Cleversafe®

Authenticating Cloud Storage with Distributed Credentials

Library of Congress – September 27, 2011

The Mission of Cloud Storage

Cleversafe

Cloud storage presents unique challenges:

- Users expect flexible access from any location
- Many nodes are involved in storing the data
- The system must be able to scale indefinitely
  - Requires decentralization of critical services
  - Decentralization eliminates single points of failure

Challenge: How can we make the *authentication system* reliable without sacrificing security?

Recent cases in the news…

2010, the blog network Gawker was compromised, exposing the passwords of **1.3 million** users

2011, hosting site SourceForge was attacked, affecting the security of over **2 million** user accounts

2011, **10 million** users of the mobile application Trapsters' e-mail address and password compromised

The Solution: Distributed Credentials

Enable end-users to recover a private key from any location on the network

- Bridges the gap between password authentication and PKI authentication
  - Appears like password authentication to end users
  - Appears like PKI authentication to service providers

Nothing enabling an offline attack exists at any location

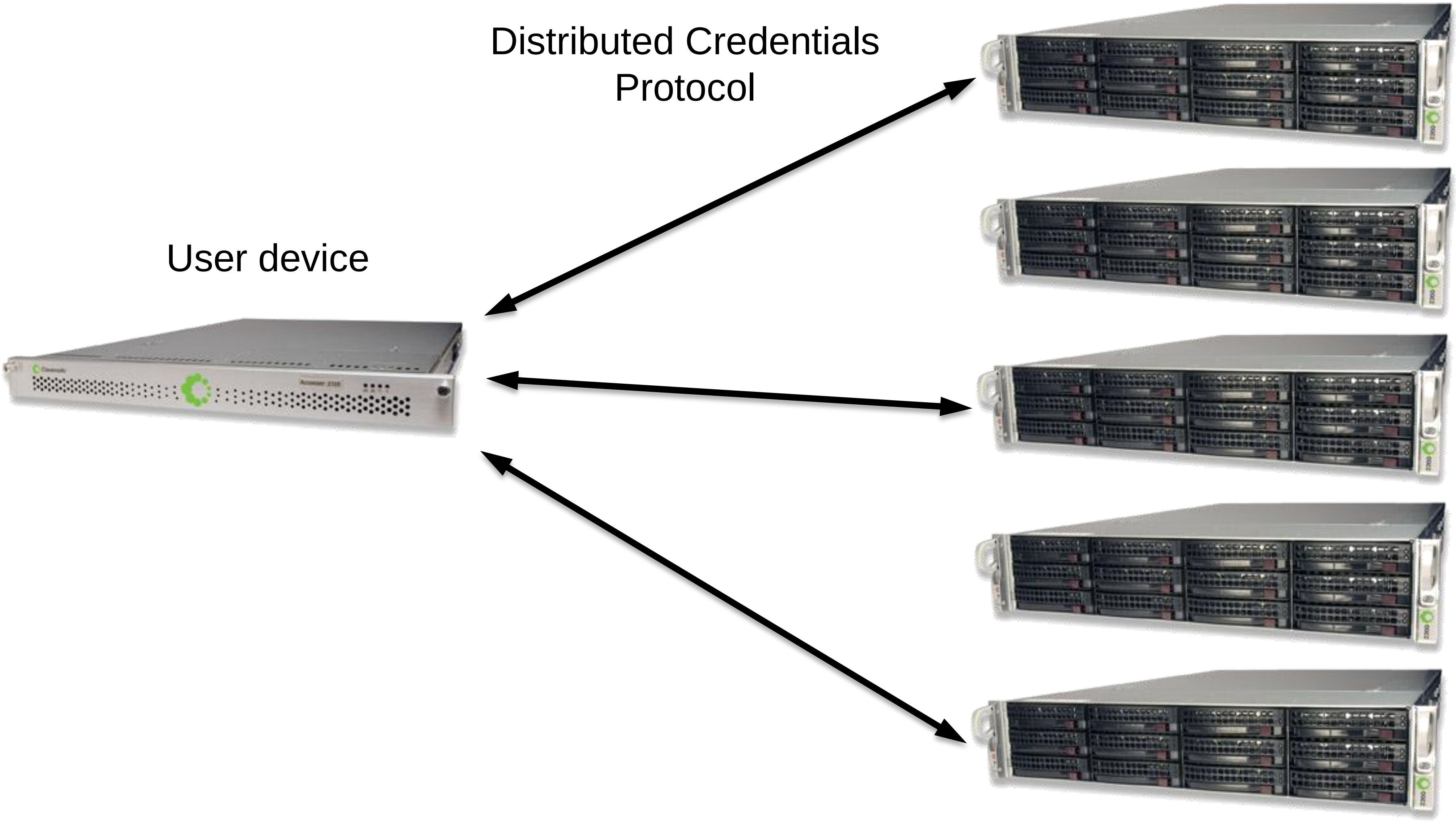
- Breach of authentication server yields nothing!

