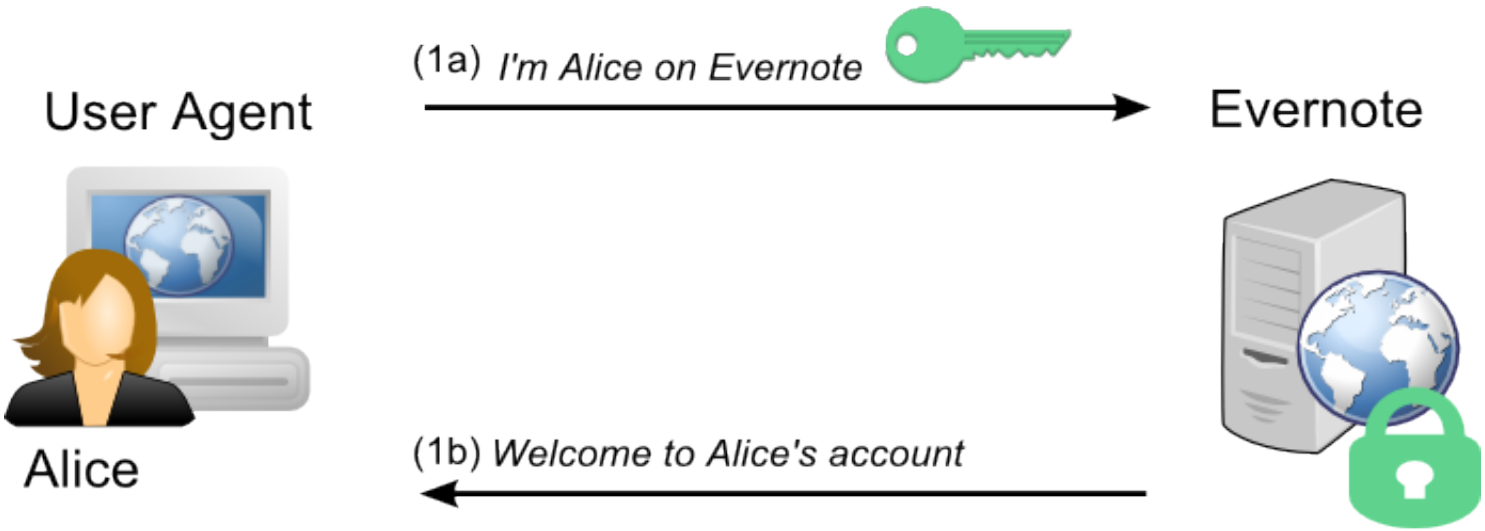
SAuth: Protecting User Accounts from Password Database Leaks

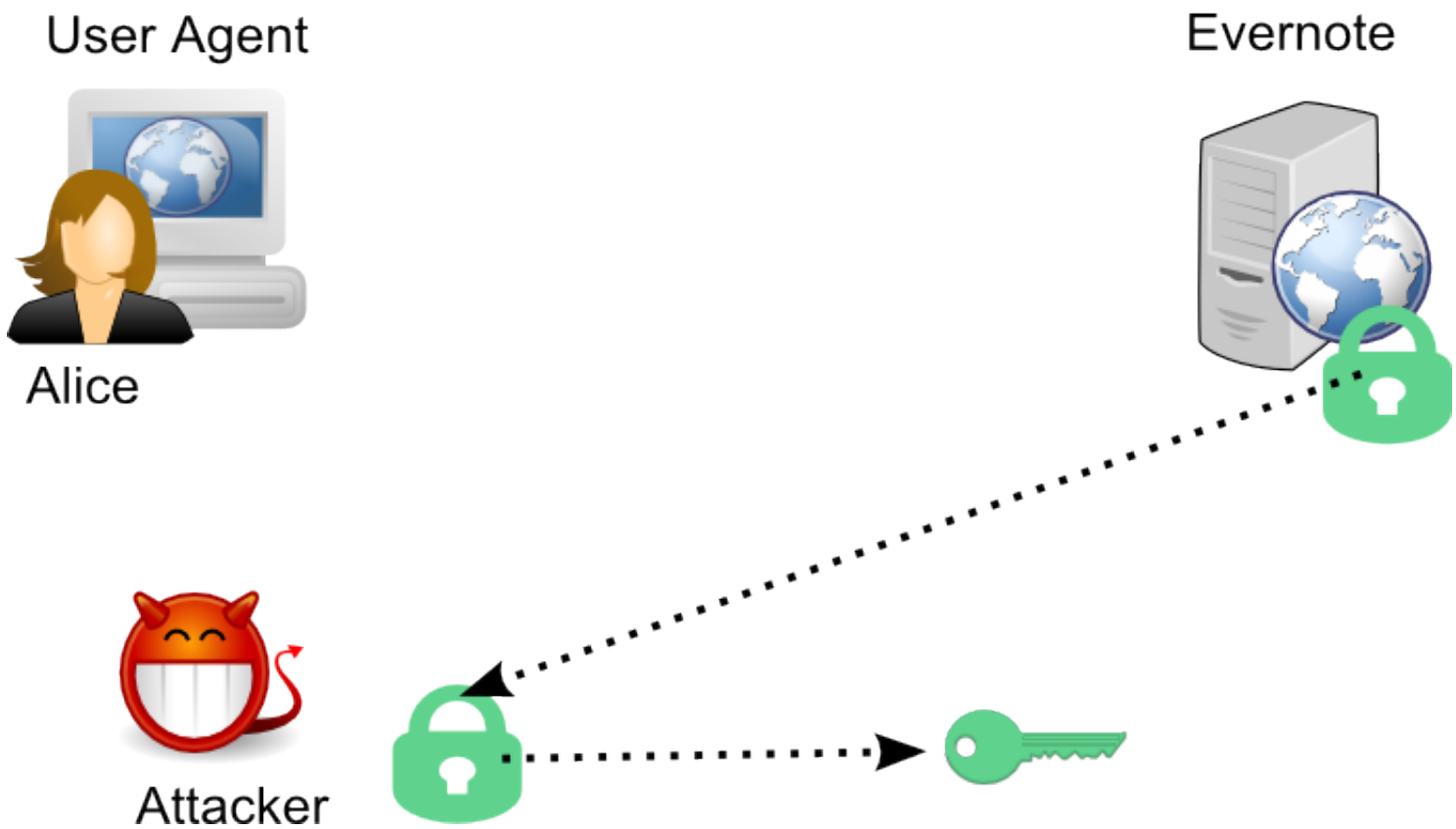
Georgios Kontaxis[‡], Elias Athanasopoulos[‡]
Georgios Portokalidis[*], Angelos Keromytis[‡]

[‡]Columbia University
[*]Stevens Institute of Technology

Authentication recognizes passwords not users …



… and unfortunately passwords get leaked



With stolen password, Attacker impersonates Alice

Password leaks happen all the time

- May go unnoticed until it’s too late

2009	RockYou Gaming	32.0 million
2010	Gawker Media *Domino attack prompted resets in other sites*	1.5 million
2011	Sony	1.0 million
2012	LinkedIn	6.5 million
2013	Twitter *Before being detected and shut down*	250.000
2013	Adobe	150.0 million

Passwords get cracked all the time

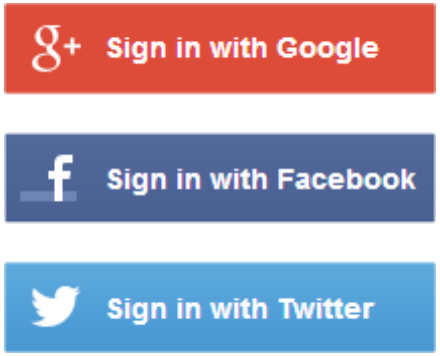
- Weak passwords
  - short, dictionary words, names, patterns, etc.

