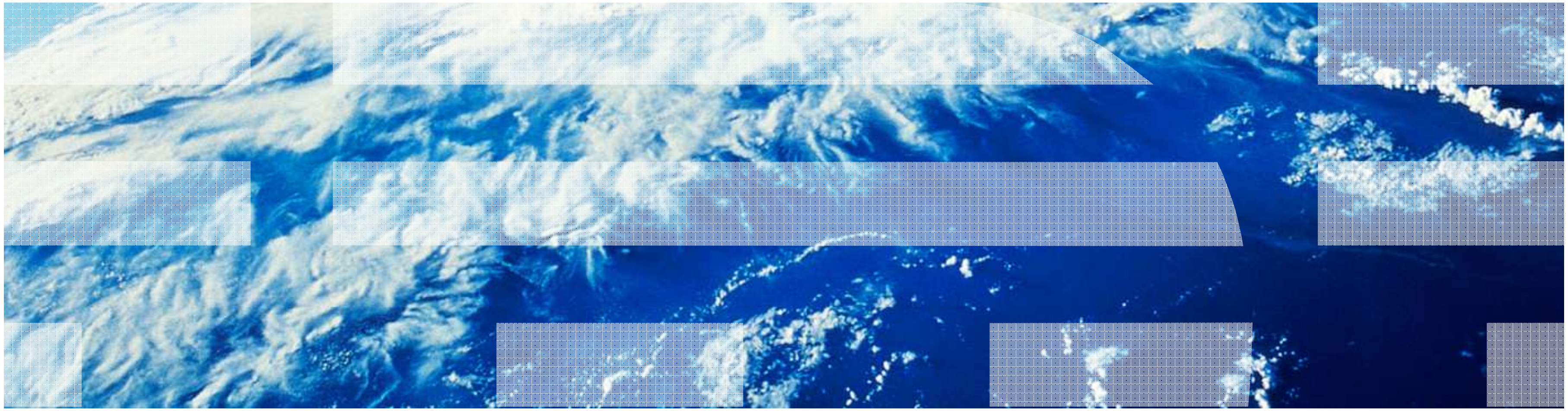
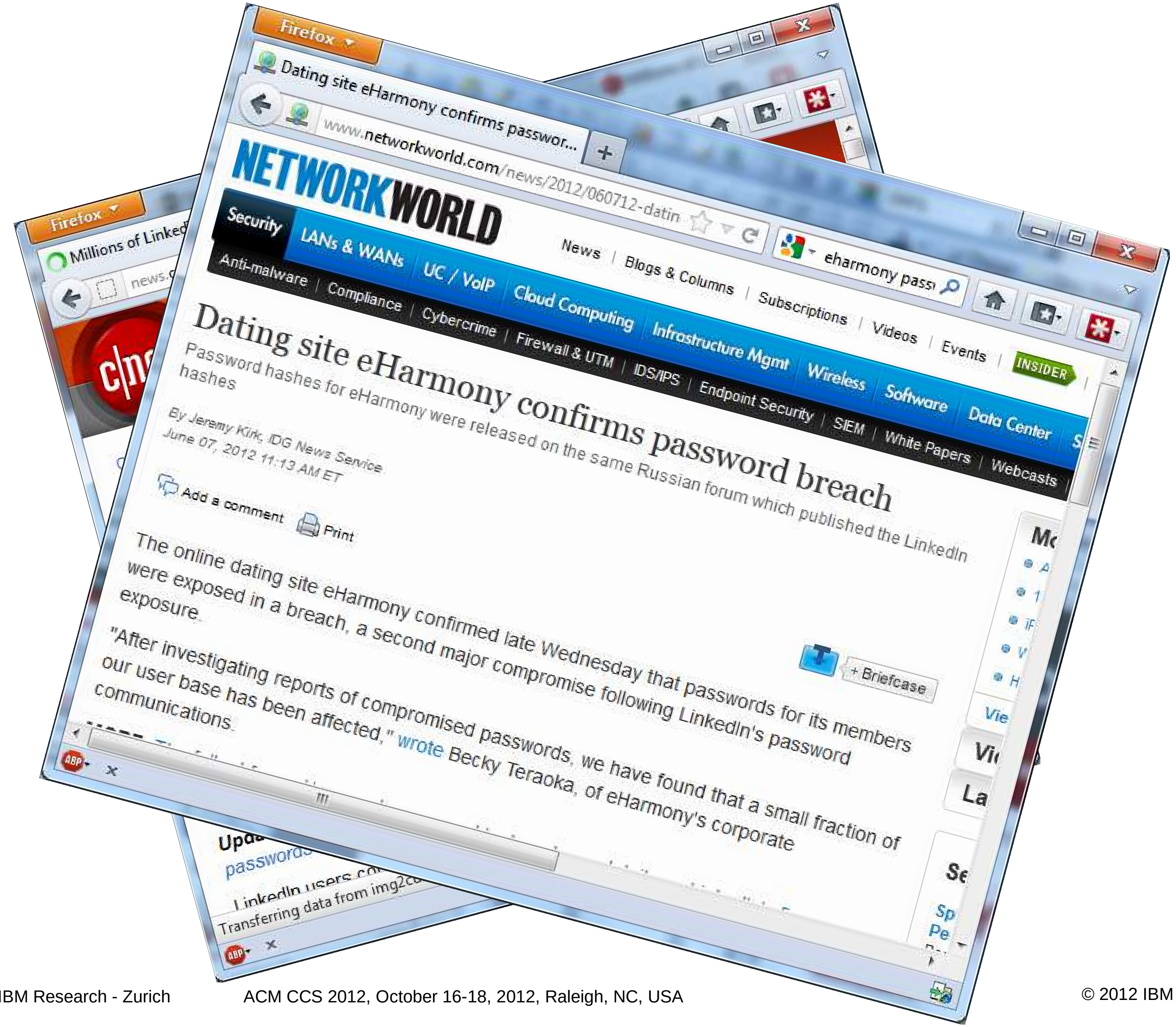
Practical Yet Universally Composable Two-Server Password-Authenticated Secret Sharing