Distributed Credentials Architecture

User device

username:	jsmith01
password:	********

Distributed Credentials Architecture

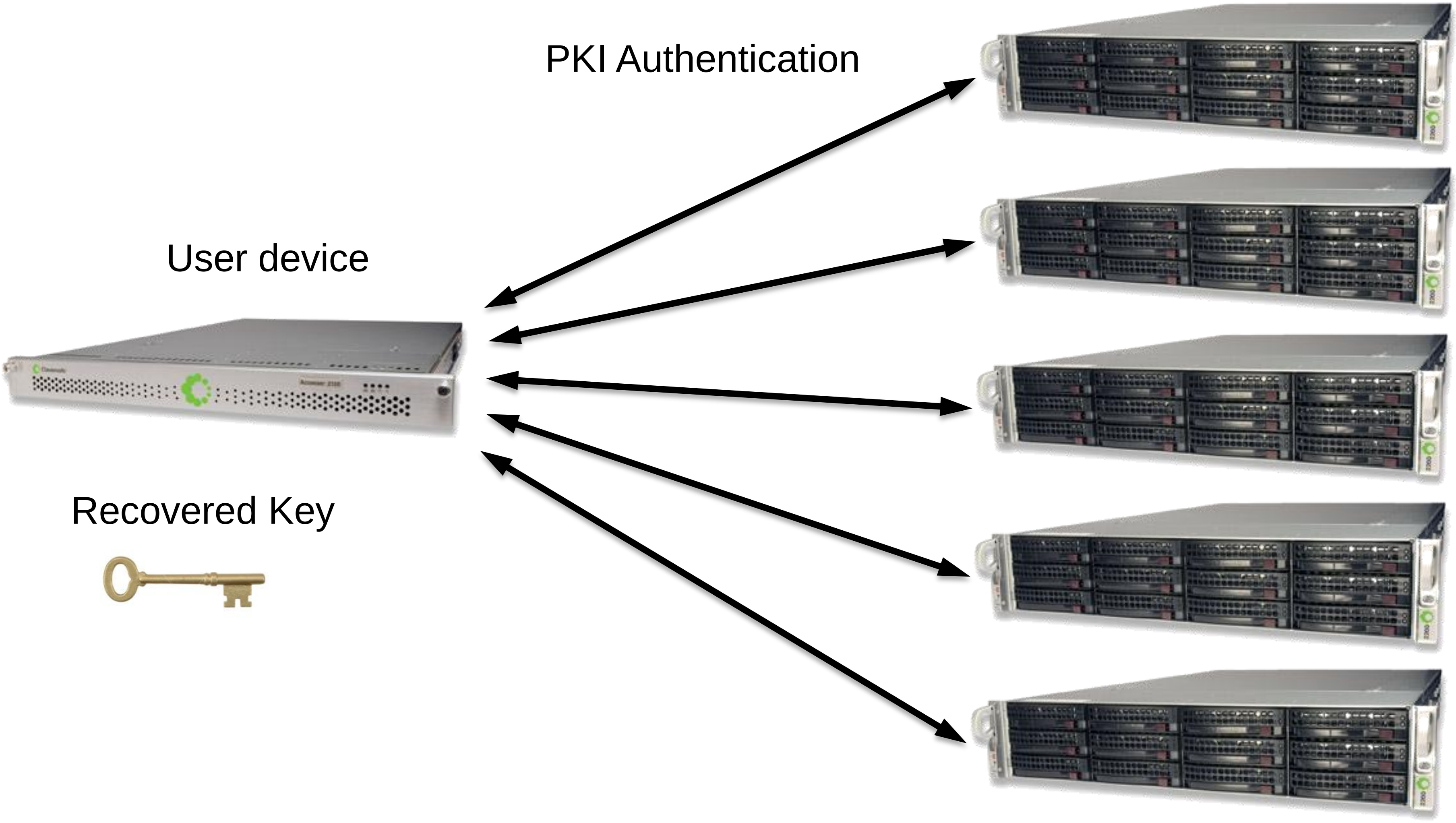


Distributed Credentials Architecture

User device

Recovered Key

Distributed Credentials Architecture



Distributed Credentials Architecture

User device

Recovered Key

Comparison of Mechanisms

	Password	PKI	DK
1. No single point of failure	✗	✓	✓
2. No single point of compromise	✗	✗	✓
3. Enables access from any location	✓	✗	✓
4. Easy to use	✓	✗	✓
5. Immune to offline brute-force attacks	✗	✗	✓ *
6. Credentials are not disclosed to use	✗	✓	✓
7. Immune to physical theft	✓	✗	✓

***** Requires a threshold number of simultaneous compromises

Questions

Backup

Authenticating to the Cloud

Cleversafe

Implementers of cloud storage are forced to choose between several sub-optimal authentication systems:

- A system whose security is inversely proportional to the number of nodes in the cloud
- A system with poor availability and scalability
- A system that is inconvenient and hard to use

At my company, we were faced with this dilemma:

- How can we make the authentication system reliable without sacrificing security?

