University of British Columbia

OpenID Security Analysis and Evaluation

San-Tsai Sun, Kirstie Hawkey, Konstantin Beznosov

Laboratory for Education and Research in Secure Systems Engineering (**LERSSE**)
University of British Columbia

summary

- CSRF attacks
  - single-sign-on CSRF (force victim to login) (70%)
  - account profile CSRF (50%)
  - login CSRF (login as attacker) (73%)
- authentication response interception
  - impersonation (67%)
  - replay attack (6%)

agenda

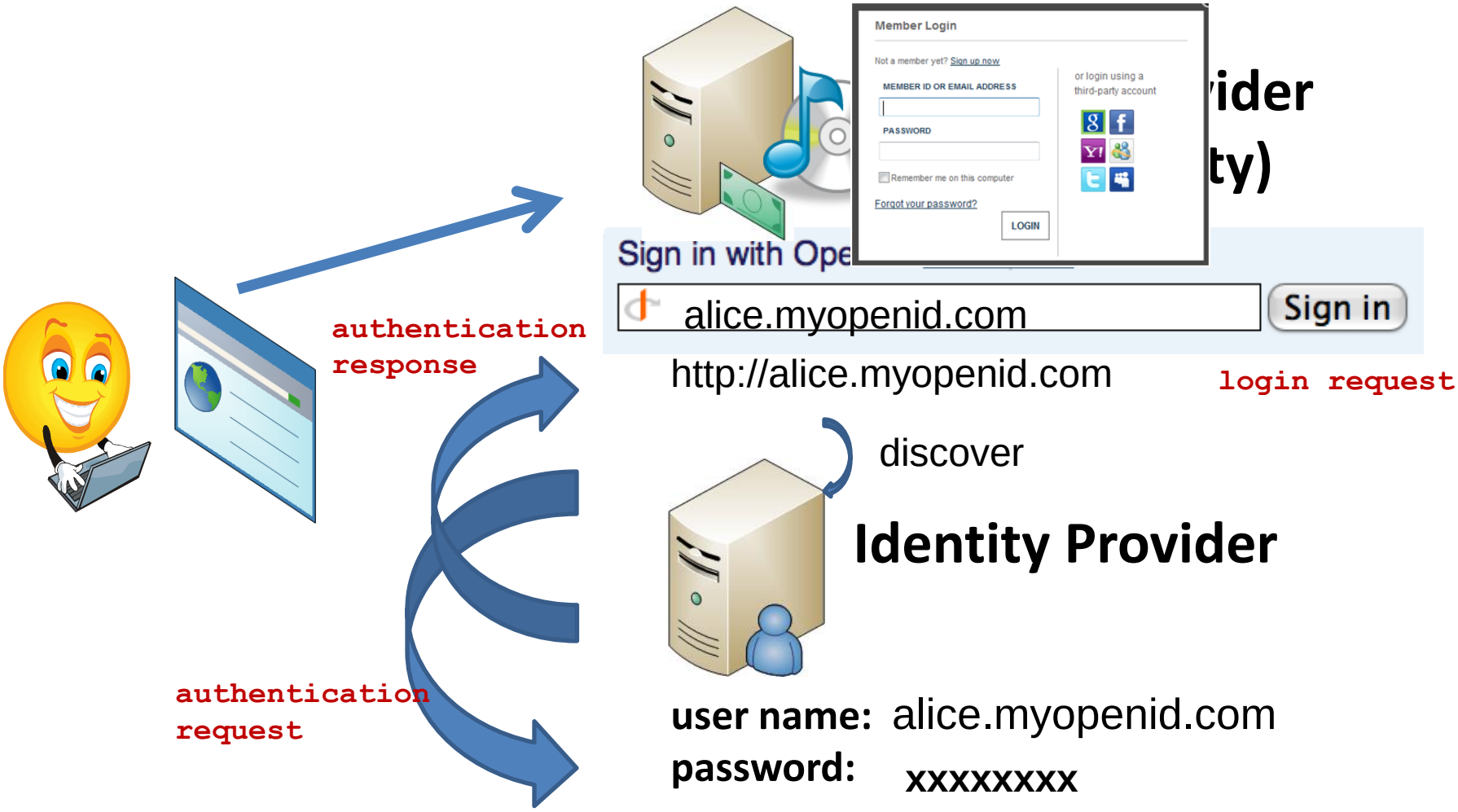
- background
- approach and evaluation result
- countermeasures

OpenID

- open and user-centric Web single sign-on protocol
- OpenID Foundation (2007) [1]
  - Microsoft, Google, IBM, Yahoo, VeriSign, Facebook, PayPal, PingIdentity
- over **one billion** OpenID enabled user accounts provided by Google, Yahoo, AOL…[1]

[1] OpenID Foundation. http://openid.net/foundation/, 2009.

