

# Spi Calculus Reading Guide

Matt, Ed, Nishant, Ilya

October 29, 2004

## 1 Introduction

Spi Calculus is an extension of pi calculus with cryptographic primitives. It is designed for describing and analyzing security protocols, such as those for authentication and for electronic commerce. The notation that will be used is listed below.

Names are used for communication channels and we assume there are an infinite set of these. The same is true for variables as well.

## 2 Notation

$L, M, N ::=$	terms
$n$	name of channel
$(M, N)$	pair
$0$	zero
$suc(M)$	successor
$x$	variable
$P, Q, R ::=$	processes
$\bar{m}(N).P$	output
$m(x).P$	input
$P \mid Q$	composition
$(\nu n)P$	restriction
$!P$	replication
$[M \text{ is } N] P$	match
$0$	nil
$\text{let } (x, y) = M \text{ in } P$	pair splitting
$\text{case } M \text{ of } 0 : P \text{ suc}(x) : Q$	integer case
$P \simeq Q$	indistinguishable

The first couple terms are self-explanatory.  $suc(M)$  is how spi calculus handles numbers. For example,  $suc(0) = 1$  and  $suc(suc(0)) = 2$ .  $\bar{m}(N).P$  means that term  $N$  is communicated over channel  $m$  and then process  $P$  runs.  $m(x).P$

means that if an interaction occurs in which  $N$  is communicated on  $m$ , then process  $P$  runs with  $N$  substituted for  $x$ .  $P \mid Q$  means that processes  $P$  and  $Q$  are running in parallel.  $(\nu n)P$  means that a new, private  $n$  is created and behaves as  $P$ .  $!P$  means that an infinite number of copies of  $P$  are running in parallel.  $[M \text{ is } N] P$  behaves as  $P$  only if  $M$  and  $N$  are the same, otherwise it does nothing. *Let*  $(x,y) = M$  *in*  $P$  behaves as  $P$  with  $x$  set to  $N$  and  $y$  set to  $L$  if term  $M$  is the pair  $(N,L)$ , otherwise, the process does nothing. *Case*  $M$  *of*  $\theta : P \text{ suc}(x) : Q$  behaves as  $P$  if term  $M$  is  $\theta$ , as  $Q$  with  $x$  set to  $N$  if  $M$  is  $\text{suc}(N)$ , otherwise, the process does nothing.  $P \simeq Q$  means that the behaviors of processes  $P$  and  $Q$  are indistinguishable.

There are two extra notations that are needed for this paper. Those above are just for the Pi Calculus component. The Spi Calculus part contains all of the above as well as those mentioned below.

$\{M\}_N$	shared-key encryption
<i>case</i> $L$ <i>of</i> $\{x\}_N$ <i>in</i> $P$	shared-key decryption

The term  $\{M\}_N$  represents the ciphertext obtained by encrypting the term  $M$  under the key  $N$  using a shared-key cryptosystem. The process *case*  $L$  *of*  $\{x\}_N$  *in*  $P$  attempts to decrypt the term  $L$  with the key  $N$ . If  $L$  is a ciphertext of the form  $\{M\}_N$ , then the process behaves as  $P$  with  $x$  set to  $M$ , otherwise, the process does nothing.

### 3 Example

Let's say we want to send an encrypted message  $M$  from  $A$  to  $B$ . We could write it in pseudo-English as:

Message 1  $A \rightarrow B$ :  $\{M\}_{K_{AB}}$  on  $c_{AB}$

In the spi calculus, we write:

1.  $A(M) \triangleq \overline{c_{AB}}\langle \{M\}_{K_{AB}} \rangle$
2.  $B \triangleq c_{AB}(x). \text{case } x \text{ of } \{y\}_{K_{AB}} \text{ in } F(M)$
3.  $\text{Inst}(M) \triangleq (\nu K_{AB})(A(M) \mid B)$

Part 1 says  $A$  sends message  $M$  encrypted with key  $K_{AB}$  on channel  $c_{AB}$

Part 2 states that  $B$  is listening for a message on  $c_{AB}$  and when it receives one, it attempts to decrypt it using key  $K_{AB}$ . If it succeeds, then  $B$  applies a function  $F$  to the result.

Part 3 says that it instantiates a new instance of what is on the right hand side. This means a new key is created and then  $A(M)$  and  $B$  run in parallel.