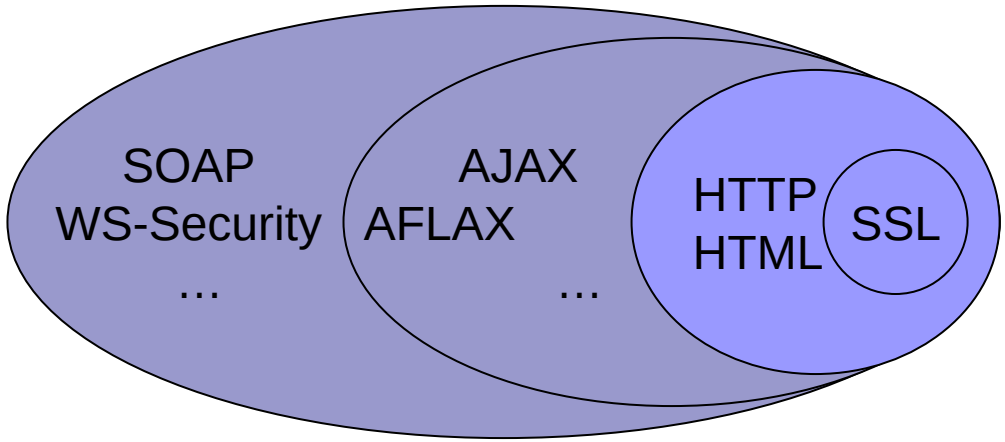
Browser Models for Usable Web Authentication

Sebastian Gajek, Mark Manulis, Ahmad-Reza Sadeghi and Jörg Schwenk

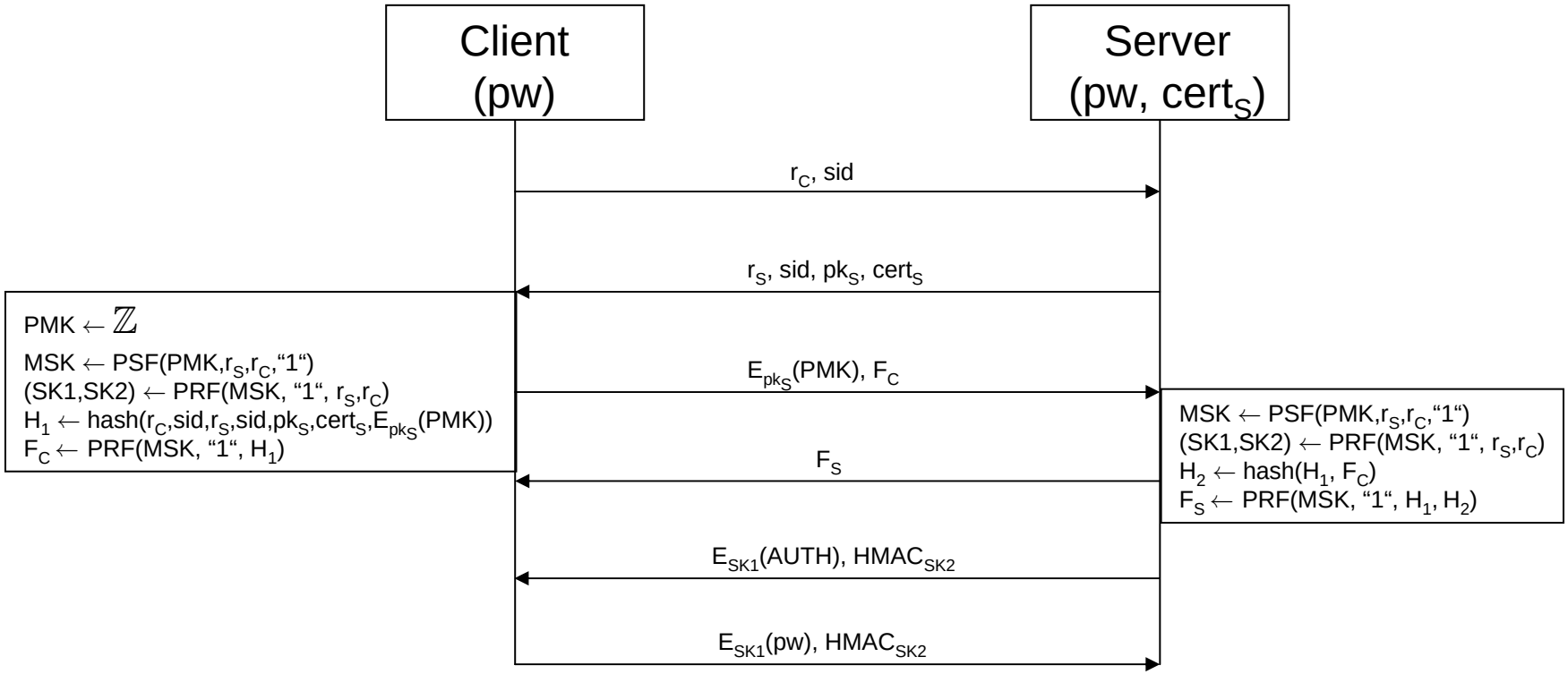
Horst Görtz Institute for IT-Security
Ruhr University Bochum, Germany