- Fast hardware
  - Commodity parallel architectures (GPUs)
  - Cloud-powered cracking platforms
    - 6 days after the 6.5 million LinkedIn password leak, 90% of them were cracked

Enhanced Authentication Today

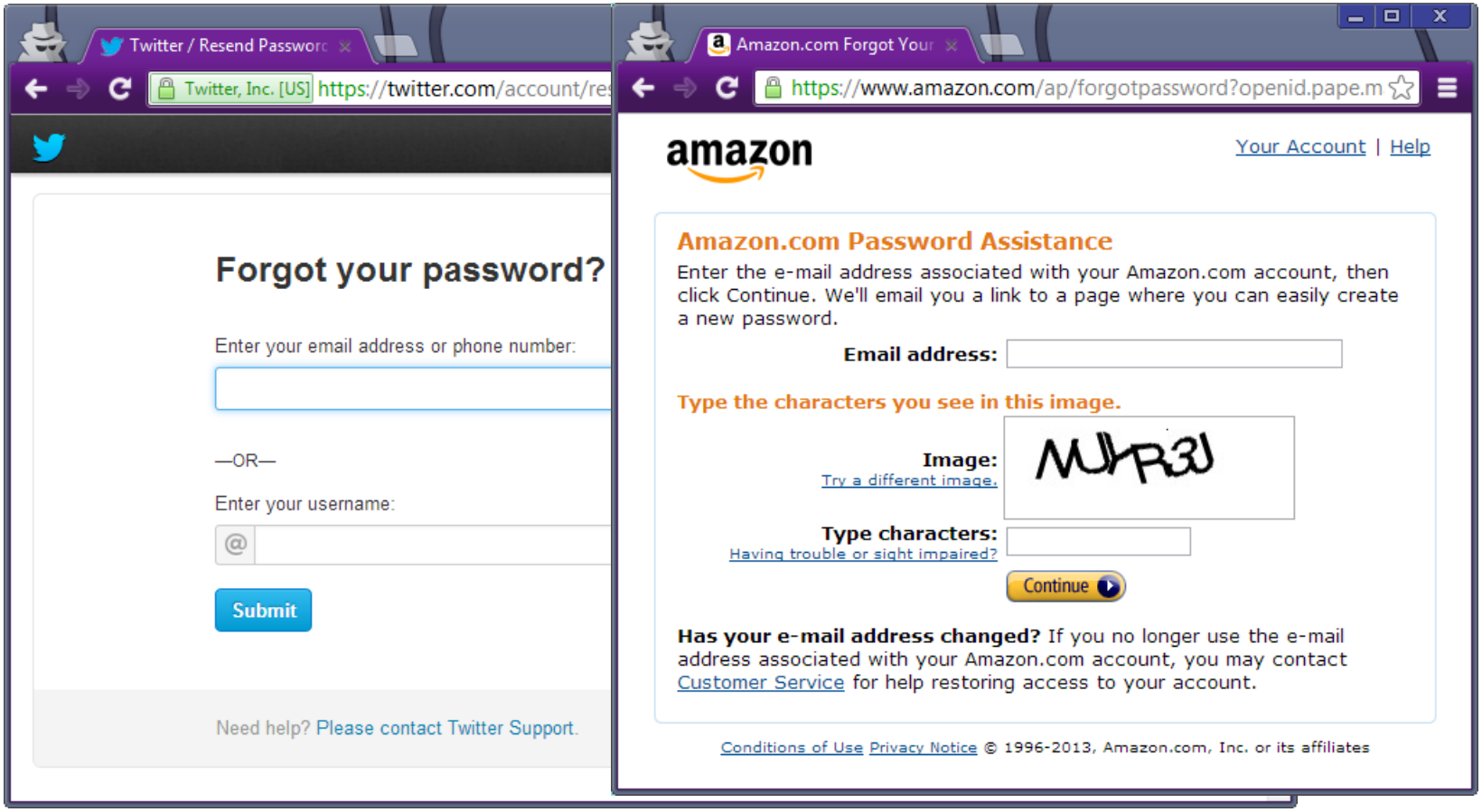
- Two-Factor Authentication
  - How many tokens/app can a user handle?

- Single sign-on services
  - Single point of failure
  - Relying party gets to find out user identity*
  - Privacy issues from coarse-grained data sharing



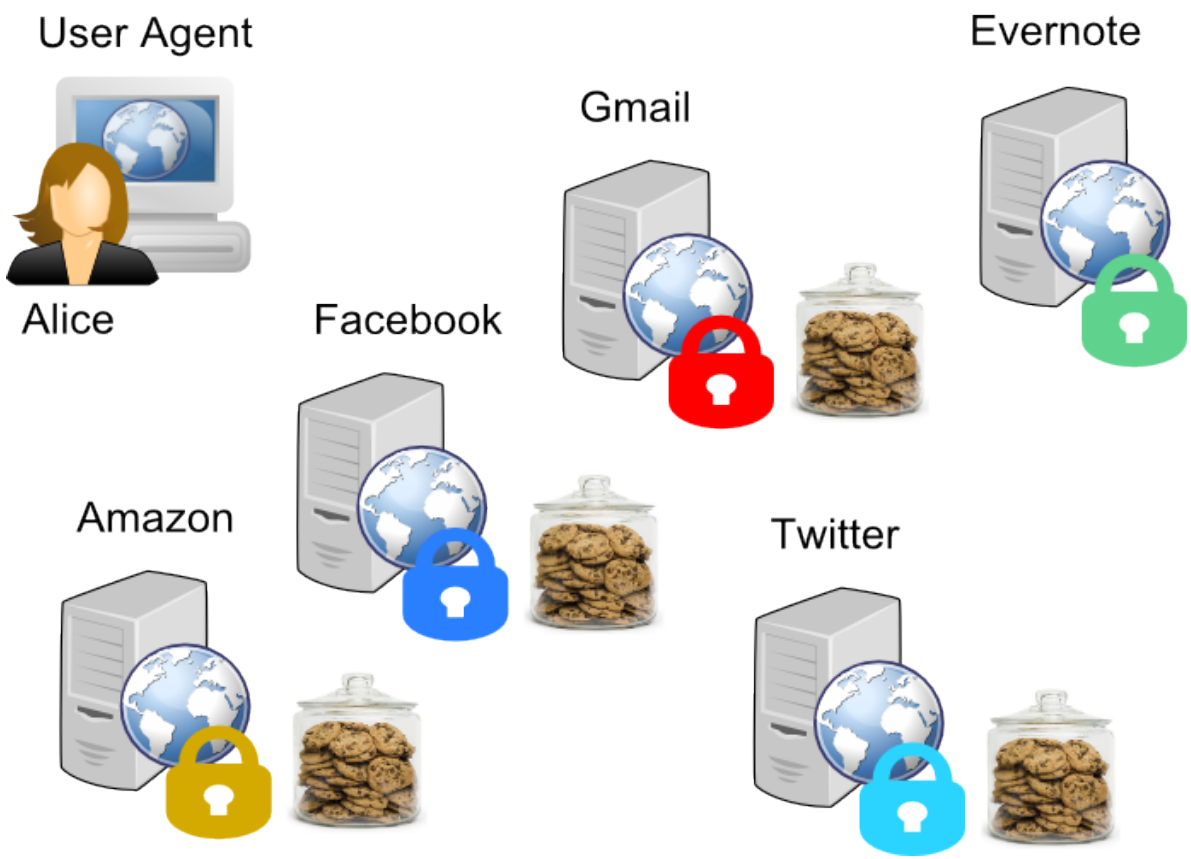
How about Authentication Synergy?

- Forgot your password?