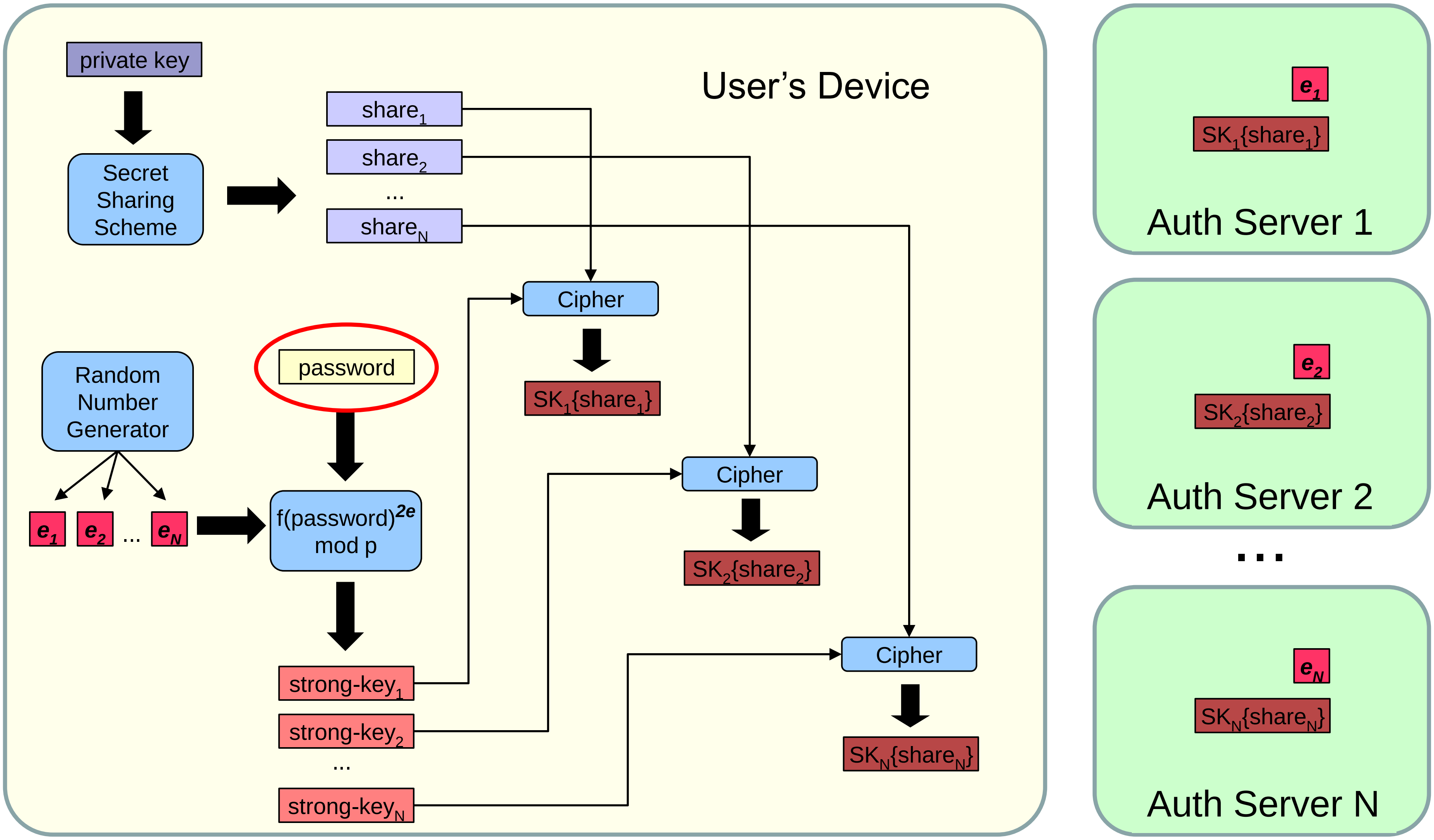
How it Works

We found that through a combination of various cryptographic protocols, an authentication system with almost ideal properties could be formed