Web 2.0 in a Browser



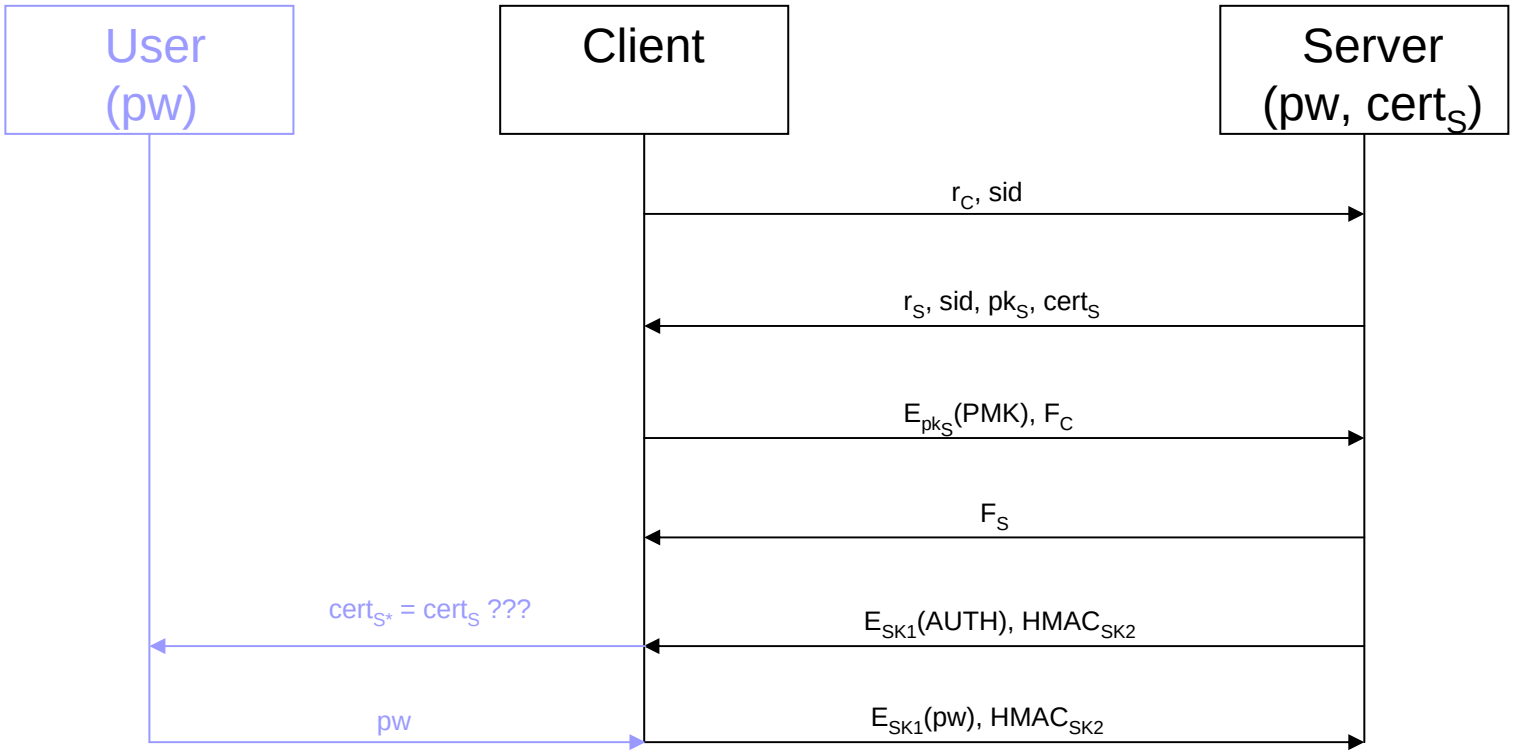
Browser Based (BB) protocols require formal security analysis!