Jan Camenisch (IBM Research – Zurich)
Anna Lysyanskaya (Brown University)
Gregory Neven (IBM Research – Zurich)

Password breaches summer 2012

Password breaches summer 2012



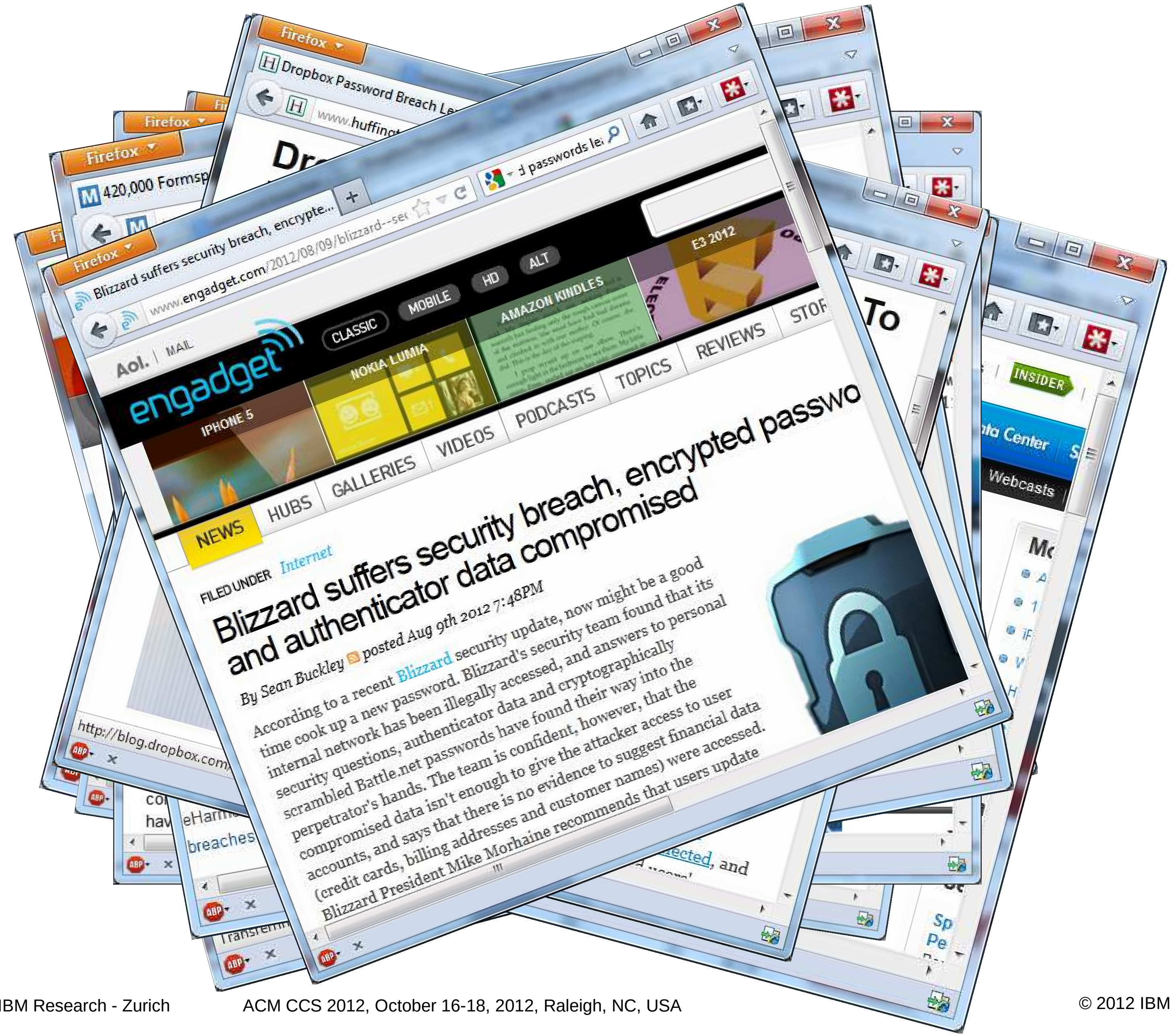
Password breaches summer 2012

Password breaches summer 2012

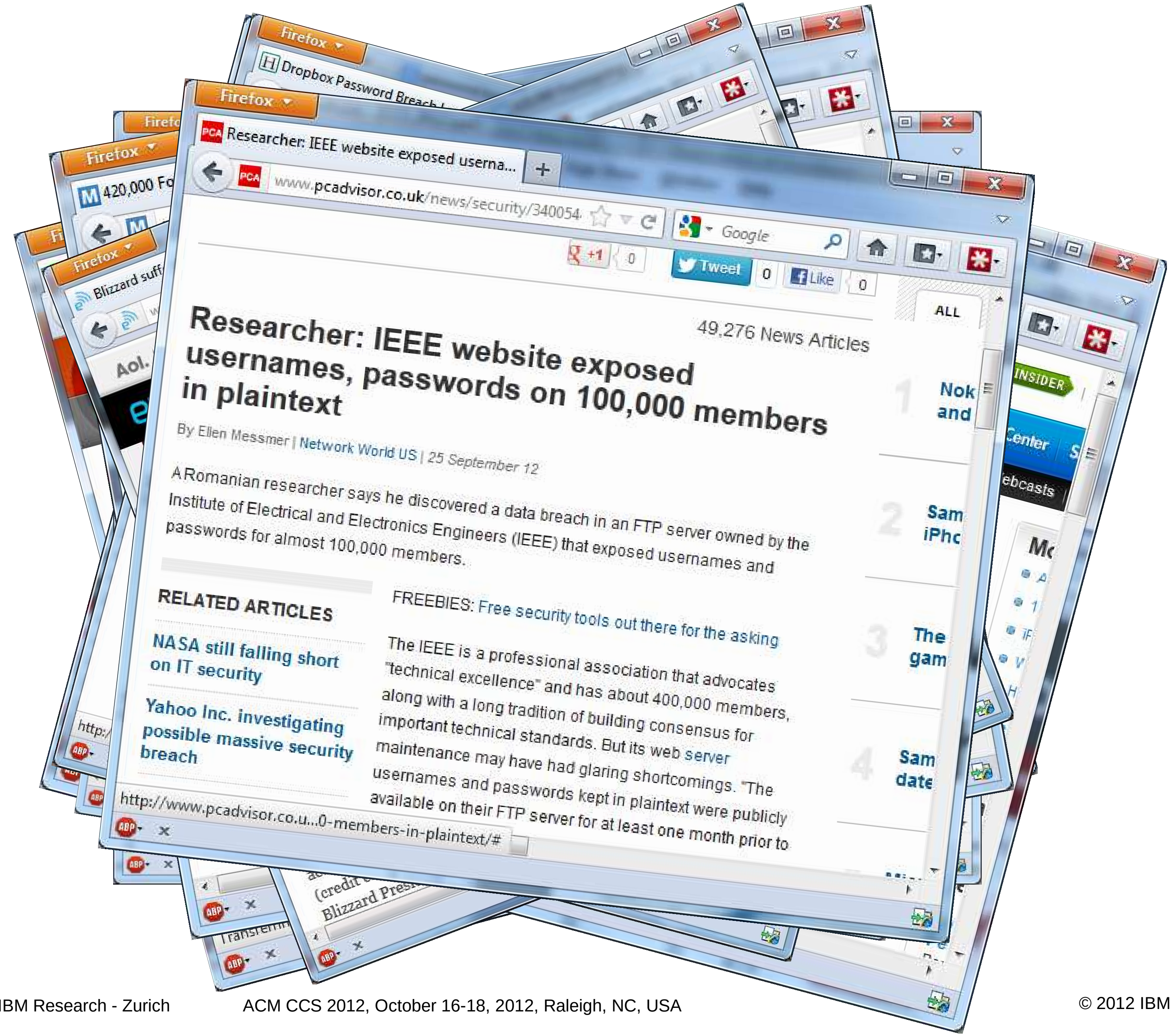
Password breaches summer 2012

Password breaches summer 2012

Password breaches summer 2012



Password breaches summer 2012



Remedies

- Use strong, hard-to-guess passwords
- Use different passwords for different sites
- Change passwords on regular basis
- Don’t write passwords down

… yeah, right.

The root of the problem (1)

- Passwords inherently insecure?
  No! We're just using them incorrectly

- **Problem 1:** Passwords useless against offline attacks
  - 16-character passwords ≈ 30 bits entropy [NIST] ≥ 1bn possibilities
  - $150 GPUs test several bn per second
  - 60% of LinkedIn passwords cracked within 24h
  - Online cracking services:
    $17 for 3bn words, $136 for 25tn, ready in 2h

- Passwords quite effective against online attacks, iff "throttling":
  block account/IP, CAPTCHAs, time delays after failed attempts

