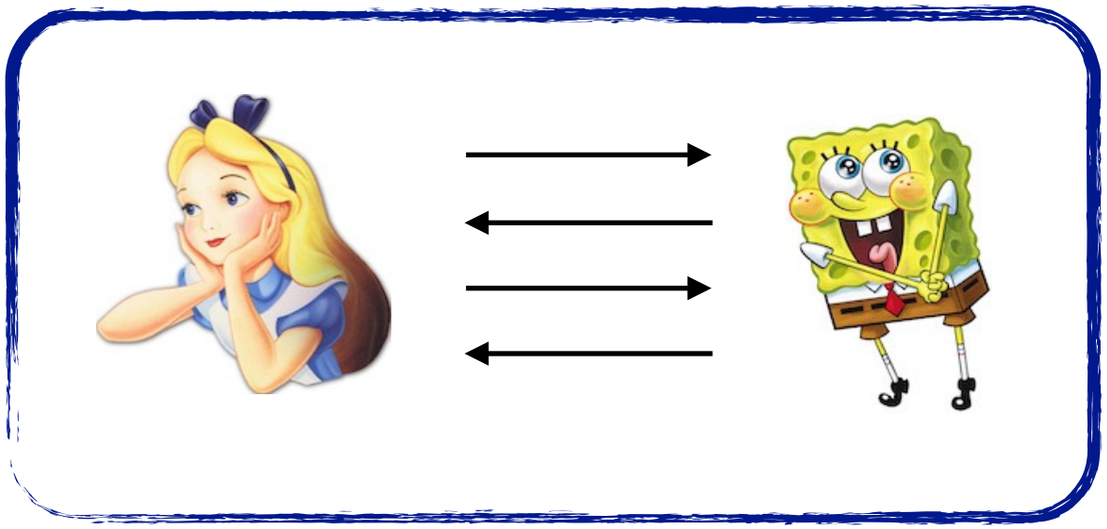
Tests for establishing security properties