Ceremonies*



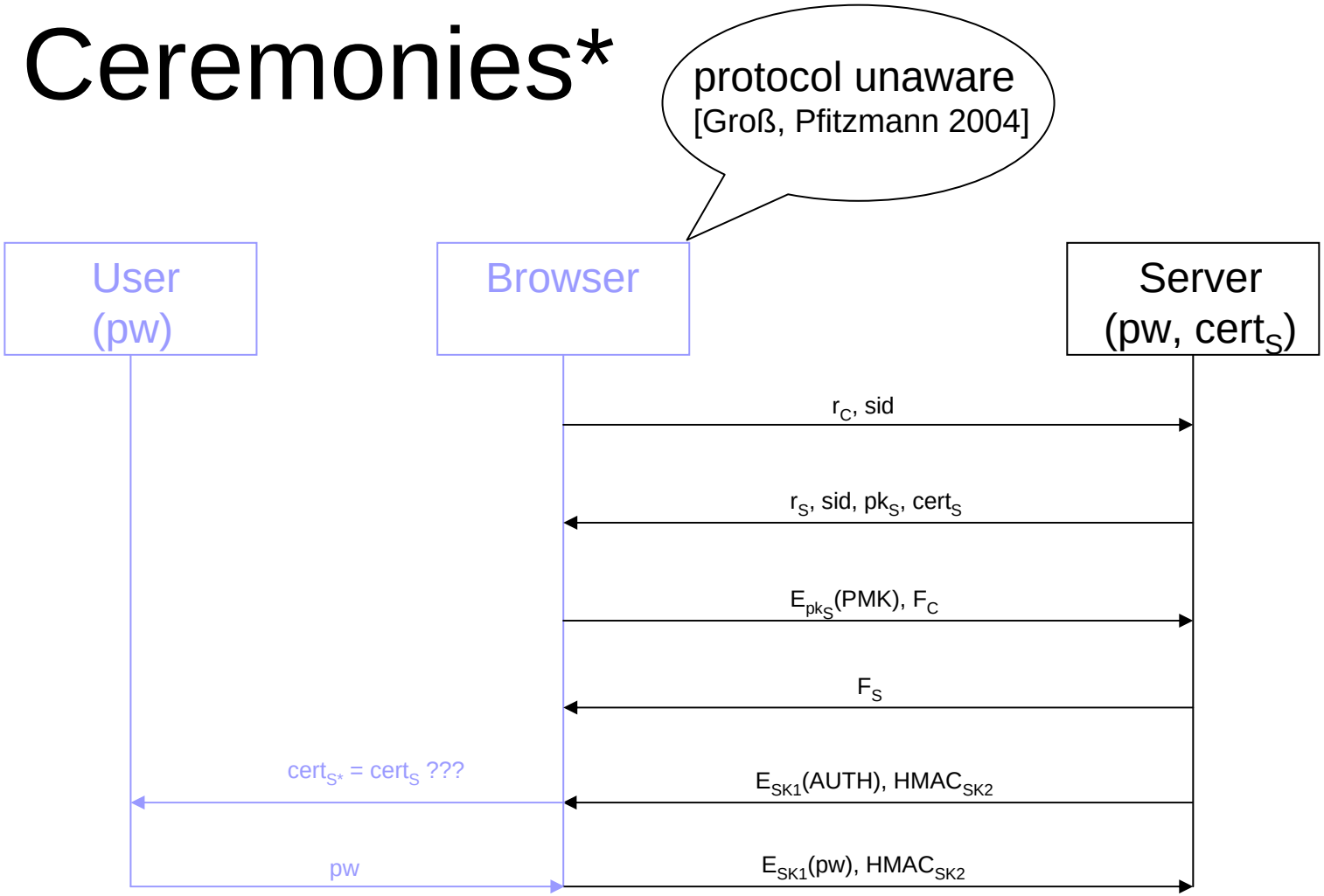
*Carl Ellison: Ceremonies. Crypto Rump Session, 2005.