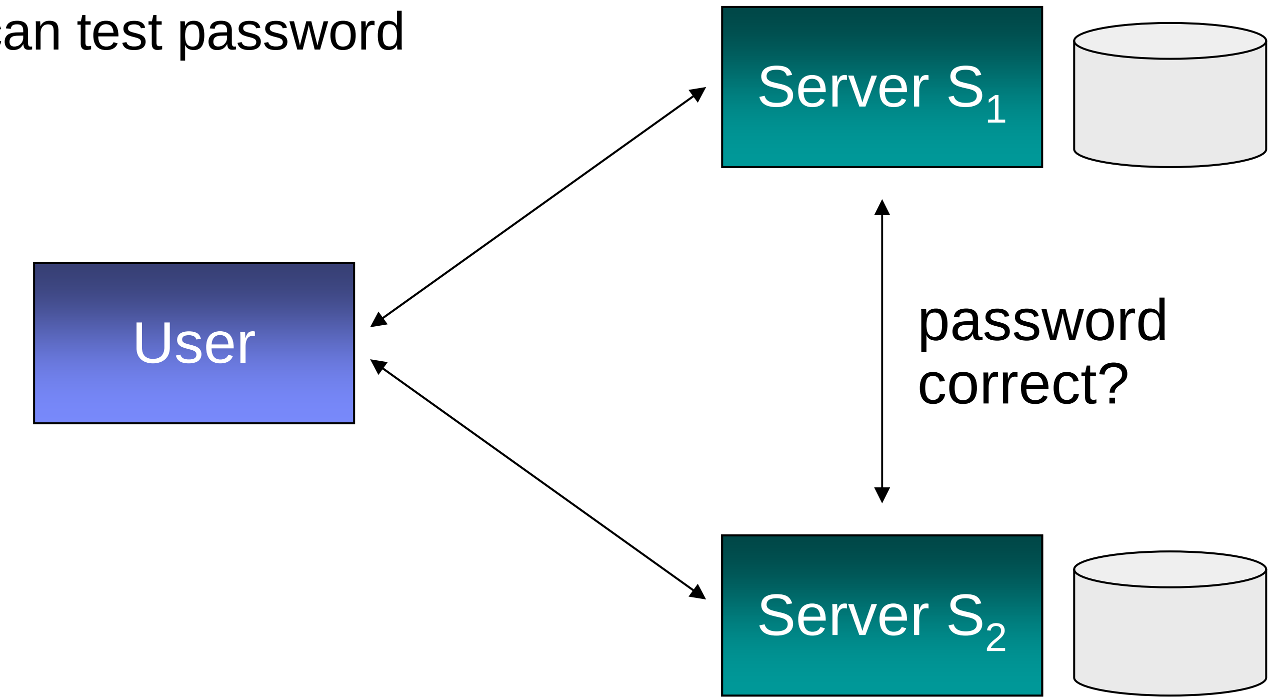
The root of the problem (2)

- **Problem 2:** Single-server solutions inherently vulnerable to offline attacks:

  server/hacker can always guess & test

- **Solution:** multi-server password verification protocols

  no server alone can test password



Overview

- Design goals & related work
- Universally composable security for 2PASS
- Protocol idea
- Conclusion

Overview

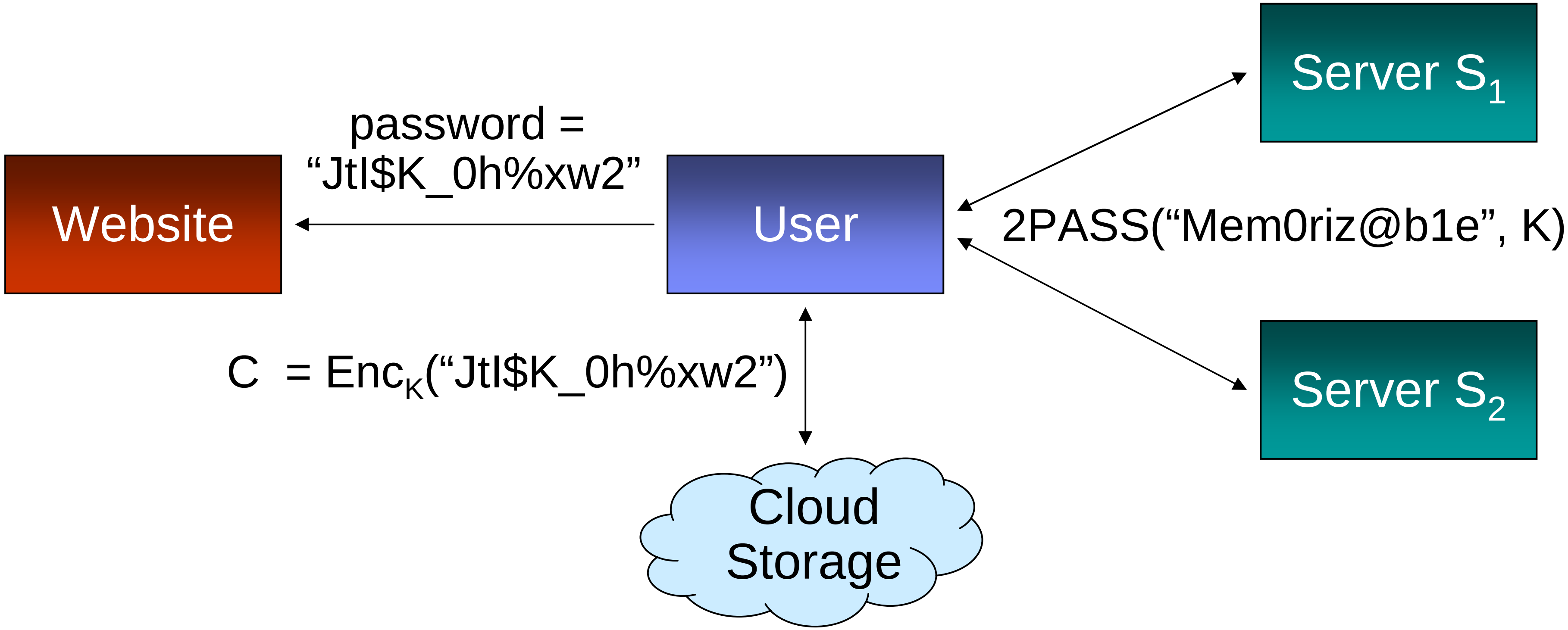
- Design goals & related work
- Universally composable security for 2PASS
- Protocol idea
- Conclusion