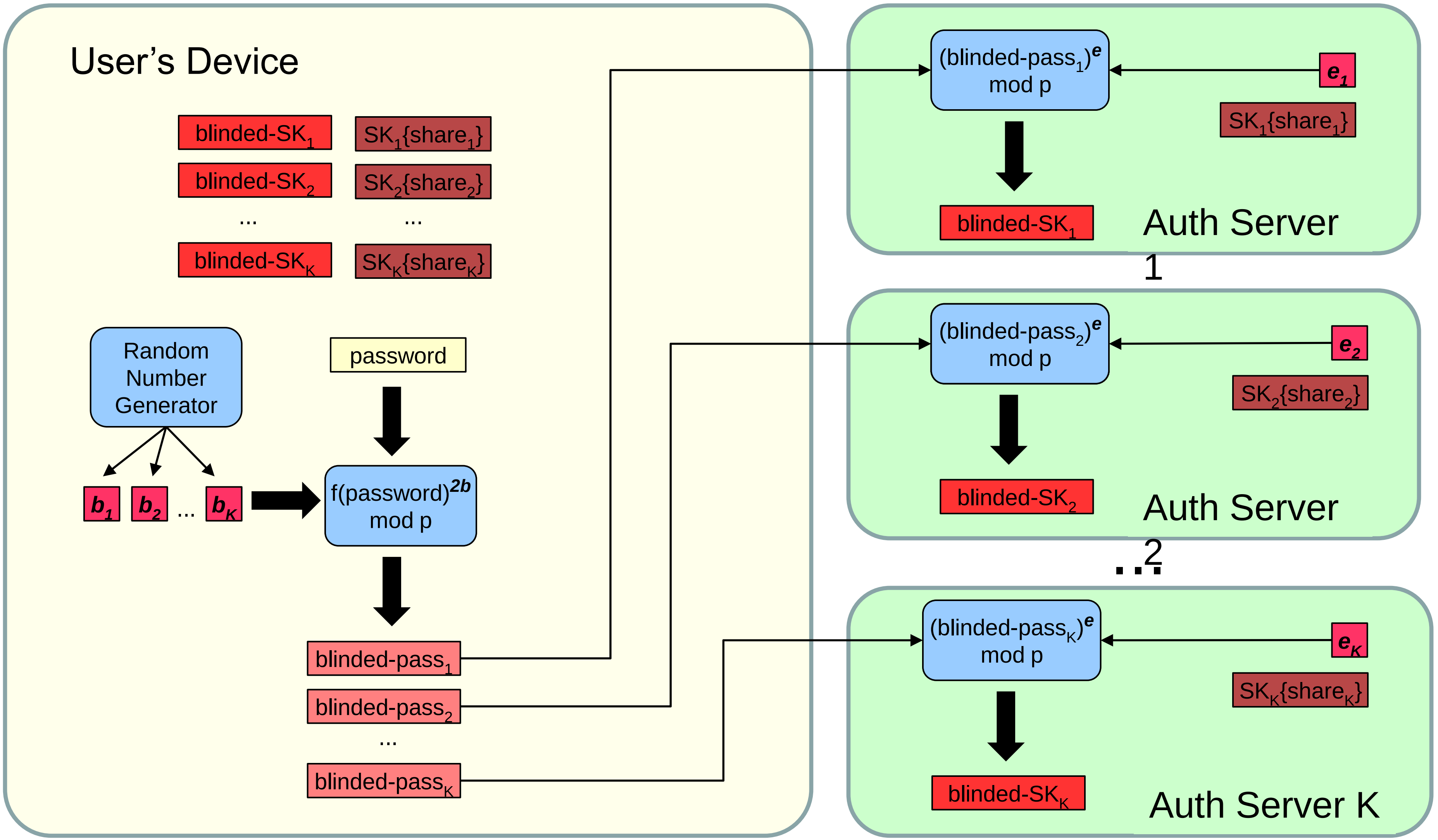
- Server-assisted strong secret generation
  - Warwick Ford and Burton S. Kaliski Jr. (2000)
- Secret Sharing
  - Adi Shamir and George Blakley (1979)
- Encryption and Digital Signatures