Ceremonies*



*Carl Ellison: Ceremonies. Crypto Rump Session, 2005.

Ceremonies*



*Carl Ellison: Ceremonies. Crypto Rump Session, 2005.

How could a
computational* model
for BB authentication protocols
look like?

*[Bellare,Rogaway 1993; Bellare,Pointcheval,Rogaway 2000]

On Naive Users…

- User $\mathcal{U}$
  - modeled as a stand-alone party
  - simple PPT Turing machine
    - stores
      - low-entropy password *pw* $\in \mathbb{D}$
        [Yan, Blackwell, Anderson, Grant 2000]
      - additional string (identifier) *w* $\in \mathbb{W}$
        [Suo, Zhu 2005]
    - detects an additional string
      - $\mathrm{detect}(\mathrm{w},\mathrm{w}')$*:* {0,1}$^{|w|} \times$ {0,1}$^{|w'|} \rightarrow$ {true,false}
    - solves human puzzles
      [Canetti, Halevi, Steiner 2006]

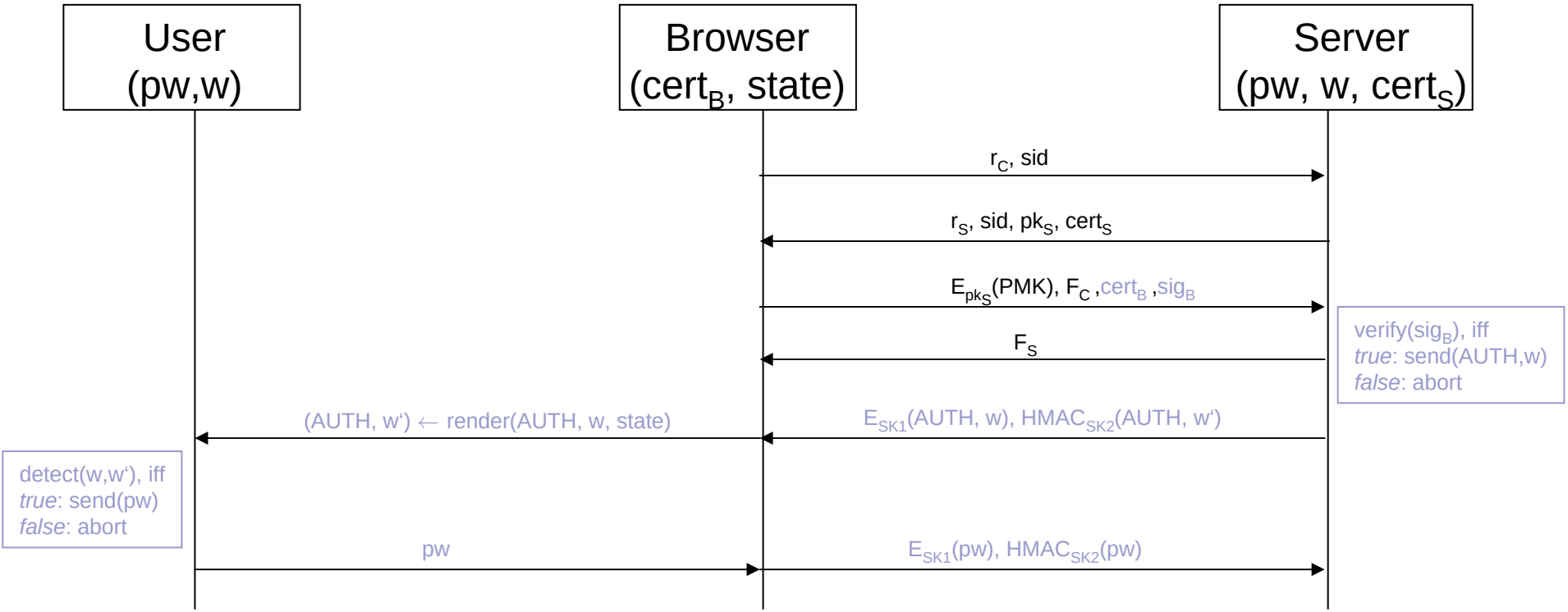
Browsers…

- Browser $\mathcal{B}$
  - simple PPT Turing machine
    - maintains session states *state*
    - visualizes server messages *m*
      - $\mathrm{render}(\mathrm{m},\mathrm{state})$: $\{0,1\}^{|m|} \times \{0,1\}^{|state|} \rightarrow \{0,1\}^{|m'|}$