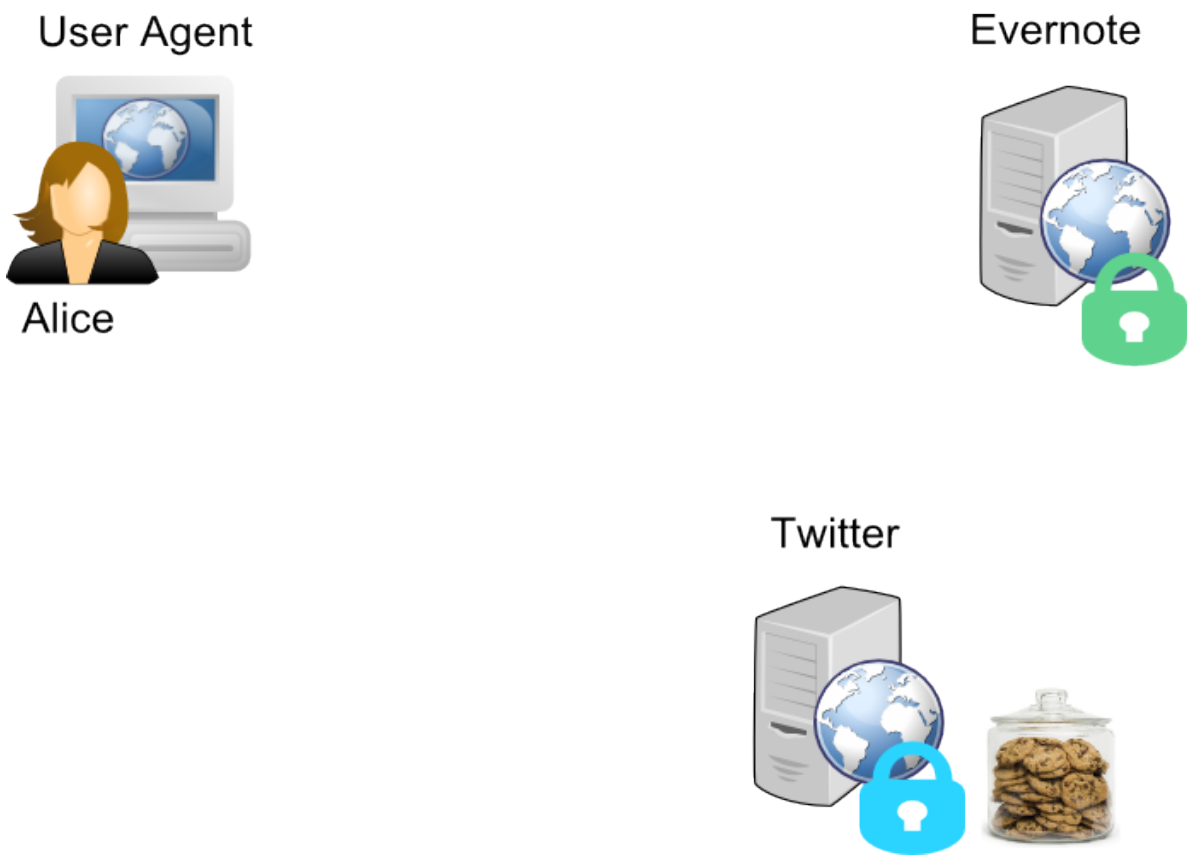
How about Authentication Synergy?

- User's Authentication State



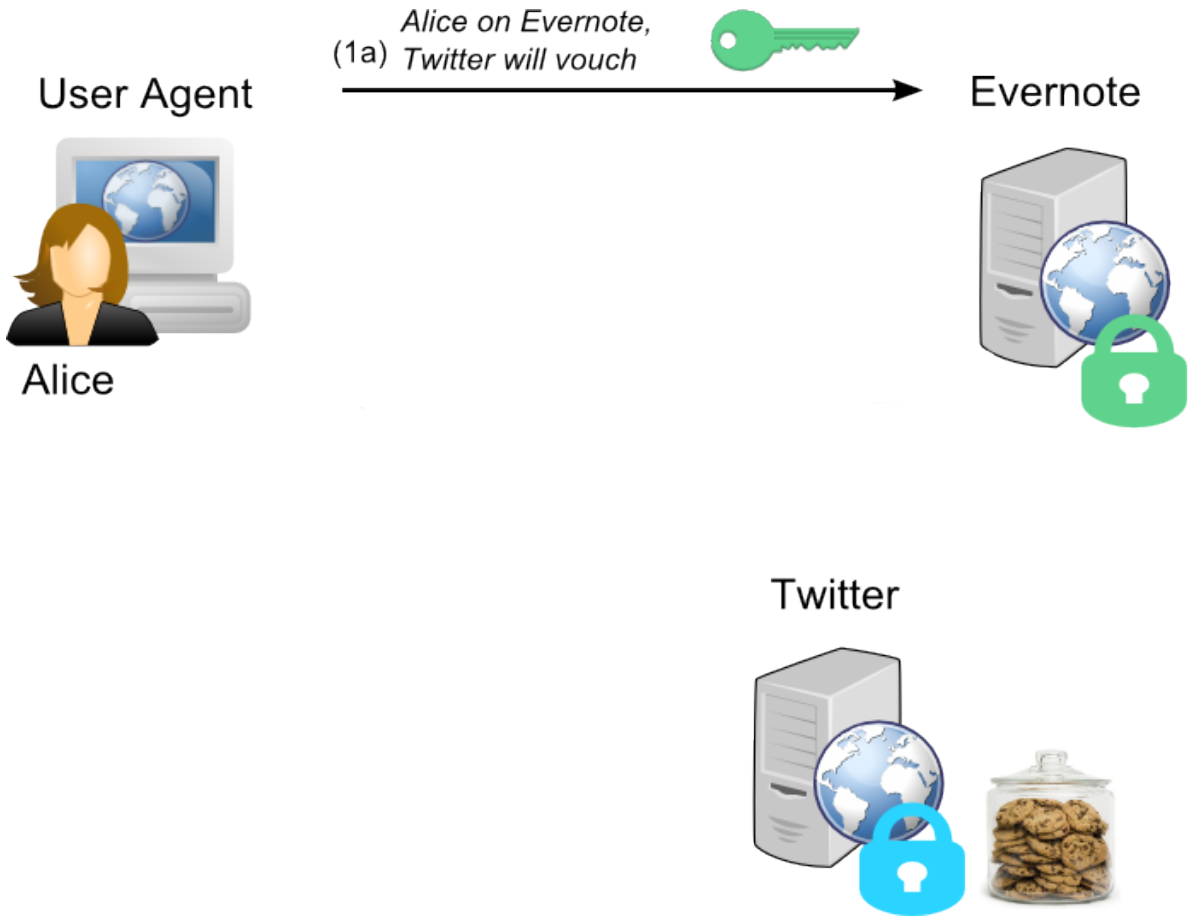
SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