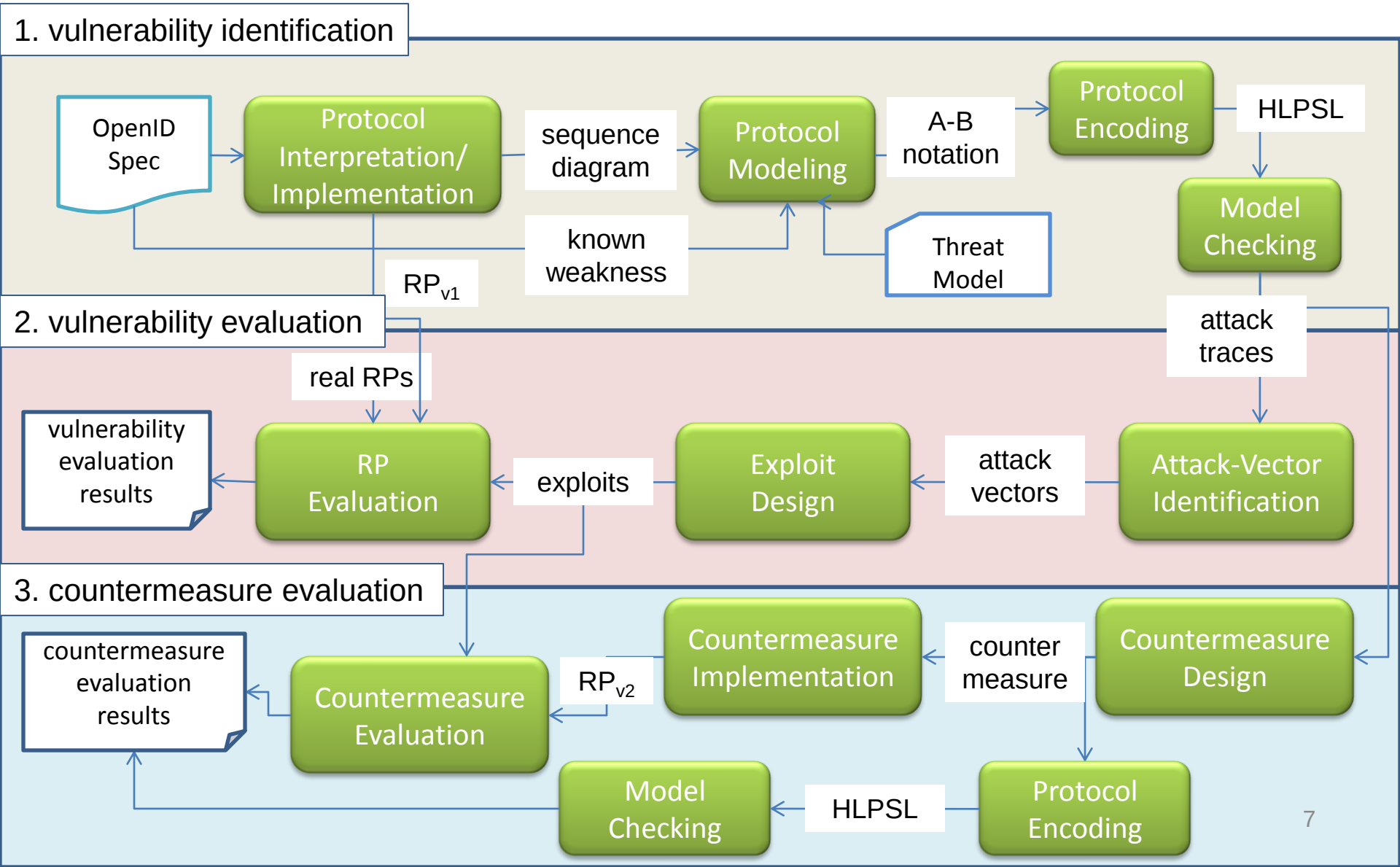
how OpenID works



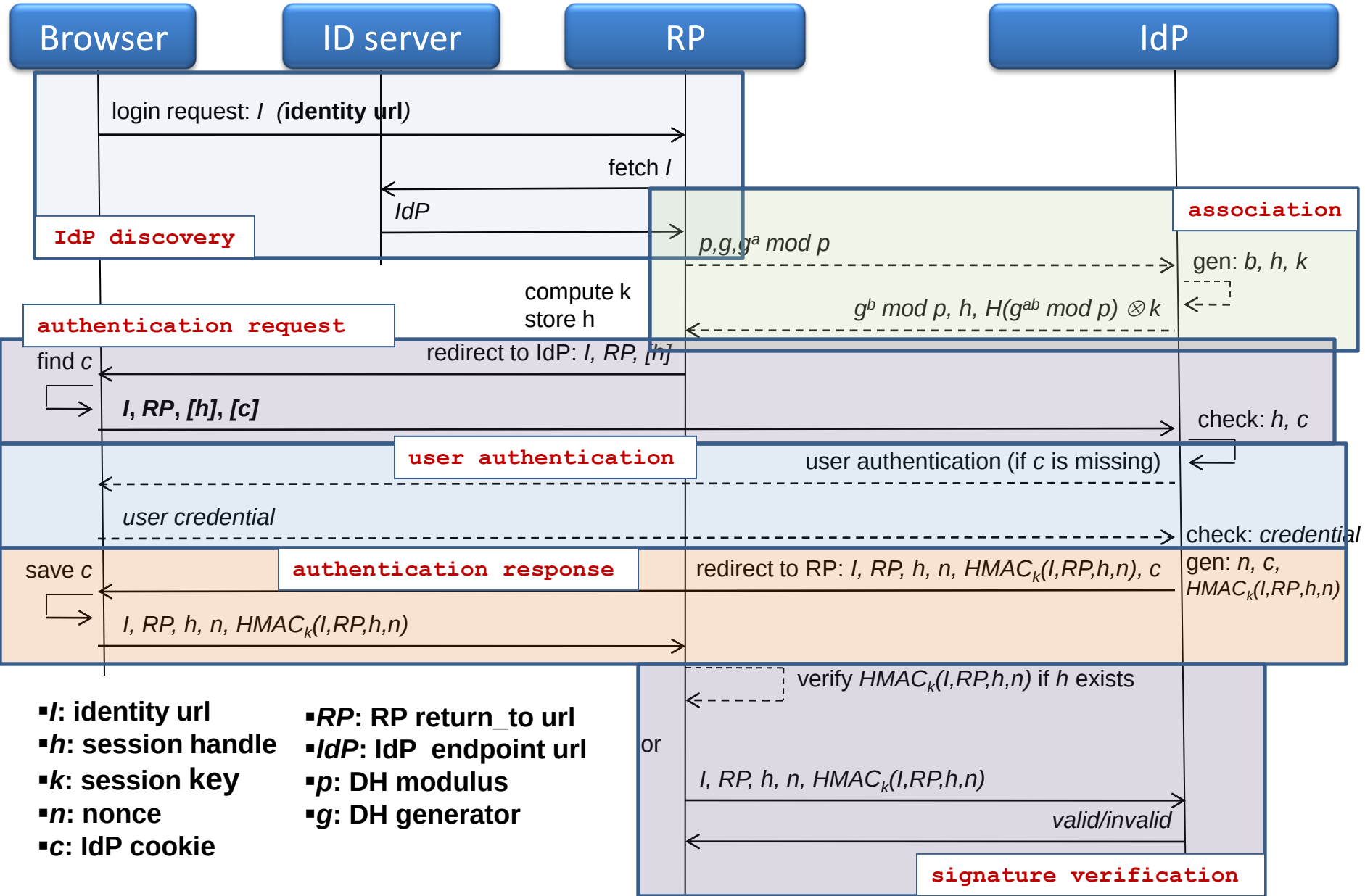
agenda

- background of OpenID
- approach and evaluation results
- countermeasures

methodology



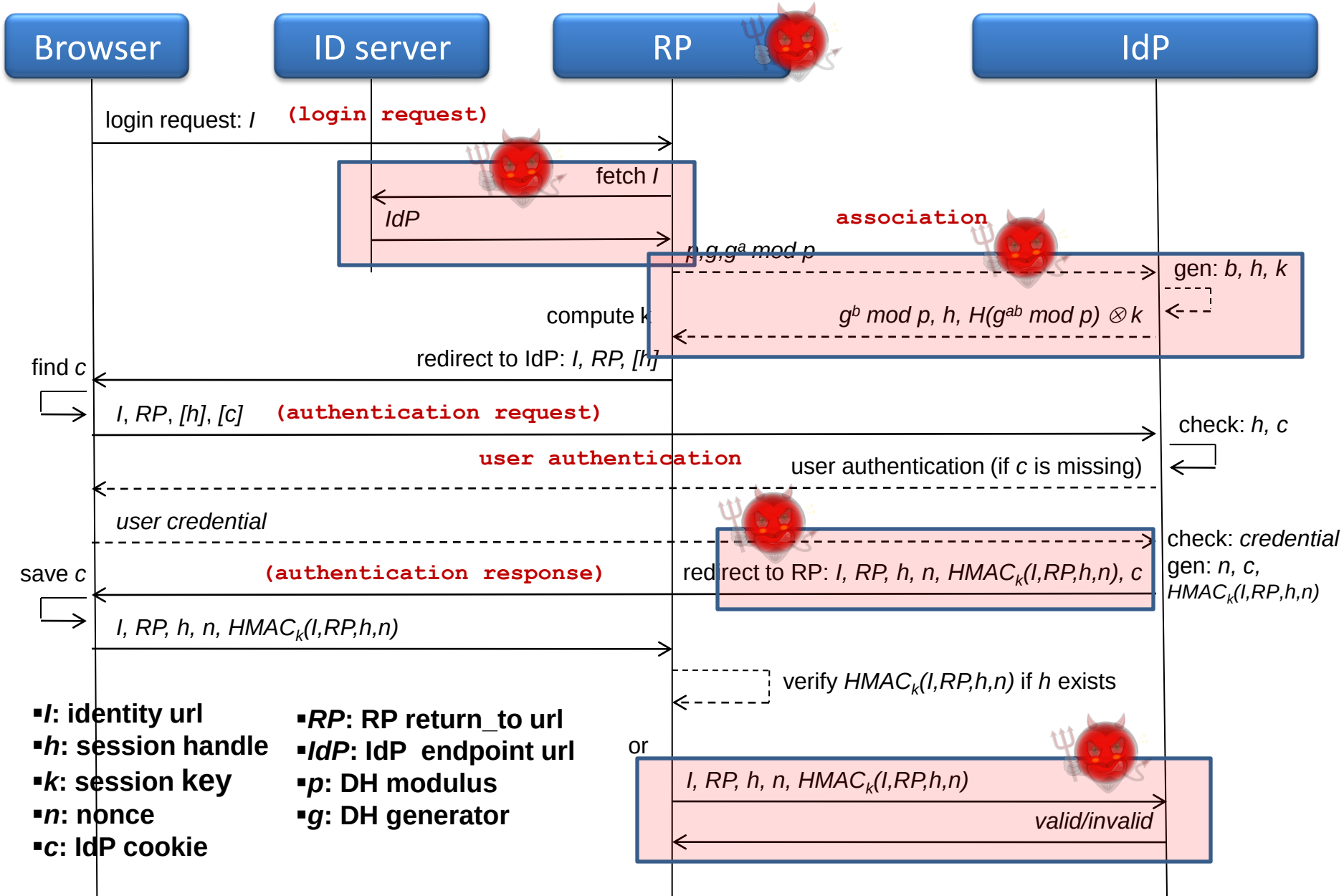
OpenID sequence diagram



- ***I***: identity url
- ***h***: session handle
- ***k***: session key
- ***n***: nonce
- ***c***: IdP cookie

- ***RP***: RP return_to url
- ***IdP***: IdP endpoint url
- ***p***: DH modulus
- ***g***: DH generator

known weakness



- ***I***: identity url
- ***h***: session handle
- ***k***: session key
- ***n***: nonce
- ***c***: IdP cookie

- ***RP***: RP return_to url
- ***IdP***: IdP endpoint url
- ***p***: DH modulus
- ***g***: DH generator

threat model

- adversary: non-RP or IdP associated attackers
- adversary goal: unauthorized access/modification to uses' data hosted on RP
- Web poster: post comments
- Web attacker:
  - setup a malicious website
  - send malicious links via spam
  - deliver malicious content via Ads network
  - exploit web vulnerabilities (i.e., XSS) of benign websites
- network attacker:
  - setup an wireless access point
  - compromise client DNS resolution

threat assumptions

- RP, IdP, user machine, and browser are not compromised
- RP, IdP are not malicious
- user credentials on IdPs are secured
- cookies in the browser are secured (integrity and confidentiality)

non-considered threats

- availability threat
  - DoS by sending massive concurrent auth requests to an IdP
  - DoS by sending massive concurrent auth responses to an RP
- identity spoofing
  - phishing attacks by RP
  - exploits vulnerabilities on IdP
- integrity of IdP discovery process
  - altering discovery information
  - compromise RP DNS resolution

