

# Security Despite System Compromise

Information Security & Cryptography Group

Martin Grochulla

June 17, 2008

# Introduction

A simple example



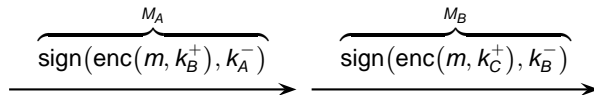
A



B



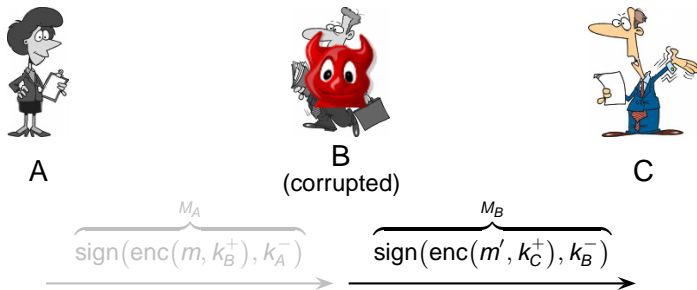
C



Can C assume that – after receiving  $M_B$  from B – A has sent  $M_A$  to B?

# Introduction

## A simple example



Can C assume that – after receiving  $M_B$  from B – A has sent  $M_A$  to B? — **No, in a compromised system, he cannot!** ...  
How can we solve this problem?

- ▶ Introduction
- ▶ The approach of Fournet et al.
- ▶ Our goals and our approach
- ▶ A simple example
- ▶ Zero-knowledge
- ▶ A simple example II

Corin, Deniélou, Fournet, Bhargavan, Liefer:  
Secure Implementations for Typed Session Abstractions  
*Computer Security Foundations Symposium, 2007*

- ▶ add message descriptor (= label) to each message
- ▶ make each label unique (timestamp, session id, ...)
- ▶ sign each label and forward signed labels along with the original message
- ▶ **no cryptography**
- ▶ safety properties proved about labels, not messages

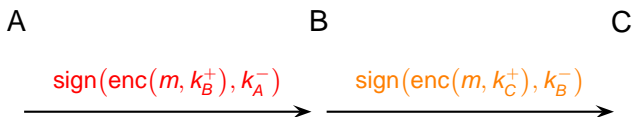
# Our goals and our approach

- ▶ We want to use cryptographic terms, names, . . .
- ▶ We want to guarantee safety properties about these terms (in the presence of compromised participants)

# Our goals and our approach

- ▶ We want to use cryptographic terms, names, . . .
- ▶ We want to guarantee safety properties about these terms (in the presence of compromised participants)
  
- ▶ Make usage of zero-knowledge proofs
- ▶ Model safety properties by annotations in the calculus
- ▶ Use a type system to analyze annotations statically

# A simple example (syntax)



Rewritten in applied pi-calculus:

$A = 1 : \text{out}(ch, \text{sign}(\text{enc}(m, k_B^+), k_A^-))$

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

$C = 2 : \text{in}(ch, y).$

let  $y_1 = \text{check}(y, k_B^+)$  then

let  $y_2 = \text{dec}(y_1, k_C^-)$  then ...

$P = \text{new } k_A^- . \text{new } k_B^- . \text{new } k_C^- . \text{new } m . (A|B|C)$

# A simple example



A

1 : out ( $ch$ ,  $\text{sign}(\text{enc}(m, k_B^+), k_A^-)$ )



B

1 : in ( $ch$ ,  $x$ ) .

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

2 : out ( $ch$ ,  $\text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$ )

# A simple example



A



B

1 : out ( $ch$ , sign(enc( $m$ ,  $k_B^+$ ),  $k_A^-$ ))

1 : in ( $ch$ ,  $x$ ) .

