c_{BS}(\{ y \}_K_{BS})} \\
    c_{BE}(n').\text{case } n' \text{ of } \{ m' \}_{K_{BE}} \text{ in } 0 
\]
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

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

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 | R and Q | 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

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

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 \[m = M\]world} \langle \text{hello} \rangle \]

Cf. \[ P = (\nu k)(\overline{k}(x) \mid k(y).\overline{c}(k)) \] does keep \( x \) secret
Side Step: The Vice of May Testing Equivalence

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 $(P \mid R) \Downarrow \beta \iff (Q \mid R) \Downarrow \beta$ for every $R$ and $\beta$. 
So what's wrong?

- Surprisingly, \( P = (\nu d)(\overline{d}() \mid d() . \overline{c}{}) \)
  and
  \( Q = (\nu d)(\overline{d}() \mid d() . \overline{c}{}) \mid d() . 0 \)
  are may testing equivalent.

- As a result, processes like
  \[
  \text{if } x > 0 \text{ then } P \text{ else } Q
  \]
  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:

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

1. For any $P \rightarrow P'$, there exists some $Q'$ such that $Q \rightarrow Q'$ and $(P', Q', H) \in X$.
2. If for some $H \vdash c \leftrightarrow d$,
   
   $$P \equiv (\nu x_1) \ldots (\nu x_m)(\overline{c}(M).P_1 \mid P_2)$$

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

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

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

   and $(P_1 \mid P_2, Q_1 \mid Q_2, H \cup (M, N)) \in X$. 
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 $N_1$. 
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

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

\[ (\nu k)(\nu n)(\nu l) c\langle \{n\}_k \rangle_l \cdot (\nu m) c\langle m \rangle \sim \{(c, c)\} \]

\[ (\nu k)(\nu n) c\langle \{n\}_k \rangle \cdot (\nu m) c\langle m \rangle \]

Proof: take

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

$$S = \{ (P', Q') \mid P \sim_H Q,$$

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

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

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

$$\text{free}(R) \text{ distinct from free}(P), \text{free}(Q), \text{and free}(H) \}$$

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...

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

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

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

♦ Write down the reduction(s) of

$$P'_E \mid (\forall K_{AS})(\forall K_{BS})(P_A \mid P_S \mid P_B)$$

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

- \((\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid P_B)\)
  keeps \(z\) totally secret. I.e.,

\[
P = (\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid [M/z]P_B)
\]

and

\[
Q = (\nu K_{AS})(\nu K_{BS})(P_A \mid P_S \mid [N/z]P_B)
\]

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

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

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

\[ P'_{S_0} \]

\[ P'_{S_1} \]

\[ P'_{S_2} \]

\[ P'_{S_3} \]
\[ P_B = \text{case } x \text{ of } \{y, a\}_K \text{ in } [a = A] \overline{c_{AB} \langle \{z\}_y \rangle} \]

\[ \mid (\nu K_{BE}) \overline{c_{BS} \langle \{K_{BE}, E\}_K \rangle} \cdot c_{BE}(n'). \text{case } n' \text{ of } \{m'\}_K \text{ in } 0 \]
\( X = \{ (P', Q', H') \mid \\
\quad P' \equiv (v c_1)\ldots(v c_u) \\
\quad ( [M_1/n]P_{A_i} \mid [M_2/x]P_{S_i} \mid [M_3,A/x',e]P'_{S_k} \mid \\
\quad [M_4,E,M/x,a,z]P_{B_l} \mid [M_5/n']P'_{B_m}) , \\
\quad Q' \equiv (v d_1)\ldots(v d_v) \\
\quad ( [N_1/n]P_{A_i} \mid [N_2/x]P_{S_i} \mid [N_3,A/x',e]P'_{S_k} \mid \\
\quad [N_4,E,N/x,a,z]P_{B_l} \mid [N_5/n']P'_{B_m} ) , \\
\quad H' \subseteq H \cup \{ (\{K_{AB},B\}_{K_{AS}}, \{K_{AB},B\}_{K_{AS}}) , \\
\quad (\{K_{AB}, A\}_{K_{BS}}, \{K_{AB}, A\}_{K_{BS}}), \\
\quad (\{M\}_{K_{AB}}, \{N\}_{K_{AB}}), \\
\quad (\{K_{BE}, E\}_{K_{BS}}, \{K_{BE}, E\}_{K_{BS}}), \\
\quad (\{K_{BE}, B\}_{K_{ES}}, \{K_{BE}, B\}_{K_{ES}}) \} , \\
\quad H' \vdash M_w \leftrightarrow N_w \text{ for } w = 1, 2, 3, 4, 5, \\
\quad c_1, \ldots, c_u \notin \text{free}(\text{fst}(H')) , \\
\quad d_1, \ldots, d_v \notin \text{free}(\text{snd}(H')) \} \)
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.