Vincent Cheval, Stéphanie Delaune, Mark Ryan

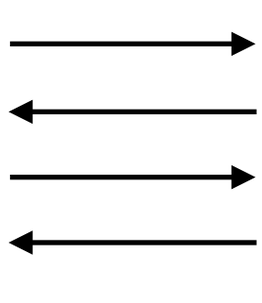
6 September, TGC, Roma

Context



The protocol: Honest participants

Context

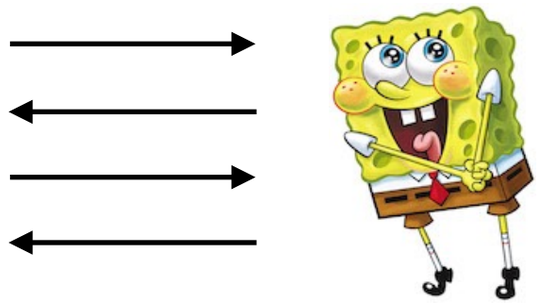


The protocol: Honest participants

Controls the network

Intruder

Context



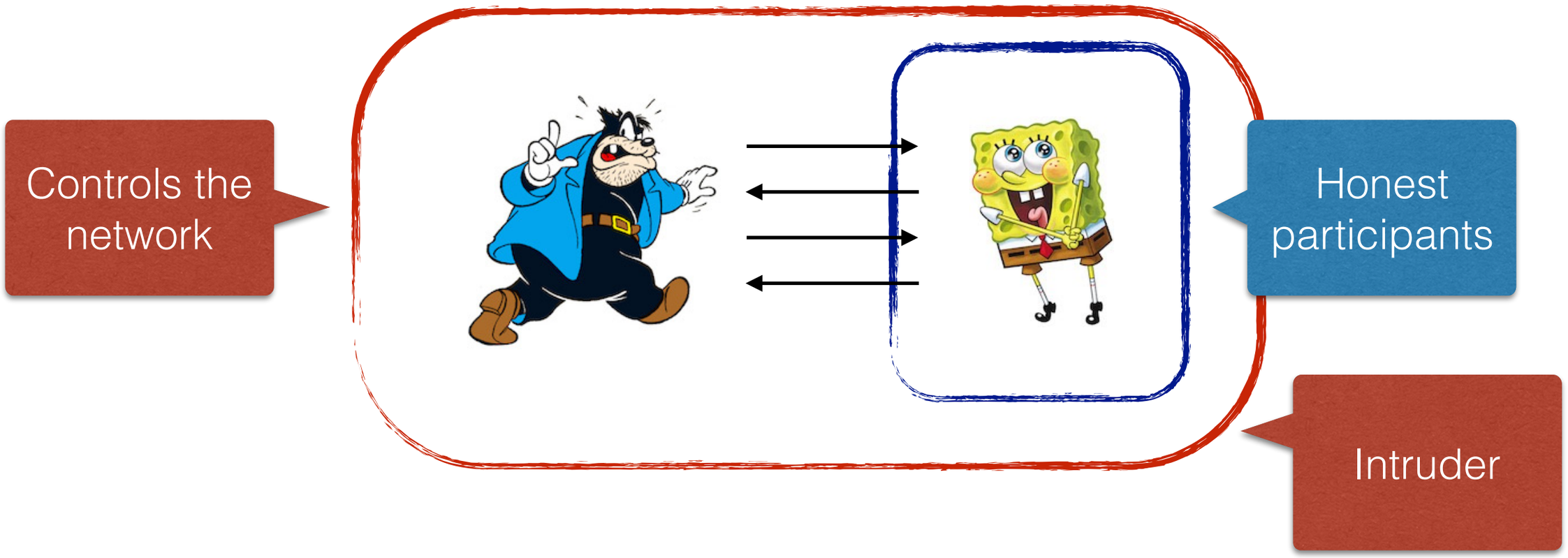
The protocol: Honest participants

Secrecy, Anonymity, Unlinkability, Authenticity, etc

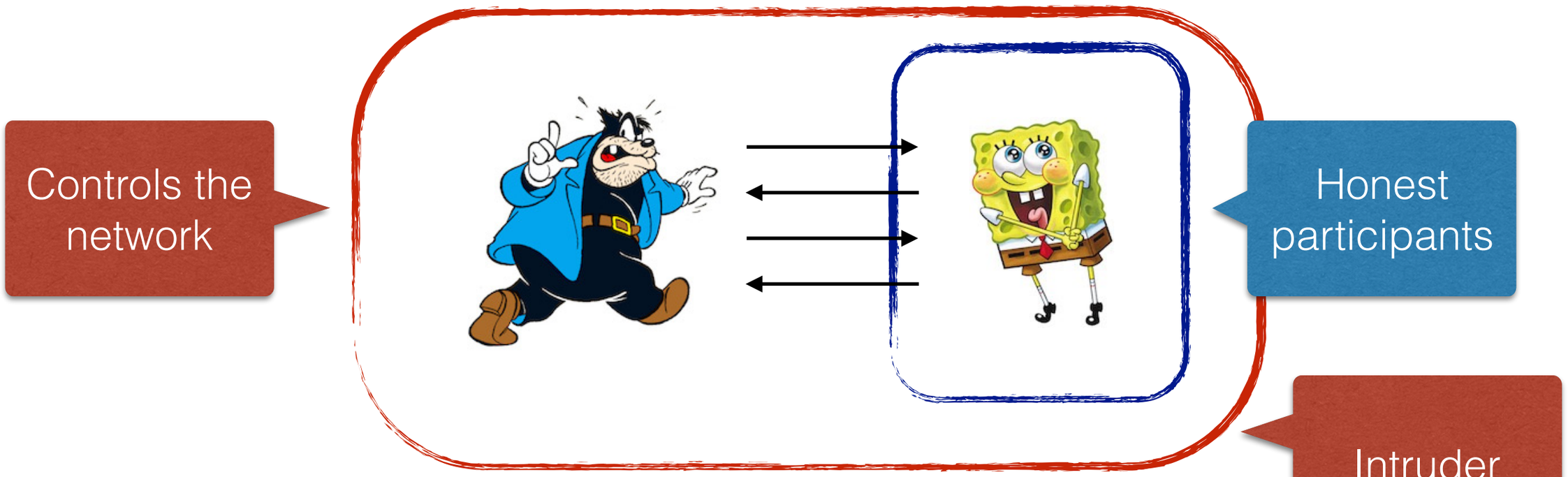
Controls the network

Intruder

Context

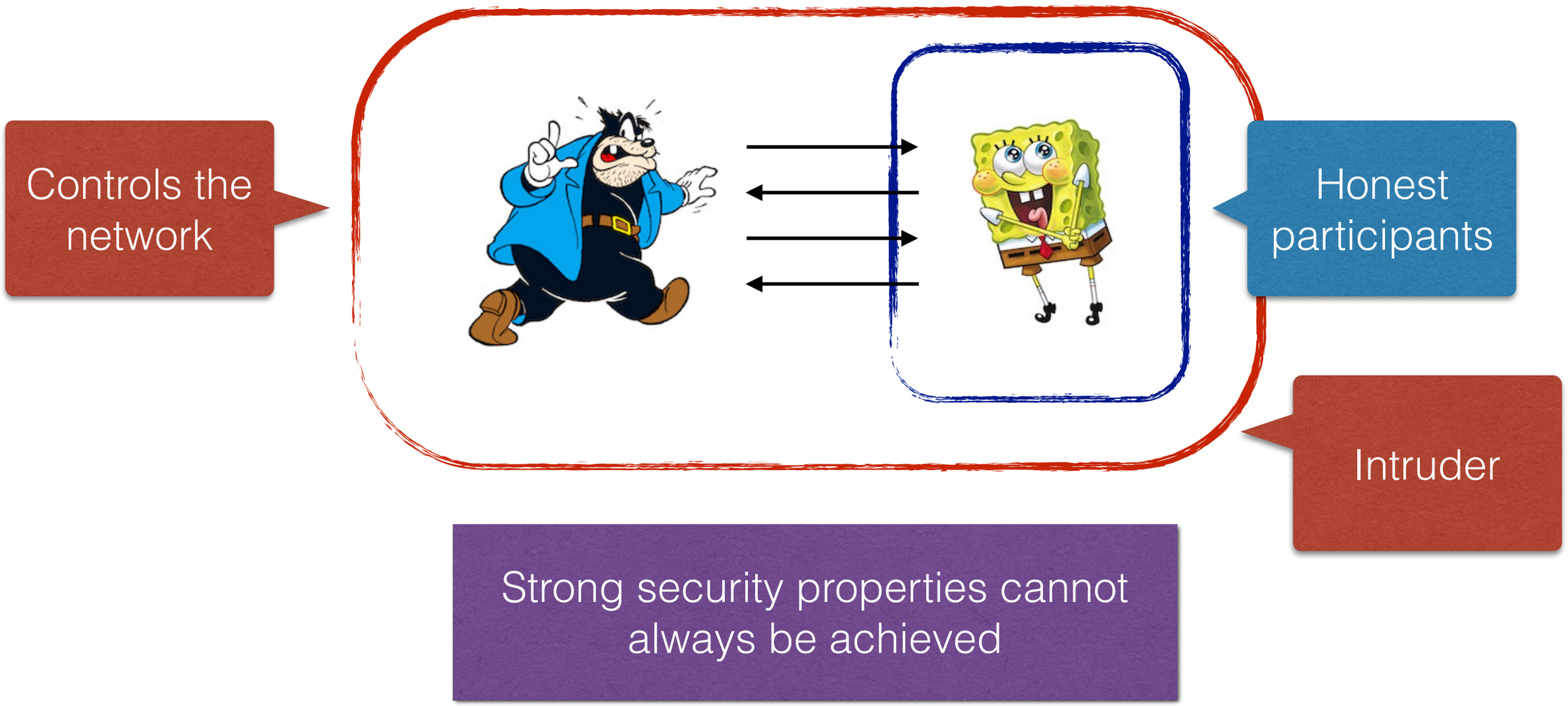


Context



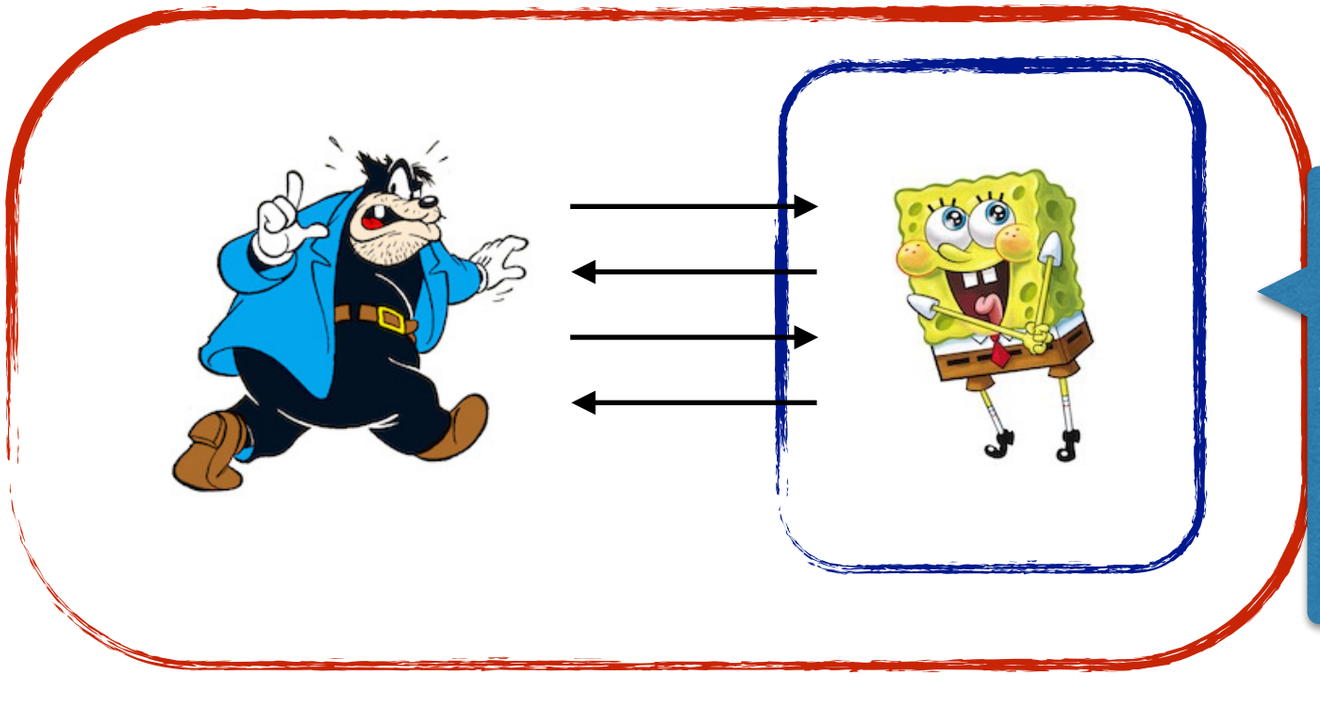
Strong security properties cannot always be achieved