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

2 : out ( $ch$ , sign(enc( $x_2$ ,  $k_C^+$ ),  $k_B^-$ ))

$k : \text{out}(a, M) . P \mid k : \text{in}(a, x) . Q \rightarrow P \mid Q\{M/x\}$

# A simple example



A



B

let  $x_1 = \text{check}(\text{sign}(\text{enc}(m, k_B^+), k_A^-), k_A^+)$  then  
let  $x_2 = \text{dec}(x_1, k_B^-)$  then  
2 : out ( $ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$ )

$k : \text{out}(a, M) . P \mid k : \text{in}(a, x) . Q \rightarrow P \mid Q \{M/x\}$

# A simple example



B

let  $x_1 = \text{check}(\text{sign}(\text{enc}(m, k_B^+), k_A^-), k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

2 : out ( $ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$ )

# A simple example



B

let  $x_1 = \text{check}(\text{sign}(\text{enc}(m, k_B^+), k_A^-), k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

2 : out ( $ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$ )

let  $x = g(\tilde{M})$  then  $P \rightarrow P\{N/x\}$ , if  $g(\tilde{M}) \Downarrow N$   
 $\text{check}(\text{sign}(N, K^-), K^+) \Downarrow N$

# A simple example



B

let  $x_2 = \text{dec}(\text{enc}(m, k_B^+), k_B^-)$  then

2 : out ( $ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$ )

let  $x = g(\tilde{M})$  then  $P \rightarrow P\{N/x\}$ , if  $g(\tilde{M}) \Downarrow N$

check( $\text{sign}(N, K^-), K^+$ )  $\Downarrow N$

# A simple example



B

let  $x_2 = \text{dec}(\text{enc}(m, k_B^+), k_B^-)$  then

2 : out ( $ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$ )

let  $x = g(\tilde{M})$  then  $P \rightarrow P\{N/x\}$ , if  $g(\tilde{M}) \Downarrow N$   
 $\text{dec}(\text{enc}(N, K^+), K^-) \Downarrow N$

# A simple example



B

2 : out ( $ch, \text{sign}(\text{enc}(m, k_C^+), k_B^-)$ )

let  $x = g(\tilde{M})$  then  $P \rightarrow P\{N/x\}$ , if  $g(\tilde{M}) \Downarrow N$   
 $\text{dec}(\text{enc}(N, K^+), K^-) \Downarrow N$

# A simple example



B

2 : out ( $ch$ ,  $\text{sign}(\text{enc}(m, k_C^+), k_B^-)$ )



C

2 : in ( $ch$ ,  $y$ ).

let  $y_1 = \text{check}(y, k_B^+)$  then

let  $y_2 = \text{dec}(y_1, k_C^-)$  then ...

# A simple example



B



C

$2 : \text{out}(ch, \text{sign}(\text{enc}(m, k_C^+), k_B^-))$

$2 : \text{in}(ch, y).$

let  $y_1 = \text{check}(y, k_B^+)$  then

let  $y_2 = \text{dec}(y_1, k_C^-)$  then ...

$k : \text{out}(a, M) . P \mid k : \text{in}(a, x) . Q \rightarrow P \mid Q\{M/x\}$

# A simple example



B



C

let  $y_1 = \text{check}(\text{sign}(\text{enc}(m, k_C^+), k_B^-), k_B^+)$  then  
let  $y_2 = \text{dec}(y_1, k_C^-)$  then ...

$$k : \text{out}(a, M) . P \mid k : \text{in}(a, x) . Q \rightarrow P \mid Q\{M/x\}$$

# A simple example



C

let  $y_1 = \text{check}(\text{sign}(\text{enc}(m, k_C^+), k_B^-), k_B^+)$  then  
let  $y_2 = \text{dec}(y_1, k_C^-)$  then ...

# A simple example



C

let  $y_1 = \text{check}(\text{sign}(\text{enc}(m, k_C^+), k_B^-), k_B^+)$  then  
let  $y_2 = \text{dec}(y_1, k_C^-)$  then ...

let  $x = g(\tilde{M})$  then  $P \rightarrow P\{N/x\}$ , if  $g(\tilde{M}) \Downarrow N$   
 $\text{check}(\text{sign}(N, K^-), K^+) \Downarrow N$

# A simple example



C

let  $y_2 = \text{dec}(\text{enc}(m, k_C^+), k_C^-)$  then ...

let  $x = g(\tilde{M})$  then  $P \rightarrow P\{N/x\}$ , if  $g(\tilde{M}) \Downarrow N$   
check( $\text{sign}(N, K^-), K^+$ )  $\Downarrow N$