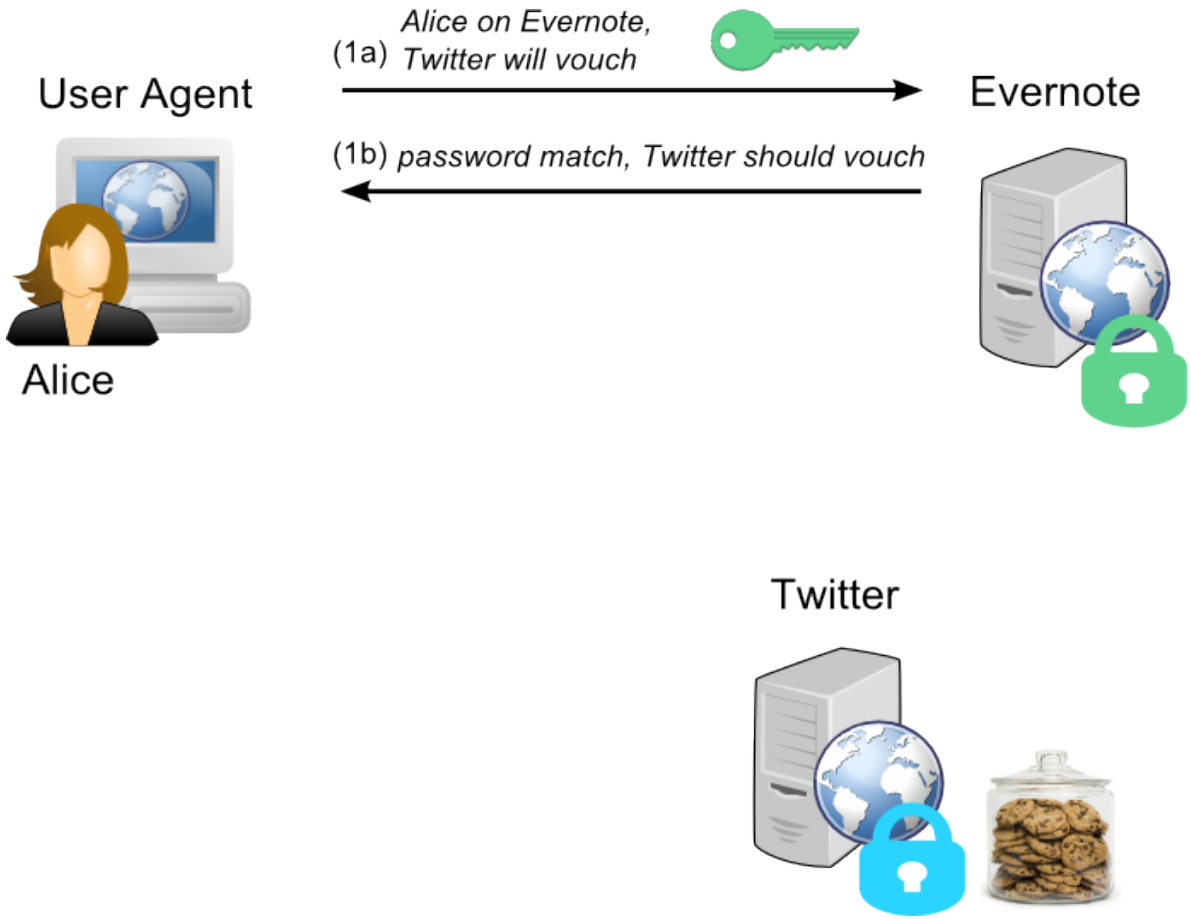
SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



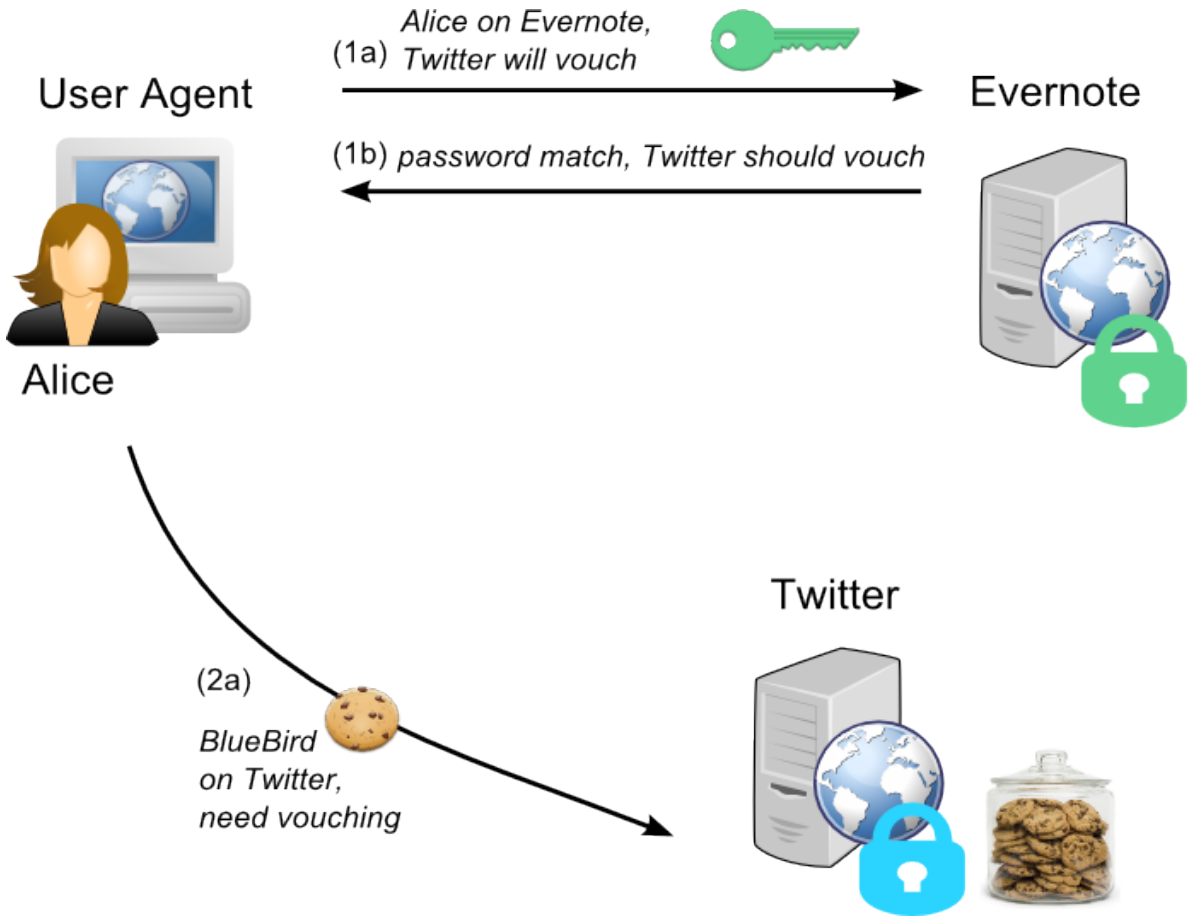
SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



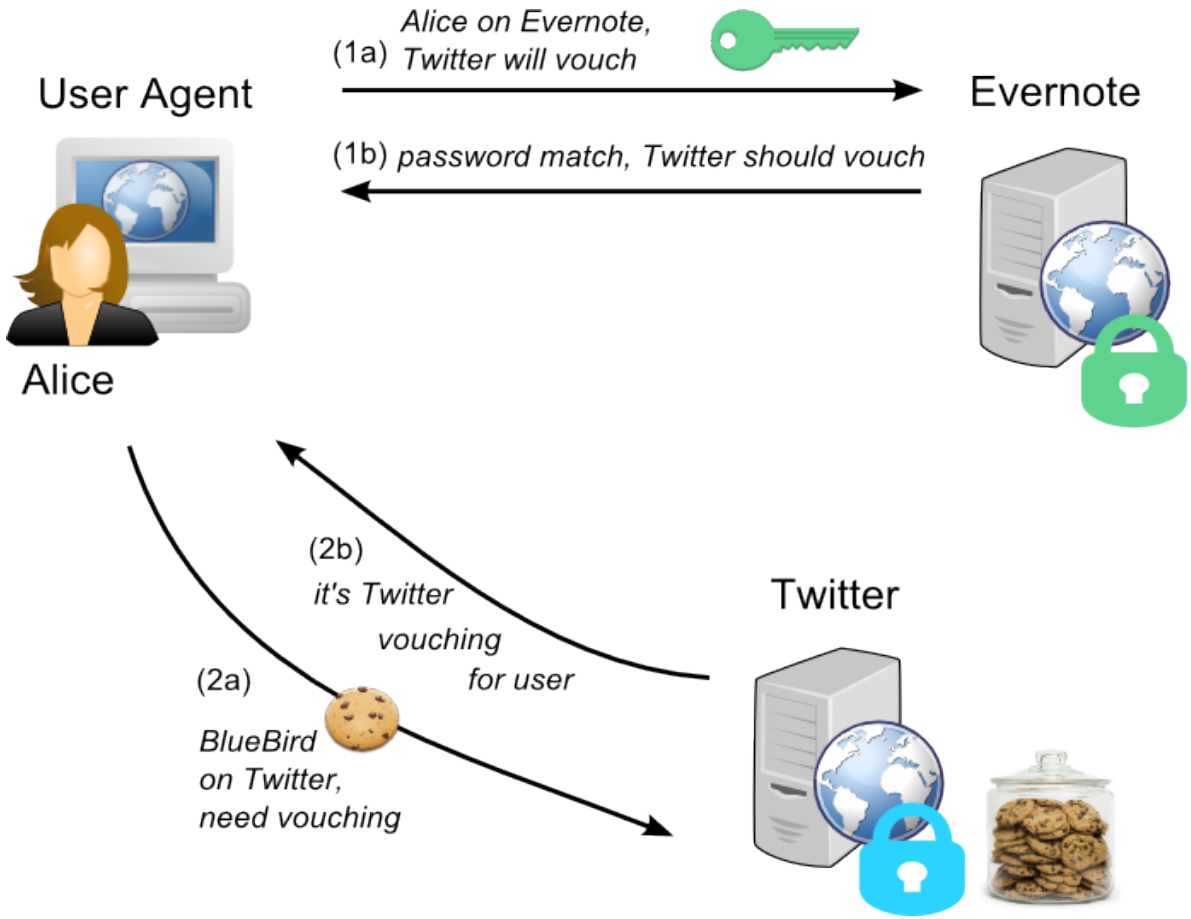
SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



