# Security Ceremony Concertina

## Design and Verification of Security Protocols and Security Ceremonies

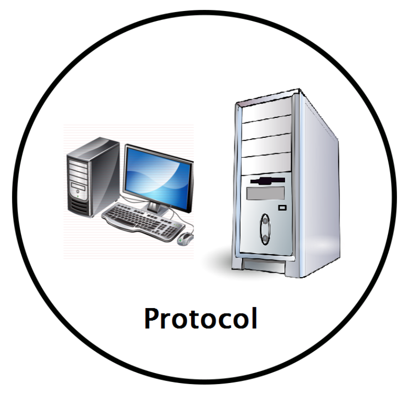
Programa de Pós-Graduacão em Ciências da Computacão
Dr. Jean Everson Martina

August-November 2016

Programa de Pós-Graduação em Ciência da Computação
UFSC

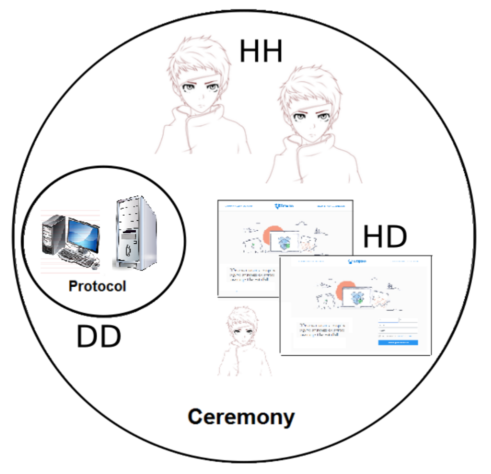
UNIVERSIDADE FEDERAL
DE SANTA CATARINA

# Motivation



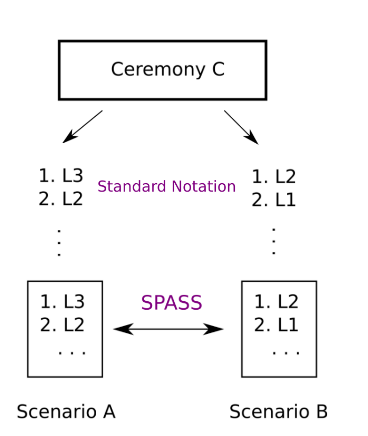

Protocols have several automated tools for formal analysis.

# Motivation




Lack of symbolic evaluation methods to verify claims embedded in security ceremonies.

# Why is formalisation important?




Need of standard procedures in order to compare scenarios.

# Goal

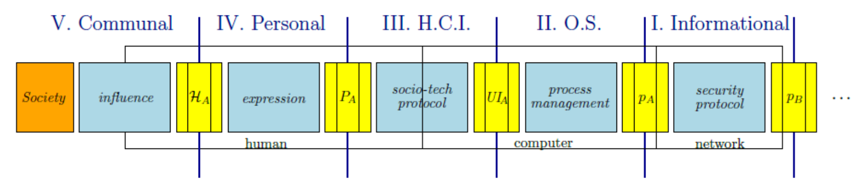
- Pave the way for symbolic evaluation of socio-technical security ceremonies through:
  - The establishment of a standard syntax for messages description.
  - An augmented threat model to encompass the subtleties of security ceremonies.

# Contributions

- Security ceremony description syntax;
- Precise threat model which encompass all subtleties of human peers;
- Proposal: Distributed Attacker (DA) model;
- Strategy for mechanisation and formalisation of ceremonies.

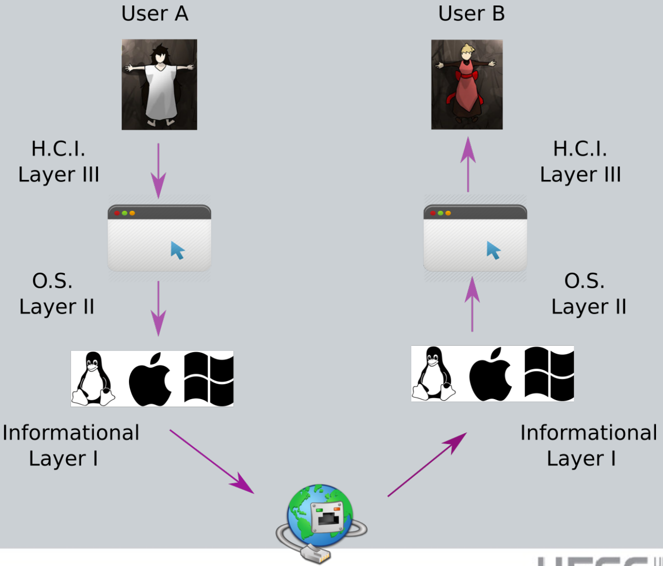
# Ceremony Concertina methodology

To represent channels DD, HD and HH as layers, we use the Security Ceremony Concertina methodology :




- As noted by them, this model is only fully understandable when put in the context of the threat model it is being used with.
- They believe that a ceremony can be layered and the analysis can be focused on specific sections of the description, trying to describe or