# A simple example



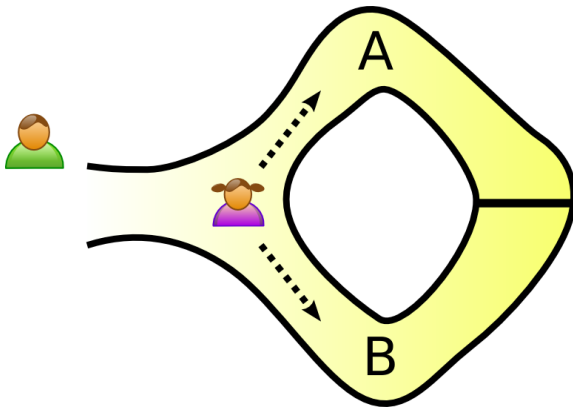
C

let  $y_2 = \text{dec}(\text{enc}(m, k_C^+), k_C^-)$  then ...

let  $x = g(\tilde{M})$  then  $P \rightarrow P\{N/x\}$ , if  $g(\tilde{M}) \Downarrow N$   
 $\text{dec}(\text{enc}(N, K^+), K^-) \Downarrow N$

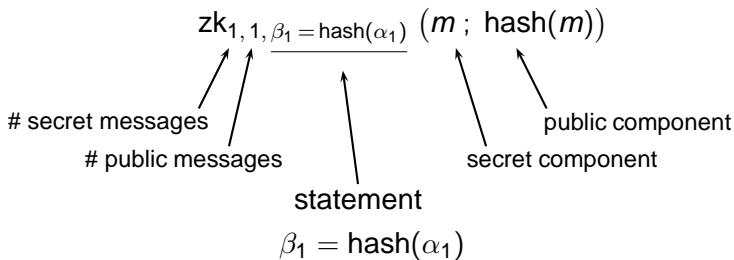
# Zero-knowledge

How zero-knowledge proofs work



# Zero-knowledge

How zero-knowledge proofs work in the calculus



How zero-knowledge proofs work in the calculus

$$\text{zk}_{1, 1, \beta_1 = \text{hash}(\alpha_1)} (m ; \text{hash}(m))$$

$$\beta_1 = \text{hash}(\alpha_1)$$

How zero-knowledge proofs work in the calculus

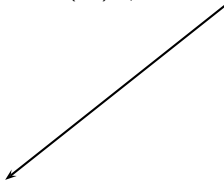
$$\text{zk}_{1,1,\beta_1 = \text{hash}(\alpha_1)} (m ; \text{hash}(m))$$



$$\beta_1 = \text{hash}(m)$$

How zero-knowledge proofs work in the calculus

$$zk_{1, 1, \beta_1 = \text{hash}(\alpha_1)} (m ; \text{hash}(m))$$



$$\text{hash}(m) = \text{hash}(m)$$

How zero-knowledge proofs work in the calculus

$$\text{zk}_{1, 1, \beta_1 = \text{hash}(\alpha_1)} (m ; \text{hash}(m))$$

$$\text{hash}(m) = \text{hash}(m) \quad \Downarrow \quad \text{true}$$

# A simple example

Compiling a zero-knowledge proof for output 2:



$B = 1 : \text{in}(ch, x) .$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

2 : out ( $ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$ )

$P = \text{new } k_A^- . \text{new } k_B^- . \text{new } k_C^- . \text{new } m . (A|B|C)$

we know

- ▶ restricted names at output 2:  $k_A^-, k_B^-, k_C^-, m$
- ▶ bound variables at output 2:  $x, x_1, x_2$

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  ?  $x_1$  ?  $x_2$  ?

$B = 1 : \text{in}(ch, x) .$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:
  
- ▶ statement:

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  ?  $x_1$  ?  $x_2$  ?

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:
  
- ▶ statement:

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables: **x No!**  $x_1$  ?  $x_2$  ?

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:
  
- ▶ statement:

# A simple example

- ▶ names we have to keep secret:  $k_A^-, k_B^-, k_C^-, m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  ?  $x_2$  ?

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:
  
- ▶ statement:

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No!  $x_2$  ?

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:
  
- ▶ statement:

# A simple example

- ▶ names we have to keep secret:  $k_A^-, k_B^-, k_C^-, m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  ?

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:  $x, x_1, k_A^+$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-, k_B^-, k_C^-, m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  ?

$B = 1 : \text{in}(ch, x) .$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:  $x, x_1, k_A^+$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  Yes!

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:
- ▶ public component:  $x, x_1, k_A^+$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  Yes.