Design goals

Two-server password-authenticated secret sharing (2PASS)

- User remembers username, password, server names
- Store & reconstruct strong secret K
  (and thereby any encrypted data)
- Single corrupt server cannot
  - perform offline attack on password (attempts)
  - learn stored secret K
  - convince honest server that wrong password is correct
- Assume PKI
  (e.g., hardcoded in client software)

Killer application: distributed password store



- One person, one password
- No (hacker of) single 2PASS server can steal website passwords
- Websites keep ordinary username/password infrastructure

Related work

Multi-server password-authenticated secret sharing (PASS)

- Bagherzandi et al. (2011):
  t-out-of-n servers, property-based security proofs

Multi-server password-authenticated key exchange (PAKE)

- Ford-Kaliski (2000), Jablon (2001), Brainard et al. (2003):
  no security proofs
- MacKenzie et al. (2002), Di Raimondo-Gennaro (2003), Szydlo-Kaliski (2005), Katz et al. (2005)
  property-based proofs

Long line of work on single-server PAKE

Our contribution: first universally composable (UC) 2PASS

Overview

- Design goals & related work
- Universally composable security for 2PASS
- Protocol idea
- Conclusion

The case for UC in password-based protocols

Property-based definitions

- Passwords chosen according to known, independent distributions
  (reality: people reuse, share, leak info about passwords)
- Adversary sees authentications with **correct** password only
  (reality: typos!)
- Problematic w.r.t. composition due to non-negligible attack probability

UC definitions

- Environment chooses passwords & password attempts
  → no assumptions on distributions, typos covered
- Composes nicely with itself & other protocols

Overall: UC more natural & practically relevant for password-based protocols

2PASS ideal functionality

- At most one corrupt server
  - –doesn't learn anything about
    - password p
    - password attempt q
    - key K
  - –learns whether p=q only if all honest servers cooperate (throttling)
  - –cannot set honest user up with wrong K'
  - –cannot make honest server accept if p≠q
- Further complications to the model
  - –Avoid implying Byzantine agreement or atomic broadcast
  - –Multiple concurrent setup/retrieve queries
  - –Query hijacking

Overview

- Design goals & related work
- Universally composable security for 2PASS
- Protocol idea
- Conclusion