…and Adversaries

- Adversary $\mathcal{A}$
  - controls all communication of the network
    - $\mathrm{EXECUTE}(\mathcal{U}, \mathcal{S})$: This query models passive attacks, where the adversary $\mathcal{A}$ eavesdrops the execution of the protocol
    - $\mathrm{SEND}(\mathcal{I},\mathrm{m})$: This query models active attacks, where $\mathcal{A}$ sends a message to the instance $\mathcal{I} \in \{\mathcal{U}, \mathcal{B}, \mathcal{S}\}$. The adversary $\mathcal{A}$ receives the response $\mathcal{I}$ generated in processing the message m according to the protocol
    - $\mathrm{EMBED}(\mathcal{S},\mathrm{m},\mathrm{m}^*)$: This query models message injection attacks (e.g., cross-site-scripting, frame spoofing), where the adversary embeds a message m^* into the original message m
    - $\mathrm{REVEAL}(\mathcal{B})$: This query reveals certain information about the browser's states (e.g., cookies, cache, browser version)

Example Authentication Protocol (for Naive Users)



Security

- Correctness

  $\forall w \in \mathbb{W}$: w‘ $\leftarrow$ render(w,state) $\Rightarrow$ detect(w,w‘) $\rightarrow$ true

- Adversary Win

  $\forall w \in \mathbb{W}$, $\forall$ *probabilistic polynomial* $\mathcal{A}$*:*

  Prob [detect(w,w) = true ::*

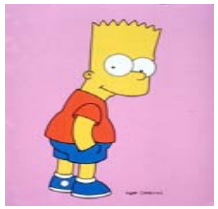
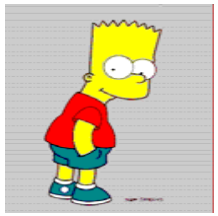
  $w \leftarrow \mathbb{W}$;

  $w' \leftarrow \mathcal{A}(k)$;

  $w^* \leftarrow$ render(w‘,state);] < 1/poly(k)

Outlook/Discussion

- Render identifiers ?
  - $\{w_1, w_2, .., w_n\} \in \mathbb{W}_n$
- Model user uncertainty?
  - $sim(w,w') \rightarrow \{true, false\}$, iff w‘ can be considered sufficiently similar to w‘ w.r.t. threshold σ
  - $sim(\bullet,\bullet)$-threshold σ taken from usability studies

Sim ( , ) = true?

- Correctness

  $\forall w \in \mathbb{W}$: $w' \leftarrow render(w^*,state)$
  $\wedge\ sim(w,w') \rightarrow true$
  $\Rightarrow detect(w,w') \rightarrow true$
- Adversary Win

  $\forall w \in \mathbb{W}$, $\forall$ *probabilistic polynomial* $\mathcal{A}$:

  $Prob\ [\ detect(w,w') = true$
  $\wedge\ sim(w,w') = true\ ::$
  $w \leftarrow \mathbb{W};$
  $w^* \leftarrow \mathcal{A}(k);$
  $w' \leftarrow render(w^*,state);\ ]$
  $< 1/poly(k)$

- Backup

Methods/Models

- Computational

  [Bellare,Rogaway 1993; Bellare,Pointcheval,Rogaway 2000; Pfitzmann,Waidner 2001; Canetti 2001; Herzberg,Yoffe 2007]

- Symbolic

  [Dolev,Yao 1983; Burrows,Abadi,Needham 1990; Meadows 1991; Lowe 1996; Lynch 1996]

- Hybrid

  [Lincoln,Mitchell,Mitchell,Scedrov 1998; Abadi,Rogaway 2000; Backes,Pfitzmann,Waidner 2003; Canetti,Herzog 2004; Blanchet 2005]