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:  $x_2, k_B^-$
- ▶ public component:  $x, x_1, k_A^+$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3) \wedge \alpha_1 = \text{dec}(\beta_2, \alpha_2)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  Yes.

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:  $x_2, k_B^-$
- ▶ public component:  $x, x_1, k_A^+$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3) \wedge \alpha_1 = \text{dec}(\beta_2, \alpha_2)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  Yes.

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:  $x_2, k_B^-$
- ▶ public component:  $x, x_1, k_A^+, k_C^+, \text{enc}(x_2, k_C^+)$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3) \wedge \alpha_1 = \text{dec}(\beta_2, \alpha_2) \wedge \beta_5 = \text{enc}(\alpha_1, \beta_4)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  Yes.

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:  $x_2, k_B^-$
- ▶ public component:  $x, x_1, k_A^+, k_C^+, \text{enc}(x_2, k_C^+)$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3) \wedge \alpha_1 = \text{dec}(\beta_2, \alpha_2) \wedge \beta_5 = \text{enc}(\alpha_1, \beta_4)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-$ ,  $k_B^-$ ,  $k_C^-$ ,  $m$
- ▶ what about the bound variables:  $x$  No.  $x_1$  No.  $x_2$  Yes.

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:  $x_2, k_B^-$
- ▶ public component:  $x, x_1, k_A^+, k_C^+, \text{enc}(x_2, k_C^+), k_B^+, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3) \wedge \alpha_1 = \text{dec}(\beta_2, \alpha_2) \wedge \beta_5 = \text{enc}(\alpha_1, \beta_4)$   
 $\wedge \beta_5 = \text{check}(\beta_7, \beta_6)$

# A simple example

- ▶ names we have to keep secret:  $k_A^-, k_B^-, k_C^-, m$
- ▶ variables we have to keep secret:  $x_2$

$B = 1 : \text{in}(ch, x).$

let  $x_1 = \text{check}(x, k_A^+)$  then

let  $x_2 = \text{dec}(x_1, k_B^-)$  then

$2 : \text{out}(ch, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-))$

- ▶ secret component:  $x_2, k_B^-$
- ▶ public component:  $x, x_1, k_A^+, k_C^+, \text{enc}(x_2, k_C^+), k_B^+, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$
- ▶ statement:  $\beta_2 = \text{check}(\beta_1, \beta_3) \wedge \alpha_1 = \text{dec}(\beta_2, \alpha_2) \wedge \beta_5 = \text{enc}(\alpha_1, \beta_4)$   
 $\wedge \beta_5 = \text{check}(\beta_7, \beta_6)$

# A simple example

- ▶ secret component  $\tilde{N}$ :  $x_2, k_B^-$
- ▶ public component  $\tilde{M}$ :  $x, x_1, k_A^+, k_C^+, \text{enc}(x_2, k_C^+), k_B^+, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-)$
- ▶ statement  $S$ :  $\beta_2 = \text{check}(\beta_1, \beta_3) \wedge \alpha_1 = \text{dec}(\beta_2, \alpha_2) \wedge \beta_5 = \text{enc}(\alpha_1, \beta_4) \wedge \beta_5 = \text{check}(\beta_7, \beta_6)$

$$2 : \text{out} \left( ch, \text{zk}_{2,7,S}(\tilde{N}; \tilde{M}) \right)$$

# Simple example transformed

$A = 1 : \text{out}(ch, \langle zk_{1,4,\beta_4=\text{check}(\beta_3,\beta_1)\wedge\beta_4=\text{enc}(\alpha_1,\beta_2)}$   
 $(m; k_A^+, k_B^+, \text{sign}(\text{enc}(m, k_B^+), k_A^-), \text{enc}(m, k_B^+)) \rangle)$