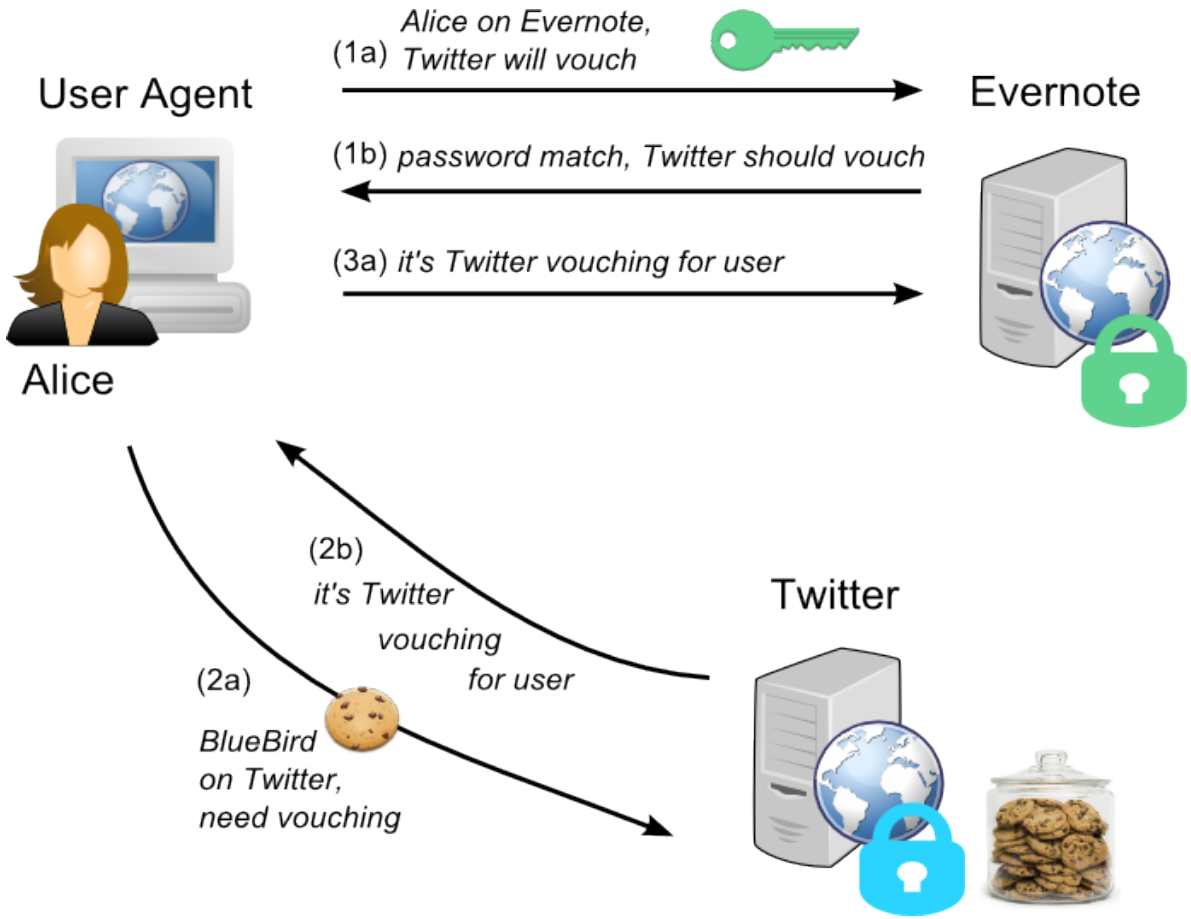
SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



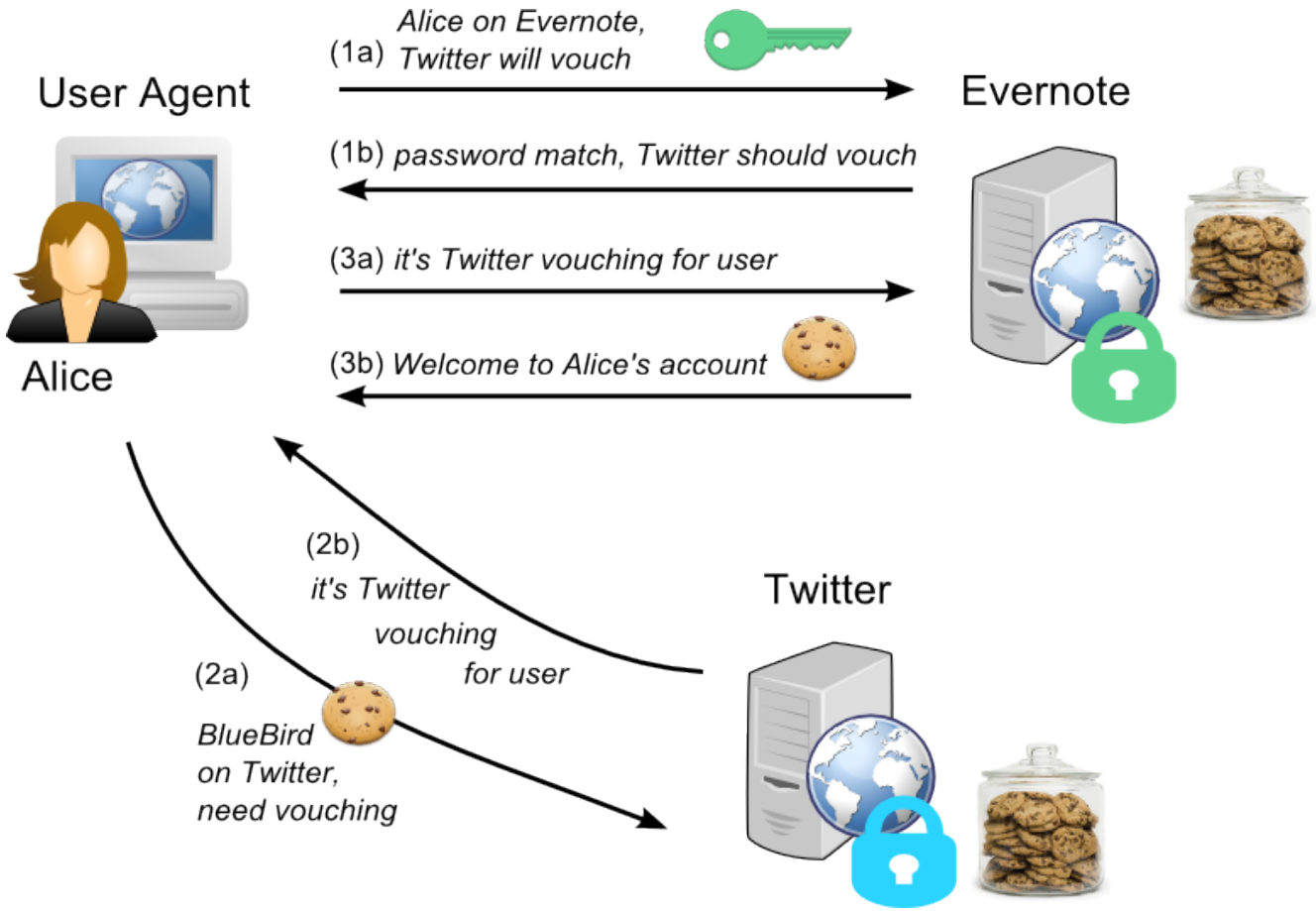
SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