Protocol: high-level idea

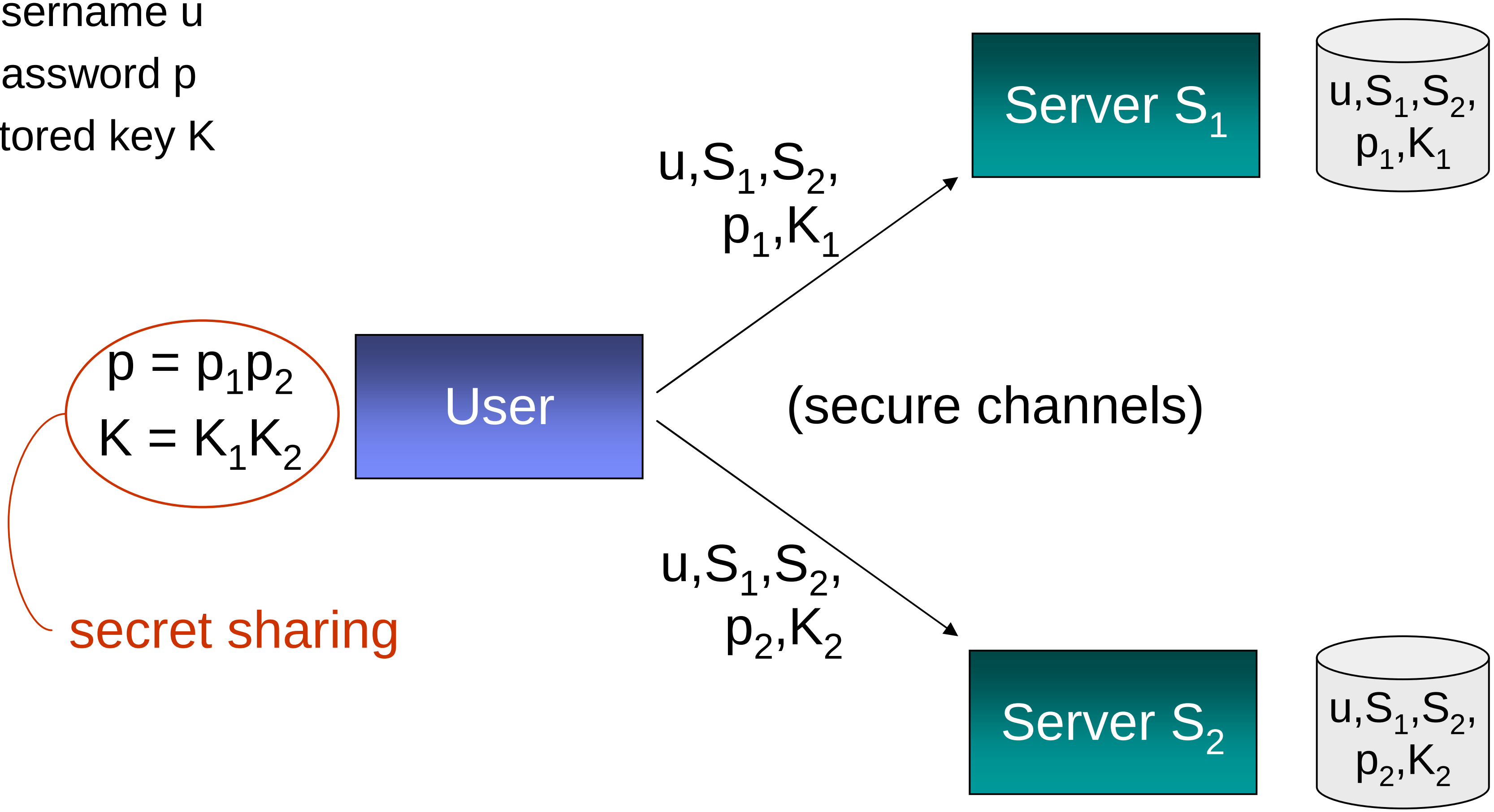
Setup

Brainard et al. (2003)

username u

password p

stored key K



Protocol: high-level idea

Retrieve

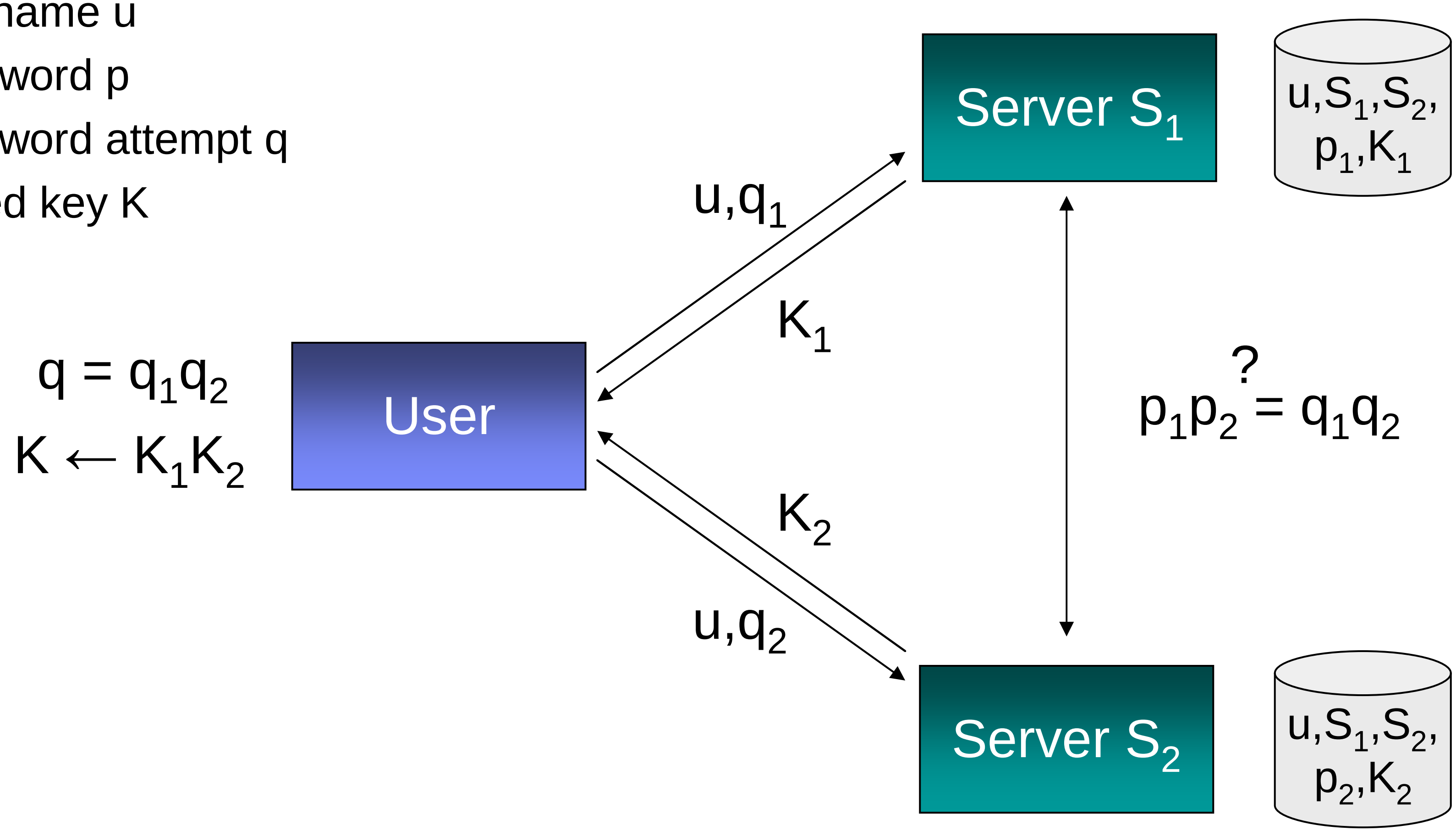
Brainard et al. (2003)