$B = 1 : \text{in}(ch, x') . \text{let } \langle x'_1 \rangle = x' \text{ in let } \langle x'_{1,1}, x'_{1,2}, x'_{1,3}, x'_{1,4} \rangle = \text{public}_4(x'_1) \text{ then}$   
 $\text{let } \langle x''_{1,3}, x''_{1,4} \rangle = \text{ver}_{1,4,2,\beta_4=\text{check}(\beta_3,\beta_1)\wedge\beta_4=\text{enc}(\alpha_1,\beta_2)}(x'_1, k_A^+, k_B^+)$   
 $\text{then let } x = \text{id}(x''_{1,3}) \text{ then let } x_1 = \text{check}(x, k_A^+) \text{ then}$   
 $\text{let } x_2 = \text{dec}(x_1, k_B^-) \text{ then } 2 : \text{out}(ch,$   
 $\langle zk_{2,7,\beta_6=\text{check}(\beta_5,\beta_1)\wedge\beta_6=\text{enc}(\alpha_1,\beta_2)\wedge\alpha_1=\text{dec}(\beta_7,\alpha_2)\wedge\beta_4=\text{check}(\beta_7,\beta_3)}$   
 $(x_2, k_B^-; k_B^+, k_C^+, k_A^+, x, \text{sign}(\text{enc}(x_2, k_C^+), k_B^-), \text{enc}(x_2, k_C^+), x_1), x') \rangle)$

$C = 2 : \text{in}(ch, y') . \text{let } \langle y'_1, y'_2 \rangle = y' \text{ in let } \langle y'_{2,1}, y'_{2,2}, y'_{2,3}, y'_{2,4} \rangle = \text{public}_4(y'_1) \text{ then}$   
 $\text{let } \langle y'_{1,1}, y'_{1,2}, y'_{1,3}, y'_{1,4}, y'_{1,5}, y'_{1,6}, y'_{1,7} \rangle = \text{public}_7(y'_1) \text{ then let } \langle y''_{2,3}, y''_{2,4} \rangle =$   
 $\text{ver}_{1,4,2,\beta_4=\text{check}(\beta_3,\beta_1)\wedge\beta_4=\text{enc}(\alpha_1,\beta_2)}(x'_2, k_A^+, k_B^+) \text{ let } \langle y''_{1,5}, y''_{1,6}, y''_{1,7} \rangle =$   
 $\text{ver}_{2,7,4,\beta_6=\text{check}(\beta_5,\beta_1)\wedge\beta_6=\text{enc}(\alpha_1,\beta_2)\wedge\alpha_1=\text{dec}(\beta_7,\alpha_2)\wedge\beta_4=\text{check}(\beta_7,\beta_3)}$   
 $(x'_1, k_B^+, k_C^+, k_A^+, y''_{2,3}) \text{ then let } x = \text{id}(x''_{1,3}) \text{ then}$   
 $\text{let } y_1 = \text{check}(y, k_B^+) \text{ then let } y_2 = \text{dec}(y_1, k_C^-) \text{ then } \dots$

# Our goals and our approach

Automation:

- ▶ Algorithm compiles and inserts zero-knowledge proofs in protocol
- ▶ Use a type system, that checks well-typing automatically given an authorization policy

$A = 1 : \text{out}(ch, \text{sign}(\text{enc}(m, k_B^+), k_A^-))$

$B = \dots$

$C = 2 : \text{in}(ch, y).$

    let  $y_1 = \text{check}(y, k_B^+)$  then

    let  $y_2 = \text{dec}(y_1, k_C^-)$  then ...

$P = \dots$

# Our goals and our approach

Automation:

- ▶ Algorithm compiles and inserts zero-knowledge proofs in protocol
- ▶ Use a type system, that checks well-typing automatically given an authorization policy

$A = (1 : \text{out}(ch, \text{sign}(\text{enc}(m, k_B^+), k_A^-))) \text{ |assume } (\text{good}(m))$

$B = \dots$

$C = 2 : \text{in}(ch, y)$ .

    let  $y_1 = \text{check}(y, k_B^+)$  then

    let  $y_2 = \text{dec}(y_1, k_C^-)$  then **assert** ( $\text{good}(y_2)$ ) ...

$P = \dots$

# Our goals and our approach

Automation:

- ▶ Algorithm compiles and inserts zero-knowledge proofs in protocol
- ▶ Use a type system, that checks well-typing automatically given an authorization policy
- ▶ Theorem: For any authorization policy, the compilation preserves well-typing even under (*partial*) system compromise

$$\forall P \forall F : \Gamma \vdash P_F \quad \Rightarrow \quad \Gamma_{ZK} \vdash P_{ZK,F} \wedge \Gamma_{ZK} \vdash P_{ZK,F}^{cor}$$

with

$$P \xrightarrow{\text{compilation}} P_{ZK}$$

Thank you!