Context



Common with inexistant or untrusted administrators: voting protocols, PKI, Bitcoin, Email managers, etc

Context

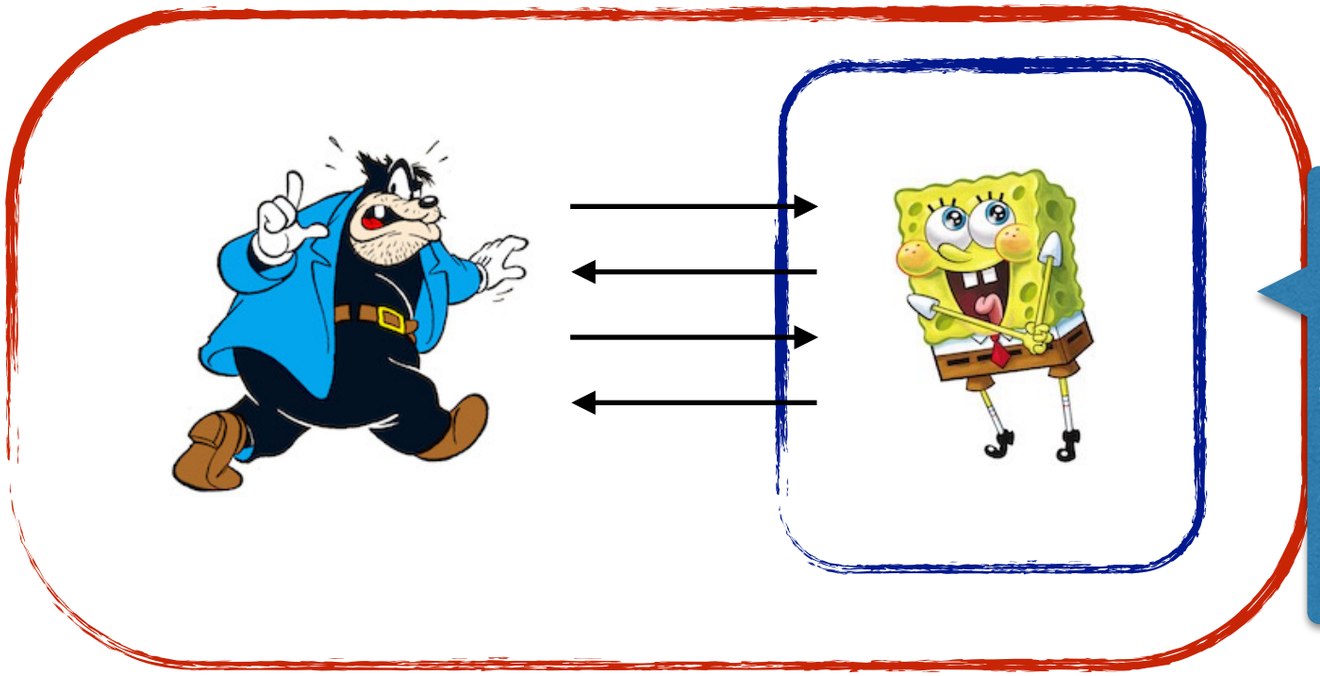


Honest participants run tests to guarantee the correctness of the outcome.

Voting protocol *Helios*: Voters verify

- Individual verifiability
- Universal verifiability
- Eligibility verifiability

Context



Honest participants run tests to guarantee the correctness of the outcome.

Voting protocol *Helios*: Voters verify

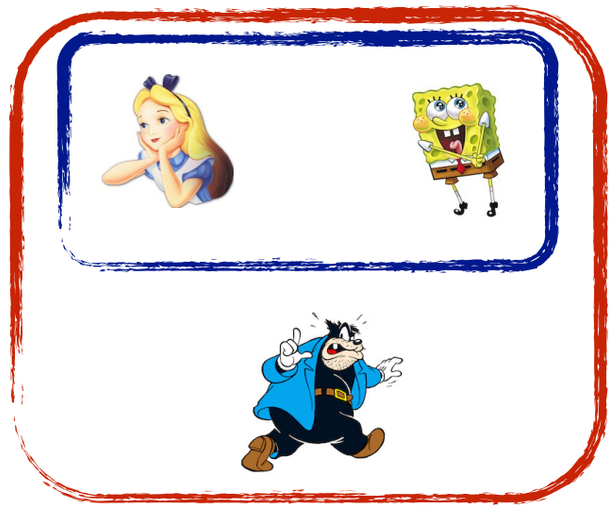
- Individual verifiability
- Universal verifiability
- Eligibility verifiability

Intruder can launch attacks → The attacks are detected by the tests performed by the users

Context

Usual security properties

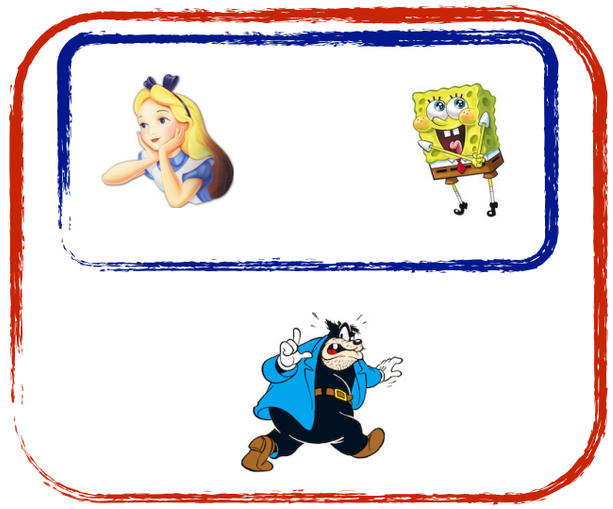
The attacks are impossible



Context

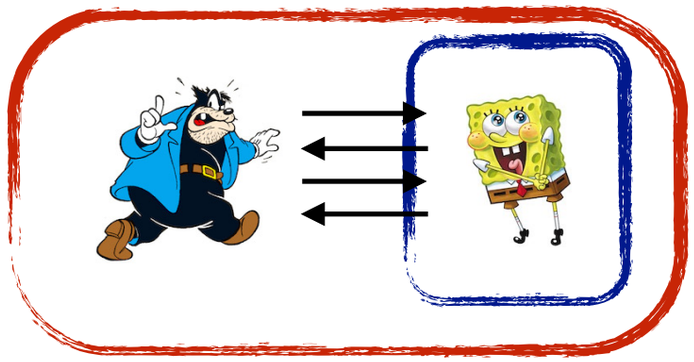
Usual security properties

The attacks are impossible



Intruder can launch attacks

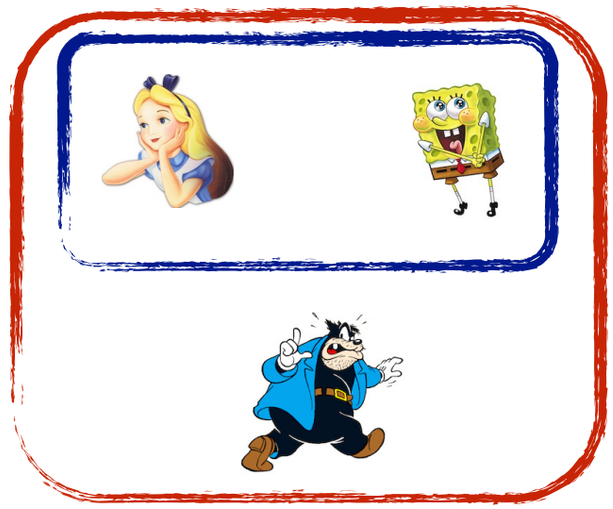
The attacks are detected by the tests performed by the users