username u

password p

password attempt q

stored key K



Setup

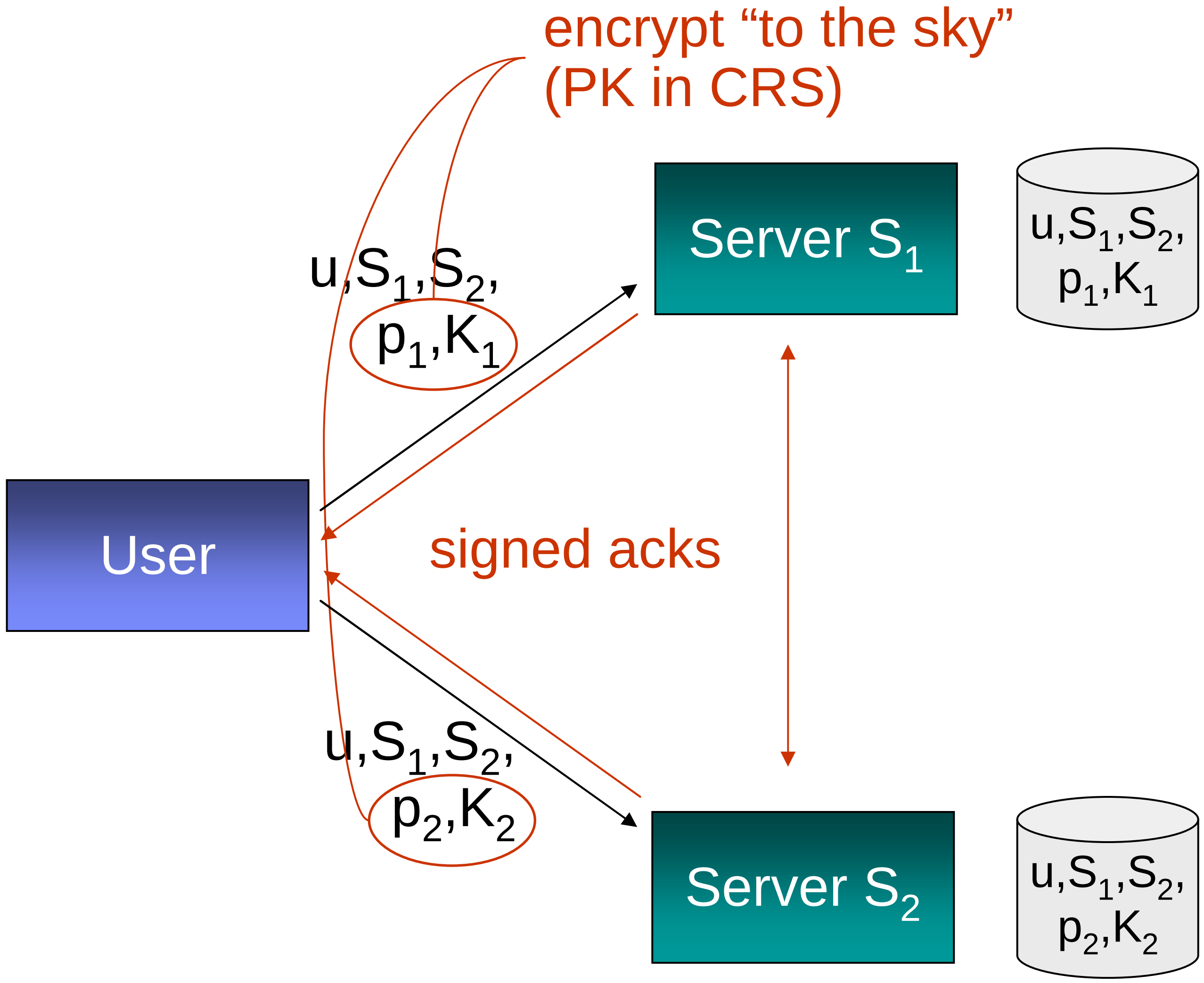
username u

password p

stored key K

$p = p_1 p_2$

$K = K_1 K_2$



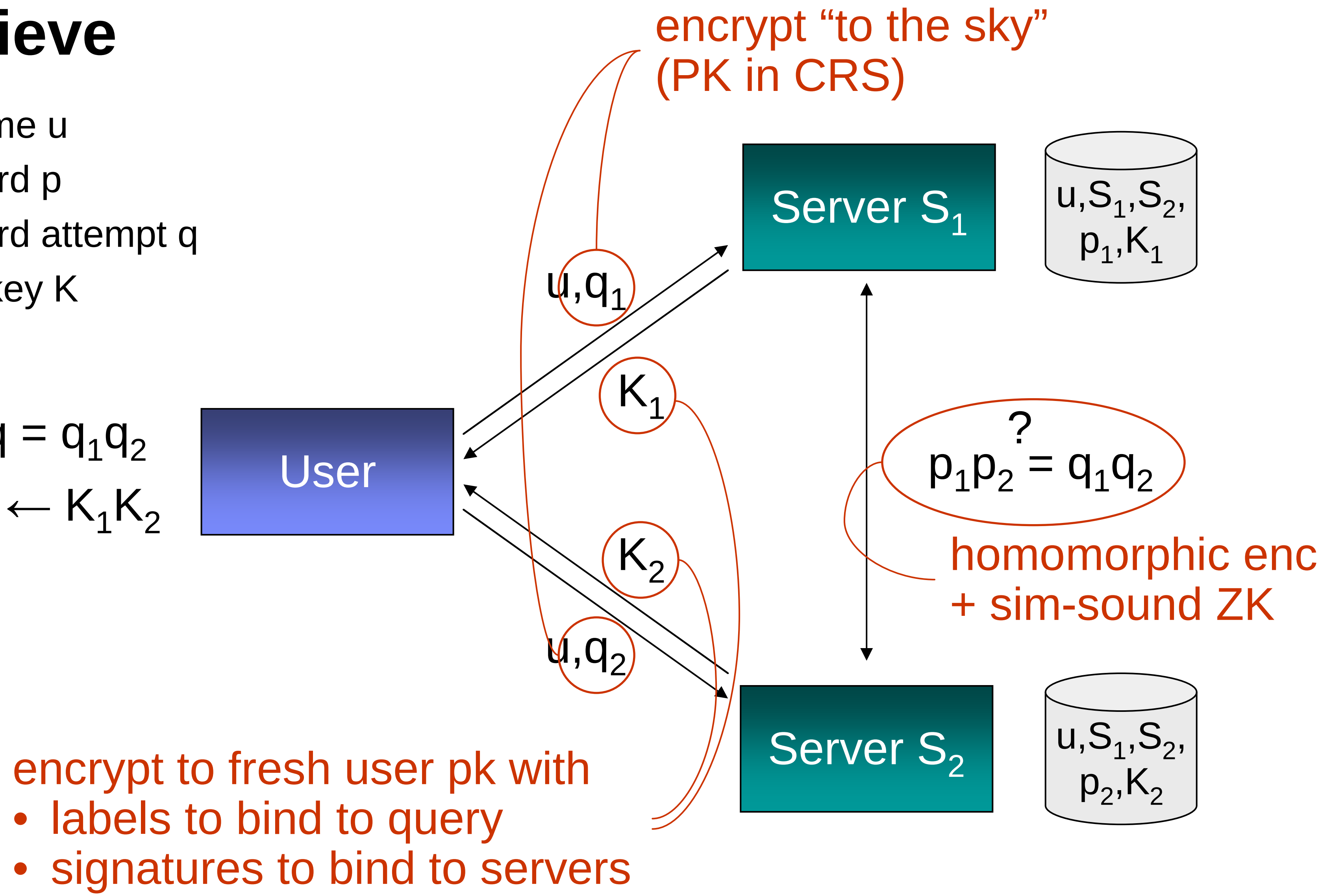
Protocol: UC strengthening

Retrieve

username u
password p
password attempt q
stored key K

$q = q_1 q_2$

$K \leftarrow K_1 K_2$



Efficiency

- Avoid heavy UC machinery by careful design
  - F_{CERT} and F_{CRS} as only UC subcomponents
  - CPA encryption to sky by revealing randomness
  - Efficient ZKPs in random-oracle model
- Most heavy lifting done by servers
- No additional secure channels needed (e.g., SSL)

	Setup	Retrieve
User comp	18 exp	19 exp
Server comp	10 exp	30 exp
User/Server comm	17 el	16 el
Server/Server comm	1 el	6 el, 9 exp, 3 hash

Overview

- Design goals & related work
- Universally composable security for 2PASS
- Protocol idea
- Conclusion

Conclusion

- One person, one password
- Single-server solutions doomed, multi-server way to go
- UC security most natural model – beyond composability
- Efficiency through careful crypto design
- Many open problems – stay tuned ☺
  - t-out-of-n PASS
  - Adaptive corruptions
  - Password-only (i.e., no public keys)
  - …