14

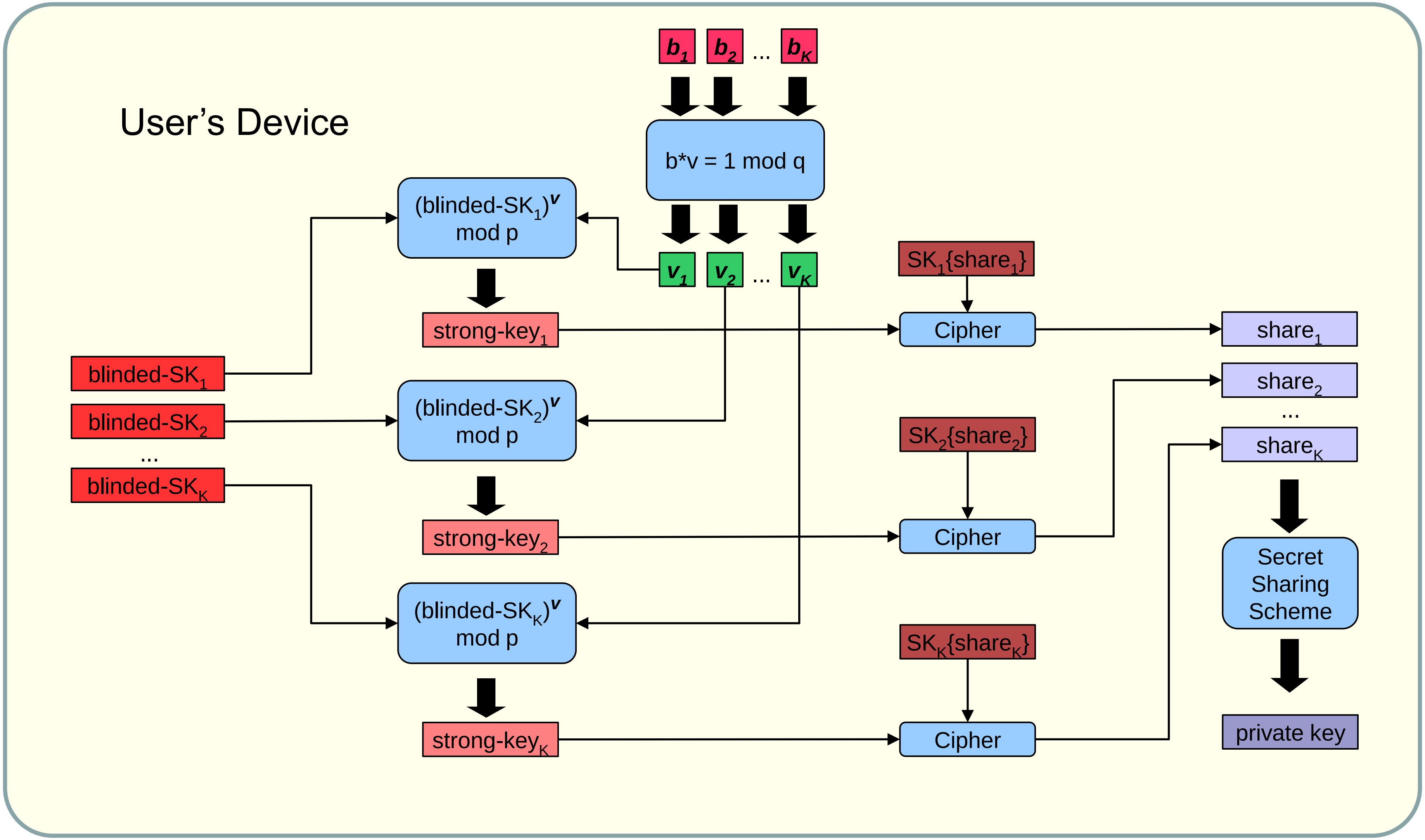
Distributed Key Storage



Distributed Key Retrieval (1 of 2)



Distributed Key Retrieval (2 of 2)



Why the math works

$$p = 2q + 1$$

P and Q - two large primes defined in the system

$$x = f(password)$$

Represent the f(password) with the number x

$$strongkey = x^{2e} \pmod p$$

Strong key is password to the power 2e

$$((x^{2b})^e)^v \equiv x^{2e} \pmod p$$

This is what will be proved…

$$bv \equiv 1 \pmod q \Rightarrow bv = nq + 1$$

Implies (bv)/q = n remainder 1, for some integer *n*.

$$((x^{2b})^e)^v = x^{2bev} = x^{2e(nq+1)} = x^{2enq+2e}$$

Substitute (bv) with (nq+1)

$$x^{2enq+2e} = (x^{2q})^{en} \cdot x^{2e}$$

Isolate the strong key

$$(x^{p-1})^{en} \cdot x^{2e}$$

Replace 2q with (p-1), since p = 2q+1

$$1^{en} \cdot x^{2e} \pmod p$$

By Fermat's little theorem: $a^{(p-1)} = 1 \pmod p$

$$x^{2e} \pmod p = strongkey$$

1 raised to any power is 1, this is the strong-key

References

[1] Estimating password strength

- NIST Special Publication 800-63, Version 1.0.2

[2] How to Share a Secret

- Adi Shamir, In Communications of the ACM 22 (11): 612–613, 1979.

[3] Server-Assisted Generation of a Strong Secret from a Password

- Warwick Ford and Burton S. Kaliski Jr. In Proc. IEEE 9th Int. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pages 176-180. IEEE Press, 2000.

[4] Compromise of 10 million user passwords from Trapster:

- http://blogs.computerworld.com/17690/over_10_million_passwords_possibly_compromised_at_trapster

[5] Compromise of 2 million user passwords from SourceForge:

- http://thenextweb.com/industry/2011/01/29/sourceforge-attacked-resets-2-million-account-passwords-to-protect-users/

[6] Vulnerability of Kerberos to offline dictionary attacks (RFC 1510, section 1.2):

- http://www.ietf.org/rfc/rfc1510.txt

[7] Compromise of 1.3 million user passwords from Gawker:

- http://gadgetwise.blogs.nytimes.com/2010/12/13/gawker-passwords-hacked-what-you-should-do/