Context

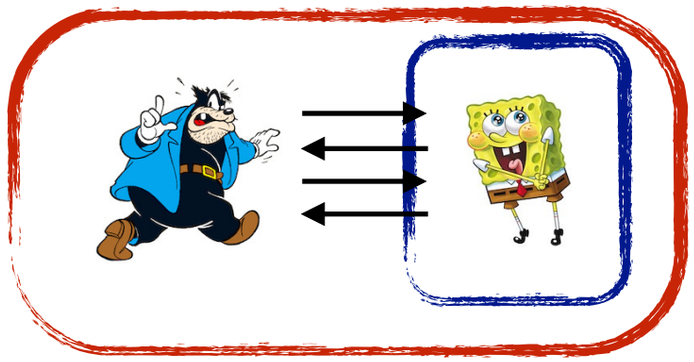
Usual security properties

The attacks are impossible



Intruder can launch attacks

The attacks are detected by the tests performed by the users



Weaker security properties:
Testable properties

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

sk_{CA}, **pk**(sk_{CA})

sk, **pk**(*sk*)

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

sk_{CA}, **pk**(sk_{CA})

I want to register my public key

sk, **pk**(*sk*)

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

sk_{CA}, **pk**(sk_{CA})

pk(*sk*)

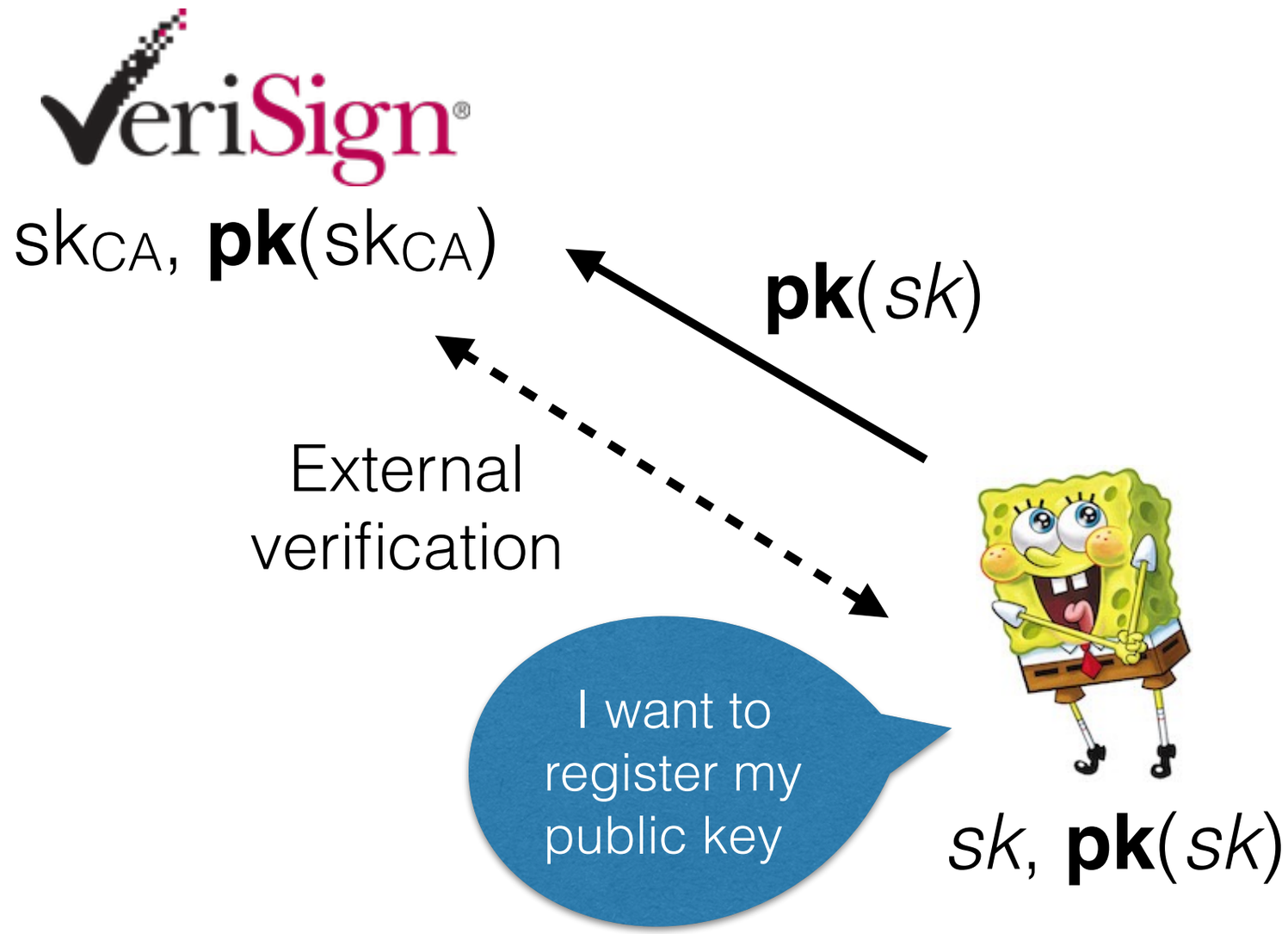
I want to register my public key

sk, **pk**(*sk*)

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…



Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

sk_{CA}, **pk**(sk_{CA})

pk(*sk*)

External verification

M, **pk**(sk_{CA})

I want to talk to Bob

sk, **pk**(*sk*)

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

$sign_{sk_{CA}}$(**pk**(sk),Bob)

sk_{CA}, **pk**(sk_{CA})

pk(*sk*)

Bob

External verification

M, **pk**(sk_{CA})

sk, **pk**(*sk*)

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

$sign_{sk_{CA}}$(**pk**(sk),Bob)

sk_{CA}, **pk**(sk_{CA})

pk(*sk*)

Bob

External verification

I verify the signature

M, **pk**(sk_{CA})

sk, **pk**(*sk*)

Certificate transparency

Public key certificate: digital identity (standard X.509)

Certificate authority: VeriSign, Comodo, Go Daddy…

sk_{CA}, **pk**(sk_{CA})

$sign_{sk_{CA}}$(**pk**(sk),Bob)

pk(*sk*)

Bob

External verification

enc(M,**pk**(*sk*))

M, **pk**(sk_{CA})

sk, **pk**(*sk*)