found weakness

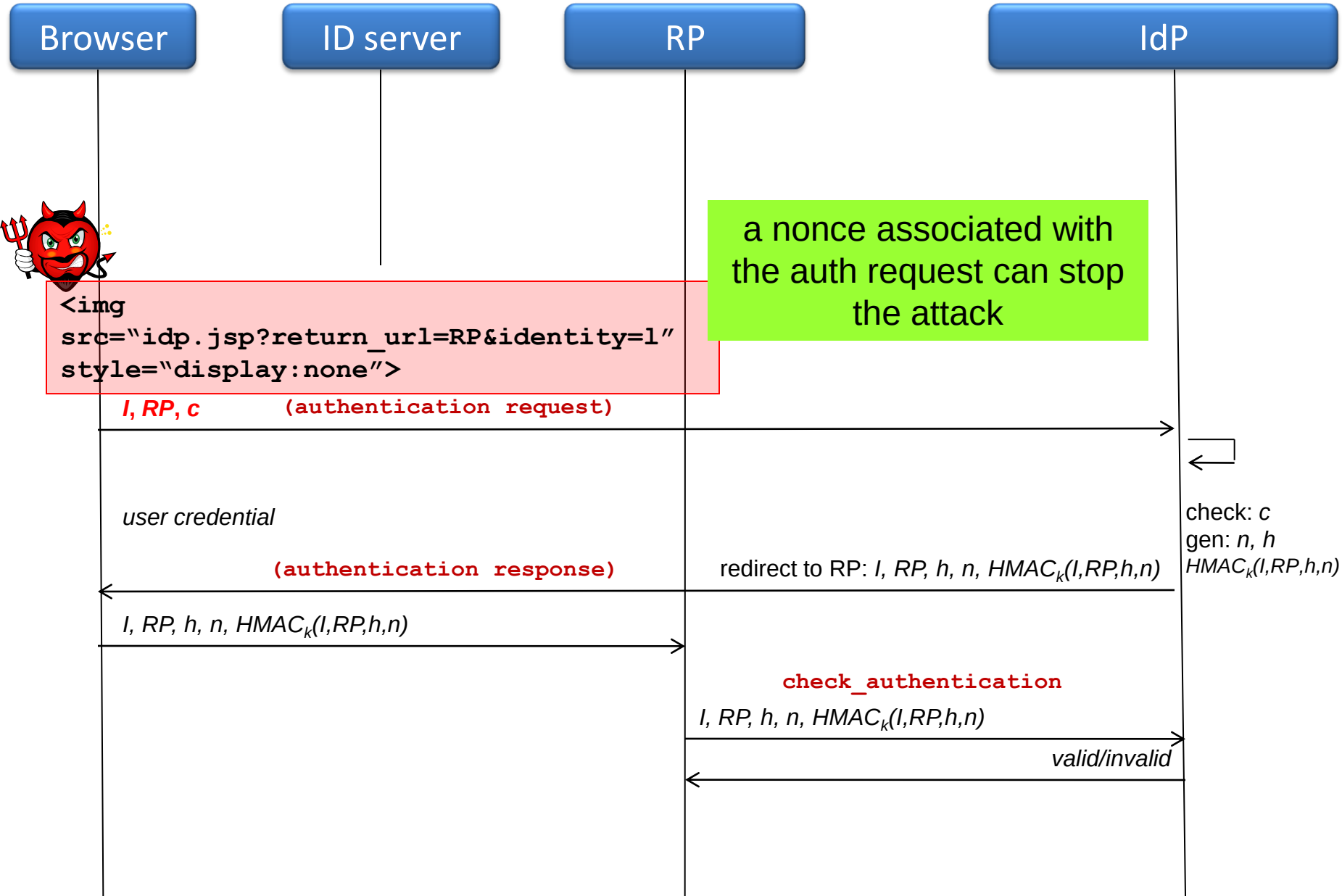
authentication response acts as an one-time access token to an RP, but

- authentication response is not bound to a specific authentication request (non-associate)
- authentication request is not bound to a specific login request
- login request is not bound to the browser session