# Ceremony Concertina methodology



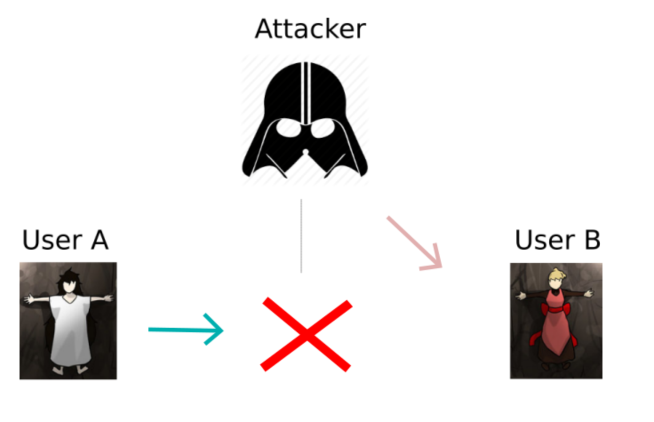
# Attacker types and capabilities

- The Dolev-Yao (DY) attacker is widely known and the most accepted for protocols.
  - Capabilities: Eavesdrop, Initiate, Atomic Break Down, Crypto, Block, Fabricate, Spoof, Re-Order, Modifying and Replaying.
- The Multi-Attacker (MA) attacker is a DY variant.
  - A MA may control more than only one channel .

# Threat Models

- To approach the threshold between a realistic and secure ceremony, Carlos et al. proposed a dynamic threat model.
    - Adjusts the Dolev-Yao full set of capabilities to make the attacker more realistic.

# Why a DY is not always realistic?




Human peers subject to laws of physics.

# Related work

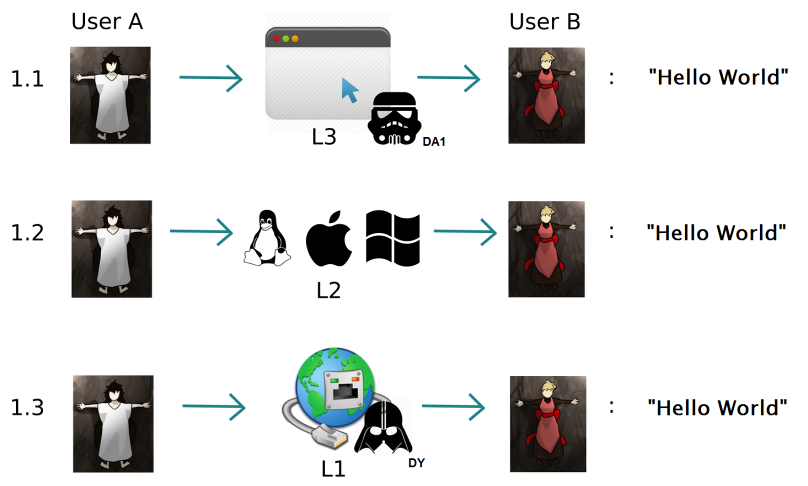
Some works already tried to address ceremony design and verification:

- Carlos et al further pursued these formalisation ideas using Isabelle (Higher-Order Logic, also known as HOL).
- Martina et al further expands Carlos et al by demonstrating how to conduct symbolic evaluation with the adaptive threat model (using FOL and a theorem prover).

# Distributed Attacker (DA) approach

| Threat Model | Share knowledge | Same abilities | Different chann |
|---|---|---|---|
| DY | No | Yes | No |
| MA | No | Yes | Yes |
| DA | Yes* | No* | Yes* |

# Ceremony description notation



# Dropbox case peers

- Entities: $U_C$ (user computer), $U_P$ (user phone) and $D_S$ (Dropbox server).
- Communication between $U_C$ and $D_S$:
  - Attacker $DA_1$ eavesdropping and blocking the user computer;
  - Key-logger (attacker $DA_2$) on user's computer;
  - DY attacker on Internet.

# Dropbox case peers

- Communication between $D_S$ and $U_P$:
  - Attackers $MA_1$ and $DA_3$ controlling the user's phone (e.g. through a virus);
  - $DA_1$ also eavesdropping on user's phone.