Certificate transparency

$sign_{sk_{CA}}(\mathbf{pk}(sk), Bob)$

sk_{CA}, $\mathbf{pk}(sk_{CA})$

$\mathbf{pk}(sk)$

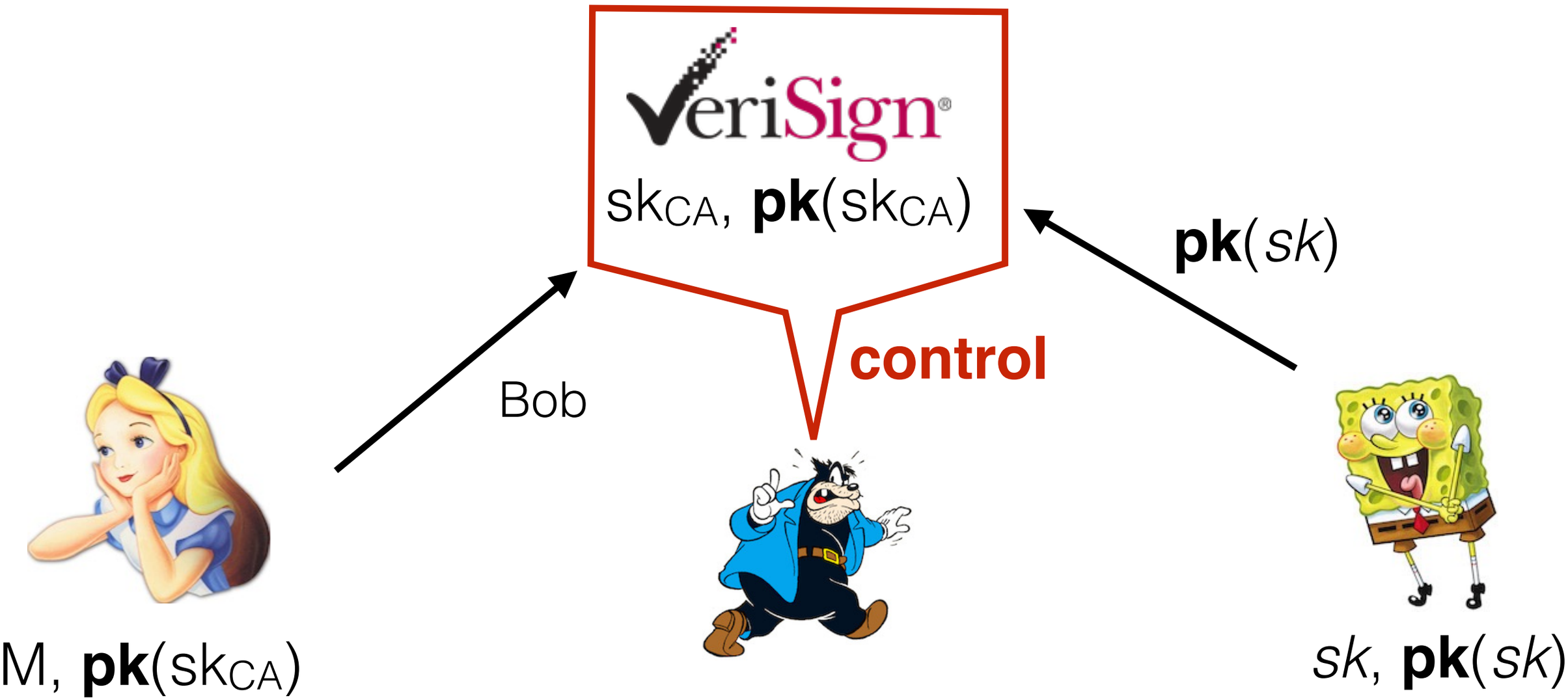
Bob

$\mathbf{enc}(M, \mathbf{pk}(sk))$

M, $\mathbf{pk}(sk_{CA})$

sk, $\mathbf{pk}(sk)$

Certificate transparency



Certificate transparency

VeriSign®

sk_{CA}, **pk**(sk_{CA})

control

Bob

M, **pk**(sk_{CA})

sk, **pk**(*sk*)

Certificate transparency

control

Bob

M, **pk**(skCA)

sk', **pk**(*sk'*)

sk, **pk**(*sk*)

Certificate transparency

VeriSign®

sk_{CA}, **pk**(sk_{CA})

control

M, **pk**(sk_{CA})

sk', **pk**(*sk'*)

sk, **pk**(*sk*)

Certificate transparency

signskCA(**pk**(sk'),Bob)

M, **pk**(skCA)

control

sk', **pk**(*sk'*)

sk, **pk**(*sk*)

Certificate transparency

signskCA(**pk**(sk'),Bob)

M, **pk**(skCA)

enc(M,**pk**(*sk'*))

sk', **pk**(*sk'*)

control

sk, **pk**(*sk*)

Certificate transparency

Classical attacks on PKI

Real attacks reported: Comodo, DigiNotar, ANSSI

VeriSign®

sk_{CA}, **pk**(sk_{CA})

$sign_{sk_{CA}}$(**pk**(sk'),Bob)

control

M, **pk**(sk_{CA})

enc(M,**pk**(*sk'*))

sk', **pk**(*sk'*)

sk, **pk**(*sk*)

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}

sk_{log}, **pk**(sk_{log})

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}

Bob, h_{log}

sk_{log}, **pk**(sk_{log})

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}

Bob, h_{log}

$cert_{Bob,\ hlog}$'

sk_{log}, **pk**(sk_{log})

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}

Bob, h_{log}

$cert_{Bob,\ hlog}$'

proof$_{\mathbf{p}}$($cert_{Bob}$,h_{log}')

sk_{log}, **pk**(sk_{log})

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}

Bob, h_{log}

$cert_{Bob,\ hlog}$'

proof$_p$($cert_{Bob}$,h_{log}')

proof$_e$(h_{log},h_{log}')

sk_{log}, **pk**(sk_{log})

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}

Bob, h_{log}

$cert_{Bob,\ hlog}$'

proof$_p$($cert_{Bob}$,h_{log}')

proof$_e$(h_{log},h_{log}')

sk_{log}, **pk**(sk_{log})

I verify
proof$_p$($cert_{Bob}$,h_{log}')
and **proof$_e$**(h_{log},h_{log}')

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}'

I verify
proof$_p$($cert_{Bob}$,h_{log}')
and **proof$_e$**(h_{log},h_{log}')

Bob, h_{log}

$cert_{Bob,\ hlog}$'

proof$_p$($cert_{Bob}$,h_{log}')

proof$_e$(h_{log},h_{log}')

sk_{log}, **pk**(sk_{log})

Certificate transparency

Data structure

- Digest of the log: a hash value
- Action: Addition
- Proofs of presence and extension, etc

pk(sk_{log}),h_{log}'

I verify
proof$_p$($cert_{Bob}$,h_{log}')
and **proof$_e$**(h_{log},h_{log}')

Bob, h_{log}

$cert_{Bob,\ hlog}$'

proof$_p$($cert_{Bob}$,h_{log}')

proof$_e$(h_{log},h_{log}')

sk_{log}, **pk**(sk_{log})

Testable authenticity
Testable secrecy

Malicious but Cautious

What kind of intruder is our system facing ?

Malicious but Cautious

What kind of intruder is our system facing ?