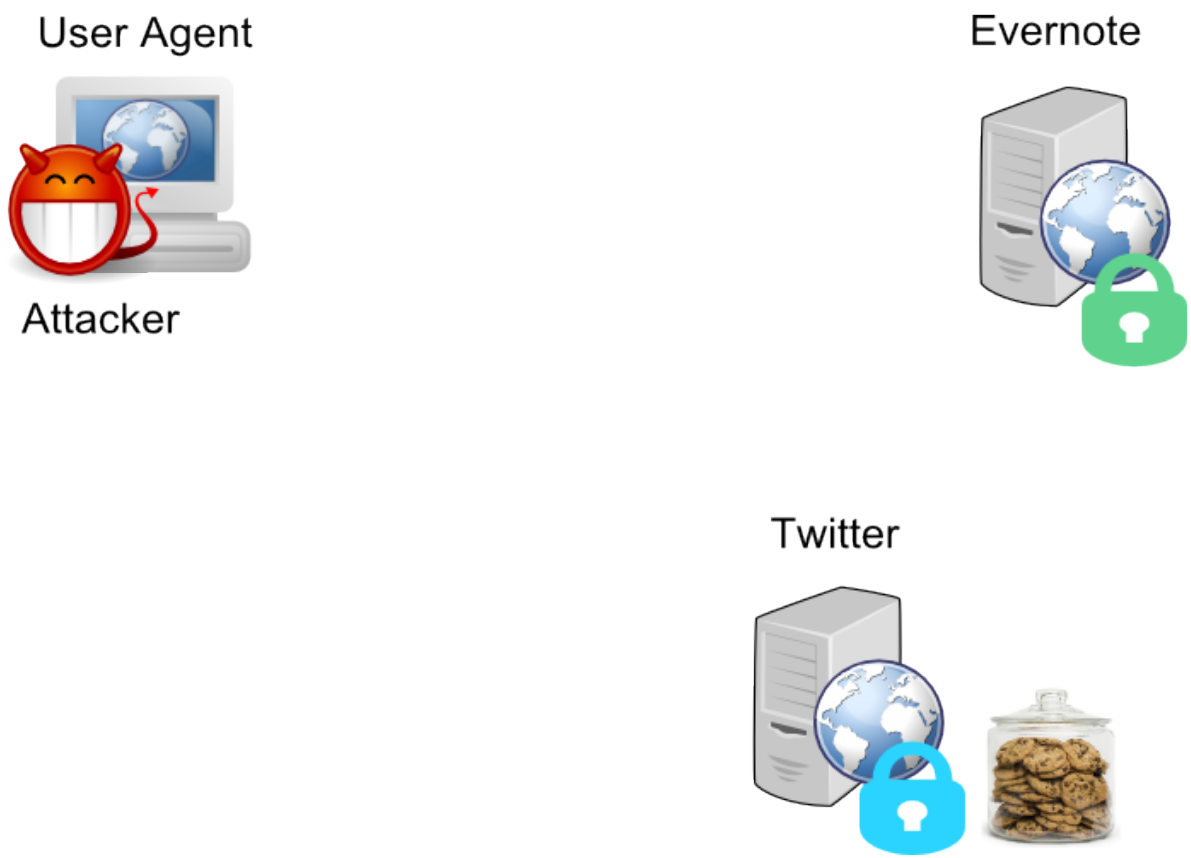
SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



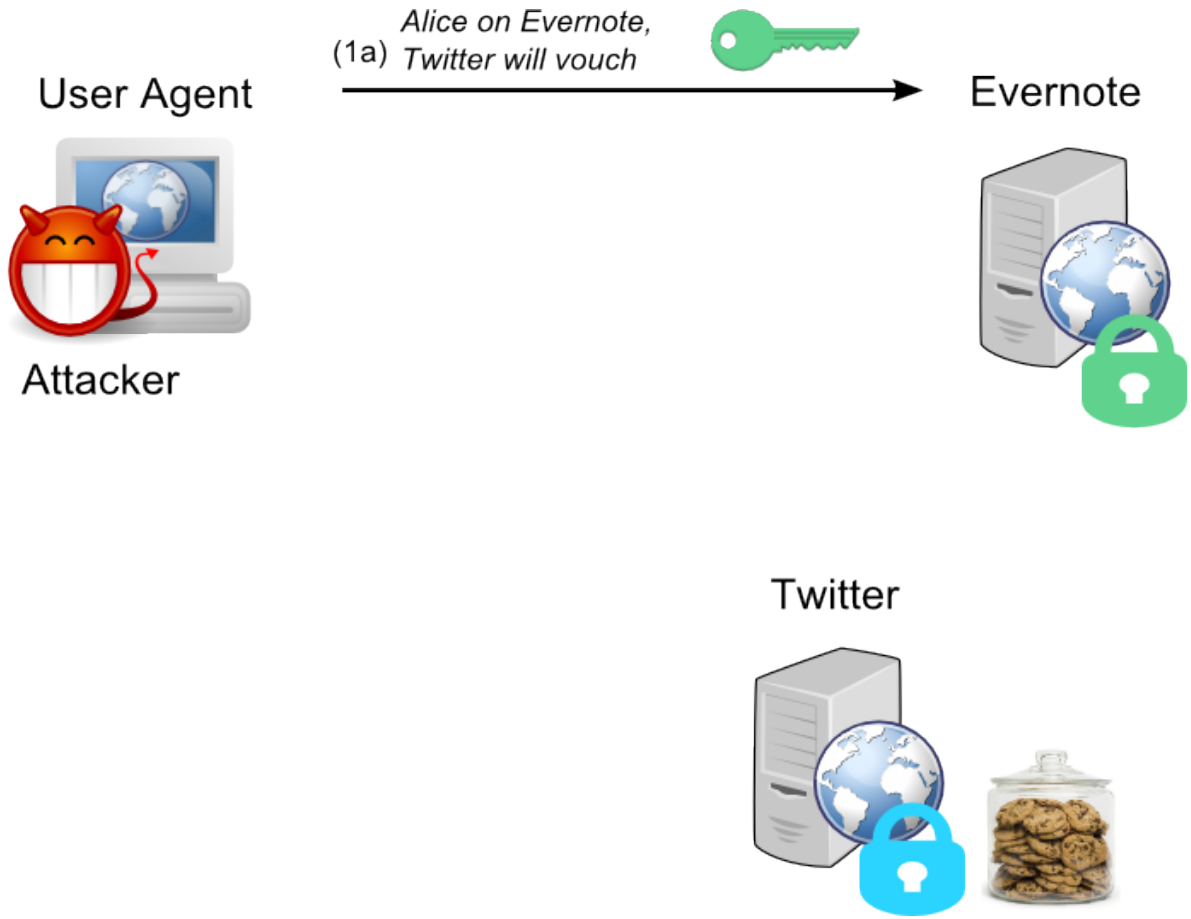
SAuth: Synergy-based Enhanced Authentication