# Dropbox 2-step verification sign-in ceremony

| | | | | | |
|---|---|---|---|---|---|
| 1.1 | $U_C$ | $\xrightarrow[L3(E+B)_{DA_1}]{}$ | $D_S$ | : | Dropbox URL |
| 1.2 | $U_C$ | $\xrightarrow[L2(E)_{DA_2}]{}$ | $D_S$ | : | Dropbox URL |
| 1.3 | $U_C$ | $\xrightarrow[L1(DY)_{DY}]{}$ | $D_S$ | : | {Dropbox URL} |
| 2.1 | $D_S$ | $\xrightarrow[L1(DY)_{DY}]{}$ | $U_C$ | : | {Dropbox page} |
| 2.2 | $D_S$ | $\xrightarrow[L2(E)_{DA_2}]{}$ | $U_C$ | : | Dropbox page |
| 2.3 | $D_S$ | $\xrightarrow[L3(E+B)_{DA_1}]{}$ | $U_C$ | : | Dropbox page |
| 3.1 | $U_C$ | $\xrightarrow[L3(E+B)_{DA_1}]{}$ | $D_S$ | : | "*Sign-in*" |
| 3.2 | $U_C$ | $\xrightarrow[L2(E)_{DA_2}]{}$ | $D_S$ | : | "*Sign-in*" |
| 3.3 | $U_C$ | $\xrightarrow[L1(DY)_{DY}]{}$ | $D_S$ | : | "*{Sign-in}*" |
| 4.1 | $D_S$ | $\xrightarrow[L1(DY)_{DY}]{}$ | $U_C$ | : | {Sign-in page} |
| 4.2 | $D_S$ | $\xrightarrow[L2(E)_{DA_2}]{}$ | $U_C$ | : | Sign-in page |
| 4.3 | $D_S$ | $\xrightarrow[L3(E+B)_{DA_1}]{}$ | $U_C$ | : | Sign-in page |

# Dropbox 2-step verification sign-in ceremony

| | | | | | |
|---|---|---|---|---|---|
| 5.1 | $U_C$ | $\xrightarrow[{}^{L3}(E+B)_{DA_1}]{}$ | $D_S$ | : | ***(email,password)*** |
| 5.2 | $U_C$ | $\xrightarrow[{}^{L2}(E)_{DA_2}]{}$ | $D_S$ | : | ***(email,password)*** |
| 5.3 | $U_C$ | $\xrightarrow[{}^{L1}(DY)_{DY}]{}$ | $D_S$ | : | *{(email,password)}* |
| 6.1 | $D_S$ | $\xrightarrow[{}^{L1}(DY)_{DY}]{}$ | $U_C$ | : | {2-step verification} |
| 6.2 | $D_S$ | $\xrightarrow[{}^{L2}(E)_{DA_2}]{}$ | $U_C$ | : | 2-step verification |
| 6.3 | $D_S$ | $\xrightarrow[{}^{L3}(E+B)_{DA_1}]{}$ | $U_C$ | : | 2-step verification |
| 7.1 | $D_S$ | $\xrightarrow[{}^{L2}(DY)_{MA_1},(DY)_{DA_3}]{}$ | $U_P$ | : | **Auth code message** |
| 7.2 | $D_S$ | $\xrightarrow[{}^{L3}(E+B)_{DA_1}]{}$ | $U_P$ | : | **Auth code message** |
| 8.1 | $U_C$ | $\xrightarrow[{}^{L3}(E+B)_{DA_1}]{}$ | $D_S$ | : | ***auth code*** |
| 8.2 | $U_C$ | $\xrightarrow[{}^{L2}(E)_{DA_2}]{}$ | $D_S$ | : | ***auth code*** |
| 8.3 | $U_C$ | $\xrightarrow[{}^{L1}(DY)_{DY}]{}$ | $D_S$ | : | *{auth code}* |
| 9.1 | $D_S$ | $\xrightarrow[{}^{L1}(DY)_{DY}]{}$ | $U_C$ | : | {User's page} |
| 9.2 | $D_S$ | $\xrightarrow[{}^{L2}(E)_{DA_2}]{}$ | $U_C$ | : | User's page |
| 9.3 | $D_S$ | $\longrightarrow$ | $U_C$ | : | User's page |

# Capability Formalisation - Example

- formula(L3_E(sent(a,b,m), DA1)).
- formula( forall([xa, xb, xm, xatt],

```
        implies(
                and(
                        Agent(xa),
                        Agent(xb),
                        Honest(xa),
                        Honest(xb),
                        Attacker(xatt),
                        Knows(xa, xm),
                        L3_E(sent(xa,xb,xm),xatt)
                ),
                and(
                        Knows(xb, xm),
                        Knows(xatt, xm),
                        L3_Sender(xa,xm)
                )
        )),
Eavesdrop_L3).
```

## Steps Formalisation - Example

```
formula(
        and(
                L3_E(sent(uc,ds,dropbox_url),da1),
                L3_B(sent(uc,ds,dropbox_url),da1)
        ),
step1).
```

```
formula(
        implies(
                and(
                        L3_E(sent(uc,ds,dropbox_url),da1),
                        L3_B(sent(uc,ds,dropbox_url),da1)
                ),
                L2_E(sent(uc,ds,dropbox_url),da2)
        ),
step2).
```

# DA Combined Knowledge - Conjecture Example

- formula(Knows(da2, password), da2_knows_password).
- formula(Knows(da3, auth_code_msg), da3_knows_code).

# Final remarks

- We proposed a more precise notation for the description of security ceremonies,
  including threat models and attacker types.
- Our **DA** attacker can have **different capabilities in each layer** and may (or not) **share his knowledge** with other attackers.

# Final remarks

- The usage of adaptive and flexible threat models enables:
  - **Specification and test** of security ceremonies;
  - **Analysis of several scenarios** for a given ceremony;
  - Classification of **properties** assured by each scenarios (remains for future work).

# Discussion

- 

Questions????

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

UNIVERSIDADE FEDERAL
DE SANTA CATARINA