Active attacker in Dolev-Yao

Malicous:
Launch attacks by any means necessary

Malicious but Cautious

What kind of intruder is our system facing ?

Active attacker in Dolev-Yao

Malicous:
Launch attacks by any means necessary

Passive attacker in Dolev-Yao

Honest but Curious:
Only listen or participate honestly to the protocol but try to retrieve secrets

Malicious but Cautious

What kind of intruder is our system facing ?

Active attacker in Dolev-Yao

Malicous:
Launch attacks by any means necessary

Malicous but Cautious:
Launch attacks but does not want to be detected

Passive attacker in Dolev-Yao

Honest but Curious:
Only listen or participate honestly to the protocol but try to retrieve secrets

Malicious but Cautious

What kind of intruder is our system facing ?

Active attacker in Dolev-Yao

Malicous:
Launch attacks by any means necessary

Malicous but Cautious:
Launch attacks but does not want to be detected

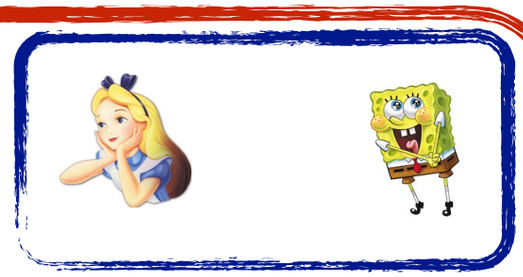
Detection of intruders is done by verifying tests

Passive attacker in Dolev-Yao

Honest but Curious:
Only listen or participate honestly to the protocol but try to retrieve secrets

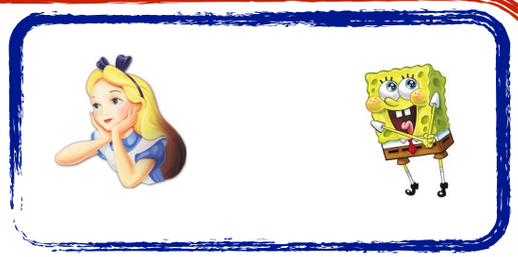
Broken protocols

Secrecy, Anonymity,
Unlinkability, Authenticity, etc



Broken protocols

Secrecy, Anonymity, Unlinkability, Authenticity, etc



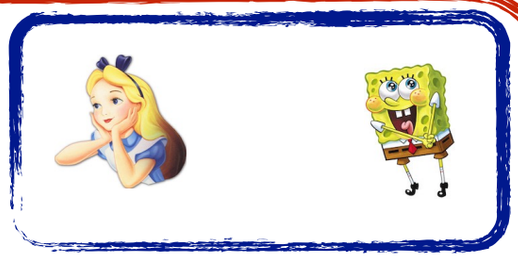
Automatic tools

Security properties are satisfied

Discovery of an attack

Broken protocols

Secrecy, Anonymity, Unlinkability, Authenticity, etc



Automatic tools

Security properties are satisfied

Discovery of an attack

Is this attack detectable ?

Is it the only one ?

Can we prove some degree of security ?

French e-passport protocol

k_e, k_m

Passport

k_e, k_m

Reader

French e-passport protocol

k_e, k_m

Passport

n_T

k_e, k_m

Reader

French e-passport protocol

k_e, k_m

Passport

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

k_e, k_m

Reader

French e-passport protocol

k_e, k_m

Passport

k_e, k_m

Reader

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

If wrong mac key

ErrorMac

French e-passport protocol

Passport

k_e, k_m

k_e, k_m

Reader

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

If wrong mac key

ErrorMac

If wrong encryption key

ErrorNonce

French e-passport protocol

k_e, k_m

Passport

k_e, k_m

Reader

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

If wrong mac key

ErrorMac

If wrong encryption key

ErrorNonce

Else

$\{(n_T, n_R, k_T)\}_{k_e}$

French e-passport protocol

Passport

k_e, k_m

k_e, k_m

Reader

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

ErrorMac

If wrong mac key

ErrorNonce

If wrong encryption key

$\{(n_T, n_R, k_T)\}_{k_e}$

Else

Unlinkability of the french passport is not satisfied

[ACRR'10]

French e-passport protocol

Passport

k_e, k_m

k_e, k_m

Reader

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

ErrorMac

If wrong mac key

ErrorNonce

If wrong encryption key

$\{(n_T, n_R, k_T)\}_{k_e}$

Else

Testable unlinkability [ACRR'10]

Extended Applied Pi Calculus

Applied Pi Calculus

$$
\begin{aligned}
P, Q := \quad & 0 \\
\mid \quad & (P \mid Q) \\
\mid \quad & \mathsf{new}\ n.P \\
\mid \quad & !P \\
\mid \quad & \mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \\
\mid \quad & \mathsf{in}(u, x).P \\
\mid \quad & \mathsf{out}(u, v).P
\end{aligned}
$$

Extended Applied Pi Calculus

Applied Pi Calculus

$$
\begin{aligned}
P, Q := \quad & 0 \\
\mid \quad & (P \mid Q) \\
\mid \quad & \mathsf{new}\ n.P \\
\mid \quad & !P \\
\mid \quad & \mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \\
\mid \quad & \mathsf{in}(u, x).P \\
\mid \quad & \mathsf{out}(u, v).P
\end{aligned}
$$

- Need to specify properties on specific sessions / participants
- Need to express the tests done by the participants

Extended Applied Pi Calculus

Patterns

Applied Pi Calculus

$$!P$$

$$!P \rightarrow !P \mid P$$

Extension

Extended Applied Pi Calculus

Patterns

Applied Pi Calculus

$$!P$$

$$!P \ \rightarrow !P \mid P$$

Extension

$$!^{\mathsf{i} \geq j} P$$

$$!^{\mathsf{i} \geq j} P \ \rightarrow !^{\mathsf{i} \geq j+1} P \mid P\{^{j}/_{\mathsf{i}}\}$$

Extended Applied Pi Calculus

Patterns

Applied Pi Calculus	Extension
$!P$	$!^{\mathsf{i} \geq j} P$
$!P \ \rightarrow !P \mid P$	$!^{\mathsf{i} \geq j} P \ \rightarrow !^{\mathsf{i} \geq j+1} P \mid P\{^{j}/_{\mathsf{i}}\}$
x	$x[\mathsf{i}_1, \ldots, \mathsf{i}_k]$
n	$n[\mathsf{i}_1, \ldots, \mathsf{i}_k]$

Extended Applied Pi Calculus

Patterns

Applied Pi Calculus	Extension
$!P$	$!^{\mathsf{i} \geq j} P$
$!P \ \rightarrow !P \mid P$	$!^{\mathsf{i} \geq j} P \ \rightarrow !^{\mathsf{i} \geq j+1} P \mid P\{^{j}/_{\mathsf{i}}\}$
x	$x[\mathsf{i}_1, \ldots, \mathsf{i}_k]$
n	$n[\mathsf{i}_1, \ldots, \mathsf{i}_k]$

Applied Pi Calculus $!\mathsf{new}\ n_1.!\mathsf{new}\ n_2.\mathsf{out}(c, (n_1, n_2))$