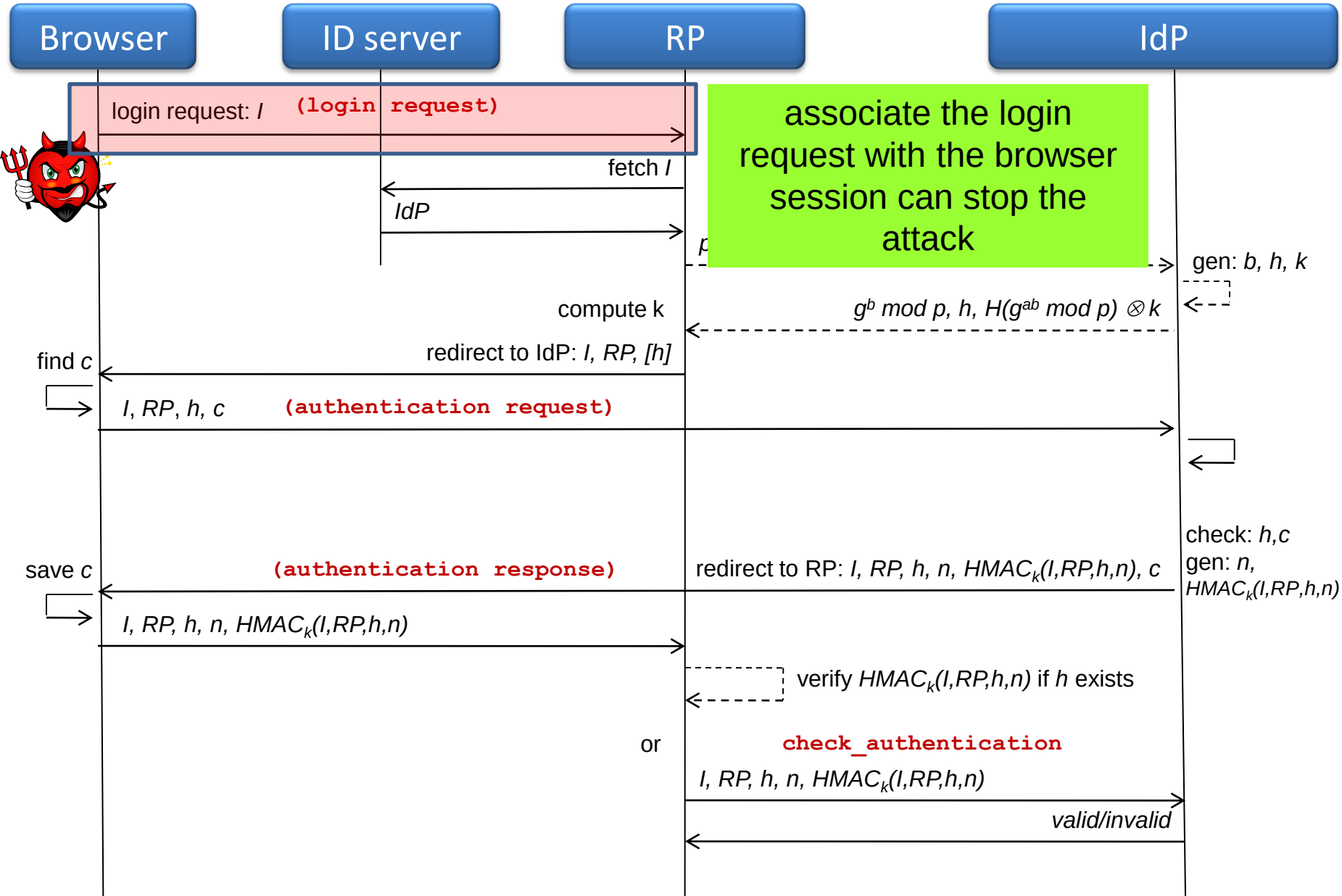
attack vectors

- CSRF
  - single sign-on (SSO) CSRF (force victim to login)
    - HTTP GET Auth Request CSRF [Web poster, Web attacker]
    - HTTP POST Login CSRF [Web attacker]
    - HTTP GET Login CSRF [Web poster, Web attacker]
  - account profile CSRF [Web poster, Web attacker]
  - login CSRF (login as attacker) [Web attacker]
- authentication response interception
  - impersonation [Network attacker]
  - replay attack [Network attacker]