- Password leak on Evernote will protect account access



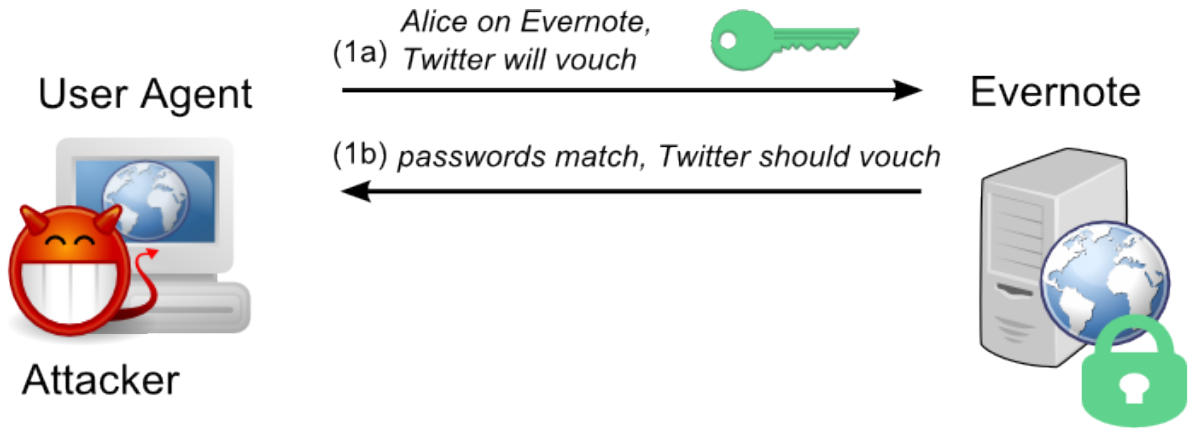
SAuth: Synergy-based Enhanced Authentication

- Attacker has compromised Alice's password on Evernote



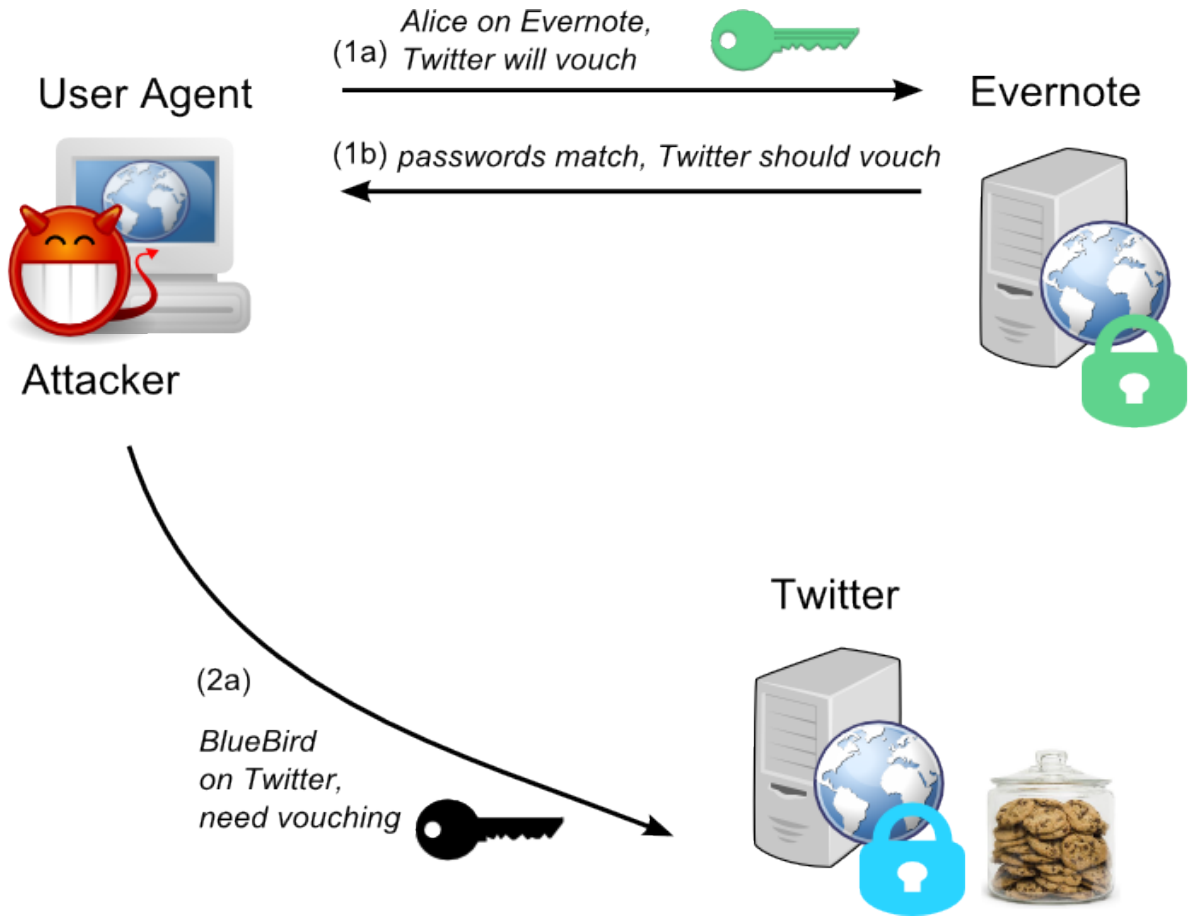
SAuth: Synergy-based Enhanced Authentication

- Attacker impersonates Alice on Evernote



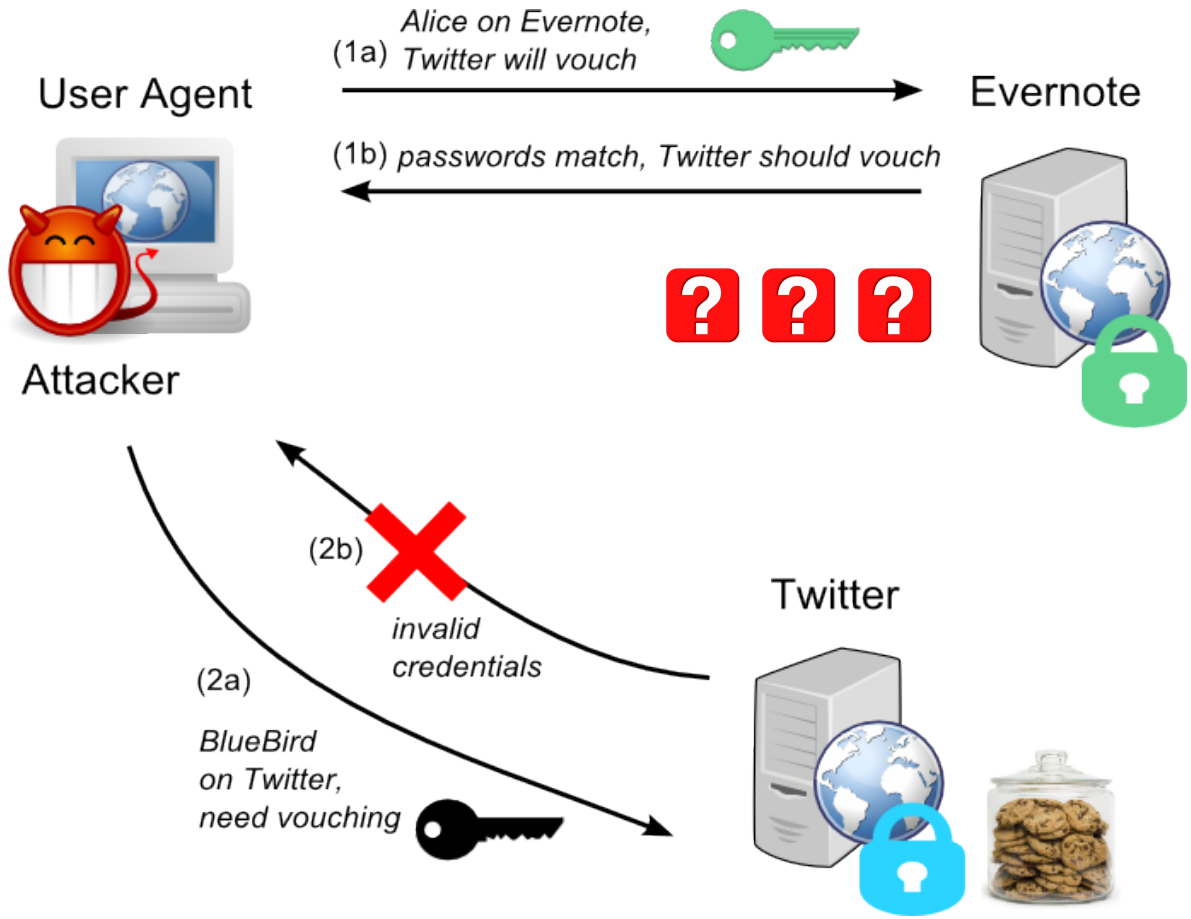
SAuth: Synergy-based Enhanced Authentication