Extended Applied Pi Calculus

Patterns

Applied Pi Calculus	Extension
$!P$	$!^{\mathsf{i} \geq j} P$
$!P \ \rightarrow !P \mid P$	$!^{\mathsf{i} \geq j} P \ \rightarrow !^{\mathsf{i} \geq j+1} P \mid P\{^{j}/_{\mathsf{i}}\}$
x	$x[\mathsf{i}_1, \ldots, \mathsf{i}_k]$
n	$n[\mathsf{i}_1, \ldots, \mathsf{i}_k]$

Applied Pi Calculus $!\mathsf{new}\ n_1.!\mathsf{new}\ n_2.\mathsf{out}(c, (n_1, n_2))$

Extension $!^{\mathsf{i}_1}\mathsf{new}\ n_1[\mathsf{i}_1].!^{\mathsf{i}_2}\mathsf{new}\ n_2[\mathsf{i}_1, \mathsf{i}_2].\mathsf{out}(c, (n_1[\mathsf{i}_1], n_2[\mathsf{i}_1, \mathsf{i}_2]))$

Extended Applied Pi Calculus

Extended Applied Pi Calculus

$$
\begin{array}{rcl}
P, Q := & & 0 \\
& | & (P \mid Q) \\
& | & \mathsf{new}\ \overline{n}.P \\
& | & !^{i \geq j} P \\
& | & \mathsf{if}\ \overline{u} = \overline{v}\ \mathsf{then}\ P\ \mathsf{else}\ Q \\
& | & \mathsf{in}(\overline{u}, \overline{x}).P \\
& | & \mathsf{out}(\overline{u}, \overline{v}).P
\end{array}
$$

Extended Applied Pi Calculus

Extended Applied Pi Calculus

$$
\begin{aligned}
P, Q := \quad & 0 \\
\mid \quad & (P \mid Q) \\
\mid \quad & \mathsf{new}\ \overline{n}.P \\
\mid \quad & !^{i \geq j} P \\
\mid \quad & \mathsf{if}\ \overline{u} = \overline{v}\ \mathsf{then}\ P\ \mathsf{else}\ Q \\
\mid \quad & \mathsf{in}(\overline{u}, \overline{x}).P \\
\mid \quad & \mathsf{out}(\overline{u}, \overline{v}).P
\end{aligned}
$$

- Need to specify properties on specific sessions / participants

Extended Applied Pi Calculus

Extended Applied Pi Calculus

$$
\begin{aligned}
P, Q := \quad & 0 \\
\mid \quad & (P \mid Q) \\
\mid \quad & \mathsf{new}\ \overline{n}.P \\
\mid \quad & !^{i \geq j} P \\
\mid \quad & \mathsf{if}\ \overline{u} = \overline{v}\ \mathsf{then}\ P\ \mathsf{else}\ Q \\
\mid \quad & \mathsf{in}(\overline{u}, \overline{x}).P \\
\mid \quad & \mathsf{out}(\overline{u}, \overline{v}).P
\end{aligned}
$$

- Need to specify properties on specific sessions / participants
- Need to express the tests done by the participants

Extended Applied Pi Calculus

Extended Applied Pi Calculus

$$
\begin{aligned}
P, Q := \quad & 0 \\
\mid \quad & (P \mid Q) \\
\mid \quad & \mathsf{new}\ \overline{n}.P \\
\mid \quad & !^{i \geq j} P \\
\mid \quad & \mathsf{if}\ \overline{u} = \overline{v}\ \mathsf{then}\ P\ \mathsf{else}\ Q \\
\mid \quad & \mathsf{in}(\overline{u}, \overline{x}).P \\
\mid \quad & \mathsf{out}(\overline{u}, \overline{v}).P \\
\mid \quad & \mathsf{rec}(\overline{z}, \overline{u}).P
\end{aligned}
$$

Extended Applied Pi Calculus

Extended Applied Pi Calculus

$$
\begin{array}{rcl}
P, Q := & & 0 \\
& | & (P \mid Q) \\
& | & \text{new } \overline{n}.P \\
& | & !^{i \geq j} P \\
& | & \text{if } \overline{u} = \overline{v} \text{ then } P \text{ else } Q \\
& | & \text{in}(\overline{u}, \overline{x}).P \\
& | & \text{out}(\overline{u}, \overline{v}).P \\
& | & \text{rec}(\overline{z}, \overline{u}).P
\end{array}
$$

Represents where the message is stored

The message recorded in the personal log

Extended Applied Pi Calculus

Extended Processes

Applied Pi Calculus	Extension
$(\mathcal{E}, P, \Phi_{\mathcal{A}})$	$(\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$

Extended Applied Pi Calculus

Extended Processes

Applied Pi Calculus

$$(\mathcal{E}, P, \Phi_{\mathcal{A}})$$

Extension

$$(\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$$

Formulas

$$\Phi_{log} \vDash M =^{?} N \longleftrightarrow v(M,N) \subseteq dom(\Phi_{log}) \text{ and } M\Phi_{log} =_E N\Phi_{log}$$

$$\Phi_{log} \vDash M \neq^{?} N \longleftrightarrow v(M,N) \subseteq dom(\Phi_{log}) \text{ and } M\Phi_{log} \neq_E N\Phi_{log}$$

Extended Applied Pi Calculus

Extended Processes

Applied Pi Calculus

$$(\mathcal{E}, P, \Phi_{\mathcal{A}})$$

Extension

$$(\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$$

Formulas

$$\Phi_{log} \vDash M =^{?} N \longleftrightarrow v(M,N) \subseteq dom(\Phi_{log}) \text{ and } M\Phi_{log} =_E N\Phi_{log}$$
$$\Phi_{log} \vDash M \neq^{?} N \longleftrightarrow v(M,N) \subseteq dom(\Phi_{log}) \text{ and } M\Phi_{log} \neq_E N\Phi_{log}$$

$$\phi = \forall i \in \mathbb{N}.\forall j \in \mathbb{N}.\left(z_{start}[i,j] =^{?} \text{true} \Rightarrow z_{end}[i,j] \neq^{?} \text{error}\right)$$

Testable unlinkability

Passport

k_e, k_m

k_e, k_m

Reader

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

ErrorMac

ErrorNonce

$\{(n_T, n_R, k_T)\}_{k_e}$

If wrong mac key

If wrong encryption key

Else

Testable unlinkability

Passport

k_e, k_m

n_T

$\{(n_R, n_T, k_R)\}_{k_e}$

$\mathsf{mac}(\{(n_R, n_T, k_R)\}_{k_e}, k_m)$

k_e, k_m

Reader

If wrong mac key — ErrorMac

If wrong encryption key — ErrorNonce

Else — $\{(n_T, n_R, k_T)\}_{k_e}$

$$!^{\mathsf{i}}\mathsf{new}\ k_e[\mathsf{i}].\mathsf{new}\ k_m[\mathsf{i}].!^{\mathsf{j}}(P_{\mathsf{Pass}}[\mathsf{i},\mathsf{j}] \mid P_{\mathsf{Reader}}[\mathsf{i},\mathsf{j}])$$