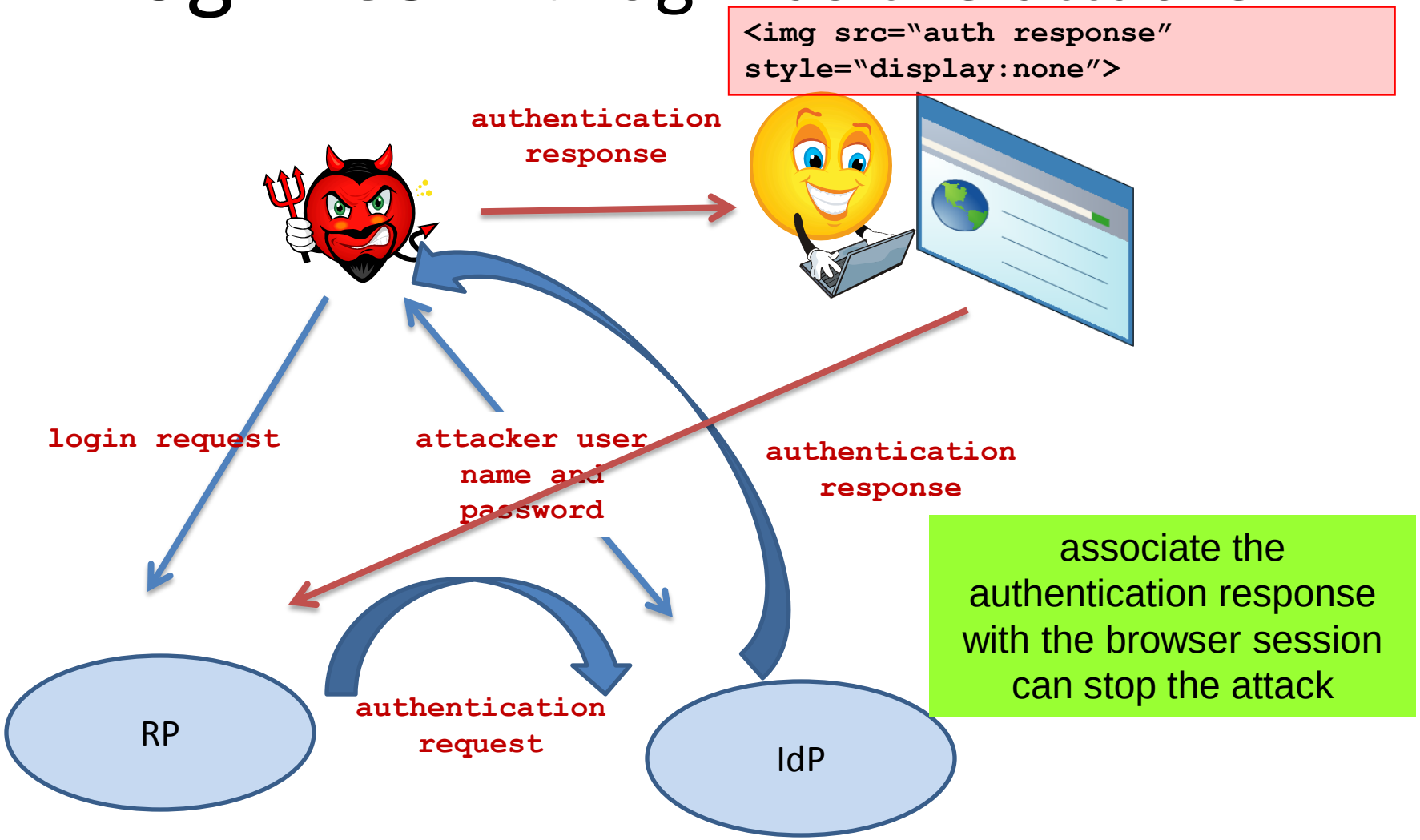
SSO CSRF: HTTP GET Auth Request



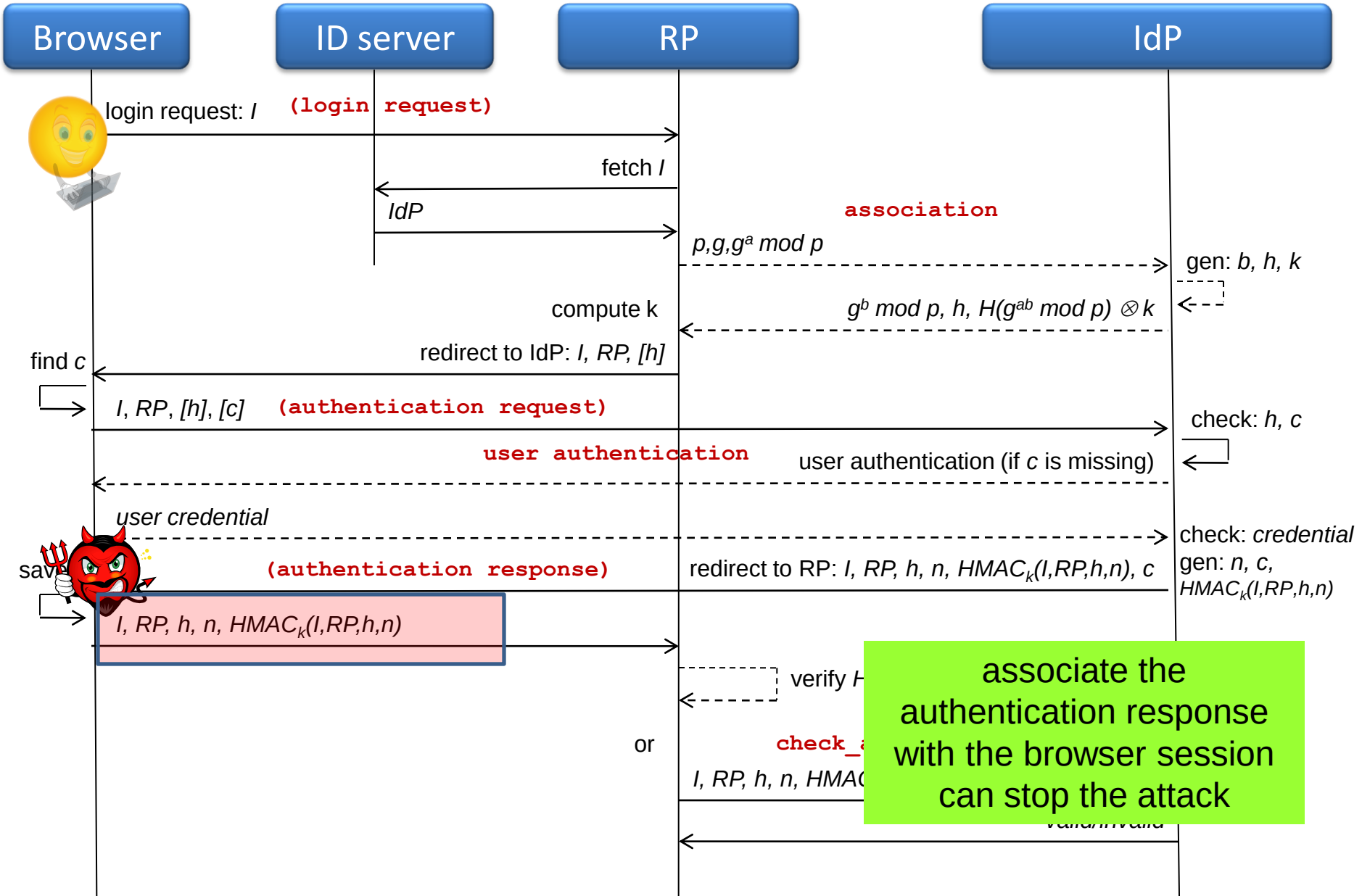
SSO CSRF : HTTP POST/GET login



login CSRF: login as the attacker



impersonation and replay attack



attack summary

- CSRF attacks
  - single-sign-on CSRF (force victim to login) (70%)
    - HTTP GET Auth Request (25%)
    - HTTP POST Login (50%)
    - HTTP GET Login (35%)
  - account profile CSRF (50%)
  - login CSRF (login as attacker) (73%)
- authentication response interception
  - impersonation (67%)
  - replay attack (6%)

agenda

- background of OpenID
- approach and evaluation result
- countermeasures

countermeasure

- when a new browser session initialized
  - RP generates a nonce N = HMAC(browser session id)
  - issues a new cookie $C_N = N$
  - append a parameter P_N=N to the OpenID login form
- when receive a login request
  - check if $P_N = C_N$ and C_N = HMAC(browser session id)
  - initiate a new authentication request
  - append a parameter R_N=N to the **`return_to`** URL
- when receive an authentication response
  - check if $R_N = C_N$ and C_N = HMAC(browser session id)

characteristics of countermeasure

- compatible with existing OpenID
- do not require any additional storage on RP
- would not reveal browser session id
- protect from cookie overwrite

future work

- evaluate more RPs
- apply our methodology to other Web single sign-on protocol
  - Facebook connect
  - Microsoft Live ID

OpenID Security Analysis and Evaluation

san-tsai sun <santsais@ece.ubc.ca>

Department of Electrical and Computer Engineering
Laboratory for Education and Research in Secure Systems Engineering (**LERSSE**)