- Attacker is unable to produce Alice's Twitter password



SAuth: Synergy-based Enhanced Authentication

- Authentication process fails, Evernote denies access



Password Reuse Woes

Stolen passwords re-used to attack Best Buy accounts

Summary: *Customer re-use of the same user name and password across multiple sites is being blamed for attacks on customer accounts at BestBuy.com.*

ZDNet

- User has 7 passwords, re-uses 5 of them
- Password shared across 6 sites [Florencio WWW ’07]

Decoy Passwords

- Uncertainty about the actual password
- Store N-1 decoy passwords along
- Attack reduced to online guessing
- All decoys are valid passwords,
  server does not know the difference

Username	*P[0]*	*P[1]*	*P[…]*	*P[N]*

- How many decoys?
  - *16,384 for NIST L2 security when password is reused*

Realistic Decoy Passwords

- User password must blend-in with the decoys
  - Crackers are already factoring in human behavior
  - Complex vs Popular Passwords

string-digit	*37%*
digit-string	*05%*
!	*10%*
$	*03%*

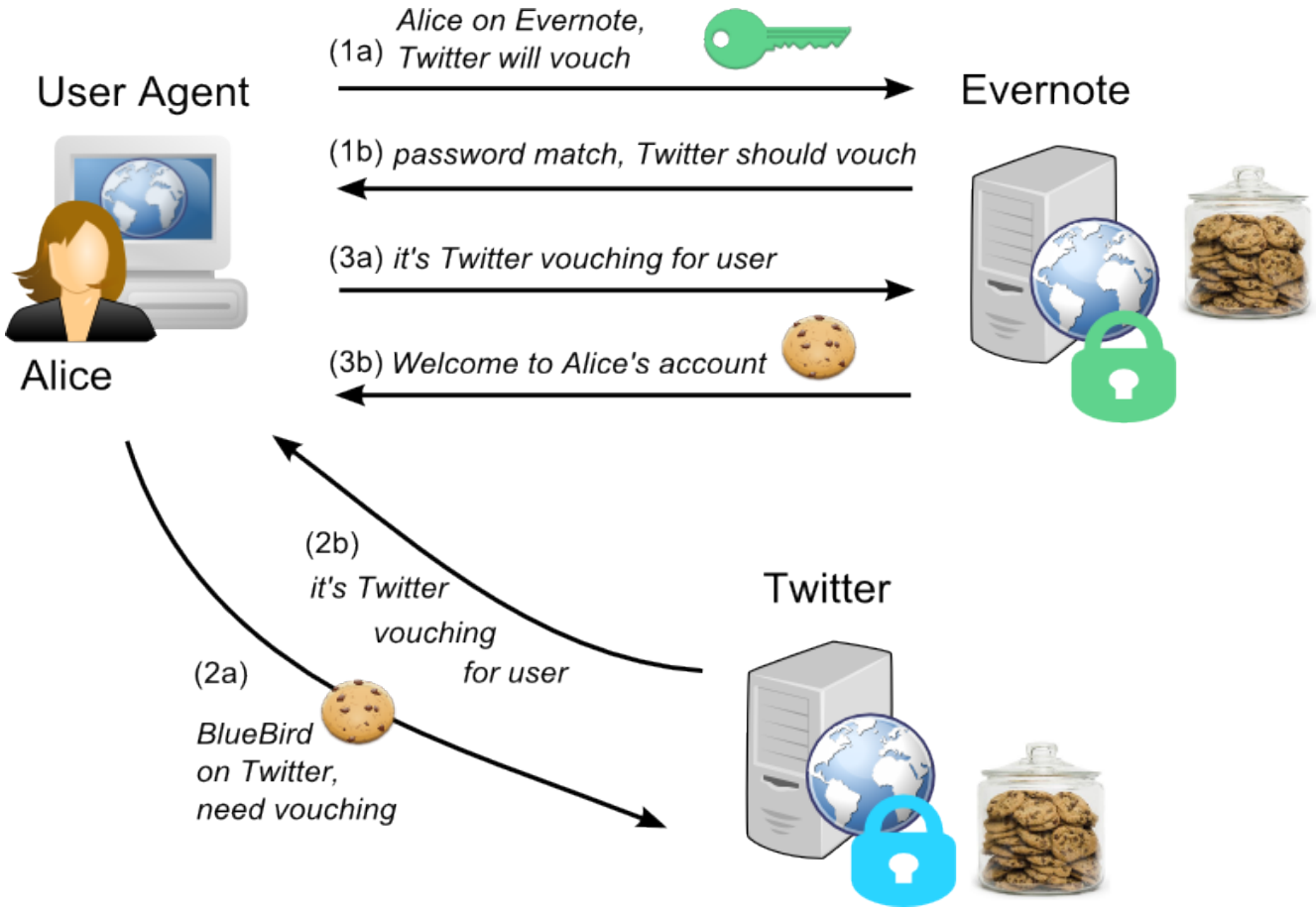
- RockYou Leak '09

  - Ideal: have the user type N passwords, remember 1
  - Practical: generation within the password ecosystem
    - Any blind automated method will generate outliers
    - Probabilistic production seeded by user's password, biased towards structures of similar popularity and semantics

Summary

- Authentication Synergy results in leak-resistant password authentication
  - Complements existing security
  - Respect for user privacy, verifiable site cooperation
  - Minimal changes server-side, no changes client-side
- Decoys mitigate password reuse habits
  - Generated off the user password, consider its context and general human password habits

tinyurl.com/sauth



kontaxis@cs.columbia.edu

Intentionally left blank

Intentionally left blank

Unintentionally left blank

Honeywords, Kamouflage and SAuth Decoy Passwords

- Honeywords
  - Does not yet consider human password habits
  - Honeywords are not valid passwords
  - Use of any honeyword will raise an alarm
  - Auxiliary honeychecking server
- Kamouflage password manager
  - Considers human password habits
  - Master password decoys are all valid
  - Online guessing attack should raise alarm
- SAuth Decoy Passwords
  - Considers human password habits
  - Decoy passwords are all valid
  - Online guessing attack should raise alarm