Testable unlinkability

Unlinkability

For all $A \overset{l_1[i_1]}{\rightarrow} A_1 \overset{l_2[i_2]}{\rightarrow} \dots \overset{l_n[i_n]}{\rightarrow} (\mathcal{E}, P, \Phi_{\mathcal{A}})$

with $i_p = i_q$

Testable unlinkability

Unlinkability

For all $A \overset{l_1[i_1]}{\to} A_1 \overset{l_2[i_2]}{\to} \dots \overset{l_n[i_n]}{\to} (\mathcal{E}, P, \Phi_{\mathcal{A}})$

with $i_p = i_q$

there exists $A \overset{l'_1[i'_1]}{\to} A'_1 \overset{l'_2[i'_2]}{\to} \dots \overset{l'_n[i'_n]}{\to} (\mathcal{E}', P', \Phi'_{\mathcal{A}})$

such that $\Phi_{\mathcal{A}} \sim \Phi'_{\mathcal{A}}$ and $i'_p \neq i'_q$

Testable unlinkability

Unlinkability

For all $A \overset{l_1[i_1]}{\rightarrow} A_1 \overset{l_2[i_2]}{\rightarrow} \dots \overset{l_n[i_n]}{\rightarrow} (\mathcal{E}, P, \Phi_{\mathcal{A}})$

with $i_p = i_q$

there exists $A \overset{l'_1[i'_1]}{\rightarrow} A'_1 \overset{l'_2[i'_2]}{\rightarrow} \dots \overset{l'_n[i'_n]}{\rightarrow} (\mathcal{E}', P', \Phi'_{\mathcal{A}})$

such that $\Phi_{\mathcal{A}} \sim \Phi'_{\mathcal{A}}$ and $i'_p \neq i'_q$

French e-passport does not satisfies unlinkability

Testable unlinkability

ϕ - Unlinkability

For all $A \overset{l_1[i_1,j_1]}{\rightarrow} A_1 \overset{l_2[i_2,j_2]}{\rightarrow} \ldots \overset{l_n[i_n,j_n]}{\rightarrow} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$

with $i_p = i_q$

Testable unlinkability

ϕ - Unlinkability

For all $A \overset{l_1[i_1,j_1]}{\rightarrow} A_1 \overset{l_2[i_2,j_2]}{\rightarrow} \ldots \overset{l_n[i_n,j_n]}{\rightarrow} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$

with $i_p = i_q$ and $\Phi_{log} \vDash \phi(i_p, j_p, j_q)$

Testable unlinkability

ϕ - Unlinkability

For all $A \xrightarrow{l_1[i_1,j_1]} A_1 \xrightarrow{l_2[i_2,j_2]} \dots \xrightarrow{l_n[i_n,j_n]} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$

with $i_p = i_q$ and $\Phi_{log} \vDash \phi(i_p, j_p, j_q)$

there exists $A \xrightarrow{l'_1[i'_1,j'_1]} A'_1 \xrightarrow{l'_2[i'_2,j'_2]} \dots \xrightarrow{l'_n[i'_n,j'_n]} (\mathcal{E}', P', \Phi'_{\mathcal{A}}, \Phi'_{log})$

such that $\Phi_{\mathcal{A}} \sim \Phi'_{\mathcal{A}}$, $i'_p \neq i'_q$ and $\Phi'_{log} \vDash \phi(i'_p, j'_p, j'_q)$

Testable unlinkability

ϕ - Unlinkability

For all $A \xrightarrow{l_1[i_1,j_1]} A_1 \xrightarrow{l_2[i_2,j_2]} \dots \xrightarrow{l_n[i_n,j_n]} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$

with $i_p = i_q$ and $\Phi_{log} \vDash \phi(i_p, j_p, j_q)$

there exists $A \xrightarrow{l'_1[i'_1,j'_1]} A'_1 \xrightarrow{l'_2[i'_2,j'_2]} \dots \xrightarrow{l'_n[i'_n,j'_n]} (\mathcal{E}', P', \Phi'_{\mathcal{A}}, \Phi'_{log})$

such that $\Phi_{\mathcal{A}} \sim \Phi'_{\mathcal{A}}$, $i'_p \neq i'_q$ and $\Phi'_{log} \vDash \phi(i'_p, j'_p, j'_q)$

What should be the formula ?

Testable unlinkability

$$A \overset{l_1[i_1,j_1]}{\rightarrow} A_1 \overset{l_2[i_2,j_2]}{\rightarrow} \ldots \overset{l_n[i_n,j_n]}{\rightarrow} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$$

$$\Phi_{log} \vDash \phi(i_p, j_p, j_q)$$

Testable unlinkability

$$A \xrightarrow{l_1[i_1,j_1]} A_1 \xrightarrow{l_2[i_2,j_2]} \dots \xrightarrow{l_n[i_n,j_n]} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$$

$$\Phi_{log} \vDash \phi(i_p, j_p, j_q)$$

$$\phi_1 = \forall i \in \mathbb{N}.\forall j \in \mathbb{N}.\big(z_{start}[i,j] =^{?} \text{true} \Rightarrow z_{end}[i,j] \neq^{?} \text{error}\big)$$

Traces where all tests of all users succeed

Testable unlinkability

$$A \xrightarrow{l_1[i_1,j_1]} A_1 \xrightarrow{l_2[i_2,j_2]} \dots \xrightarrow{l_n[i_n,j_n]} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$$

$$\Phi_{log} \vDash \phi(i_p, j_p, j_q)$$

$$\phi_1 = \forall i \in \mathbb{N}. \forall j \in \mathbb{N}. \left(z_{start}[i,j] =^{?} \text{true} \Rightarrow z_{end}[i,j] \neq^{?} \text{error} \right)$$

Traces where all tests of all users succeed

$$\phi_2(i_0) = \forall j \in \mathbb{N}. \left(z_{start}[i_0,j] =^{?} \text{true} \Rightarrow z_{end}[i_0,j] \neq^{?} \text{error} \right)$$

Traces where all tests of one specific user succeed

Testable unlinkability

$$A \xrightarrow{l_1[i_1,j_1]} A_1 \xrightarrow{l_2[i_2,j_2]} \dots \xrightarrow{l_n[i_n,j_n]} (\mathcal{E}, P, \Phi_{\mathcal{A}}, \Phi_{log})$$

$$\Phi_{log} \vDash \phi(i_p, j_p, j_q)$$

$$\phi_1 = \forall i \in \mathbb{N}. \forall j \in \mathbb{N}. \big(z_{start}[i,j] =^? \text{true} \Rightarrow z_{end}[i,j] \neq^? \text{error}\big)$$

Traces where all tests of all users succeed

$$\phi_2(i_0) = \forall j \in \mathbb{N}. \big(z_{start}[i_0,j] =^? \text{true} \Rightarrow z_{end}[i_0,j] \neq^? \text{error}\big)$$

Traces where all tests of one specific user succeed

$$\phi_3(i_0, j_0, j_0') = z_{end}[i_0, j_0'] \neq^? \text{error} \wedge z_{end}[i_0, j_0'] \neq^? \text{error}$$

Traces where tests of two sessions of one specific user succeed

Conclusion

New framework based on Applied Pi Calculus

Detection of attacks

Security analysis on broken protocols

Malicious but Cautious intruder

Based on testable security properties

ϕ-secrecy

ϕ-anonymity

ϕ-authenticity

ϕ-unlinkability

Conclusion

New framework based on Applied Pi Calculus

Detection of attacks

Security analysis on broken protocols

Malicious but Cautious intruder

Based on testable security properties

ϕ-secrecy ϕ-anonymity

ϕ-authenticity ϕ-unlinkability

Automatic verification ?

APTE

AkiSs

ProVerif