The Challenges of Distributing Distributed Cryptography

Ari Juels

Chief Scientist, RSA

RSA DISTRIBUTED CREDENTIAL PROTECTION

SCRAMBLE, RANDOMIZE, AND SPLIT CREDENTIALS

WATCH VIDEO

New RSA Innovation Helps Thwart "Smash-and-Grab" Credential Theft »

RSA Distributed Credential Protection: Protect passwords, credentials, and secrets by scrambling, randomizing, and splitting across two servers, addressing primary points of server compromise.

SCRAMBLE, RANDOMIZE, SPLIT

Don't be the next news headline. Scramble, randomize, and split your secrets and credentials into two locations. Make it too much effort for an attacker to breach your password stores.

RE-RANDOMIZE SECRETS

Re-randomization of secrets and credentials can happen proactively on an automatic schedule or reactively, making information taken from one server useless in the event of a detected breach.

USER TRANSPARENCY

With re-randomization, the protection on the secrets and credentials changes, but the actual secrets and credentials don't. Users continue to use their known passwords without any extra hassle.

SECRETS NEVER REASSEMBLED

Authenticate without actually recombining the secrets and credentials, eliminating a point of potential compromise. Compare the secrets and credentials over a secure channel.

What is this new and mysterious technology?

- Hint: It's 20+ years old.
  - R. Ostrovsky and M. Yung. How to withstand mobile virus attack. PODC, 1991.
  - O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. STOC, 1987.
- Answer: Distributed (+proactive) cryptography
- The specific implementation is 8+ years old.
  - J. Brainard, A. Juels, B. Kaliski, and M. Szydlo. A New Two-Server Approach for Authentication with Short Secrets. USENIX Security, 2003.
  - M. Szydlo and B. Kaliski, Proofs for Two-Server Password Authentication, CT-RSA, 2005.
- But it's the first broadly available commercial implementation of distributed cryptography!
- Why is it taking so long?

New technologies need to await a happy convergence of factors

THE WORLD: THURSDAY, JANUARY 2, 1902

WIRELESS TELEPHONE TEST IS A SUCCESS.

Public Exhibition by the Inventor in the Streets of Murray, Kentucky.

(Special to The World.)

MURRAY, Ky., Jan. 1.—Nathan Stubblefield, a local electrician and inventor, who claims to have solved the problem of wireless telephony, gave a public exhibition of his apparatus in the main street to-day. Citizens stood at the instruments and communicated with one another with perfect ease at a distance of six blocks.

The apparatus is simple. Wire is buried

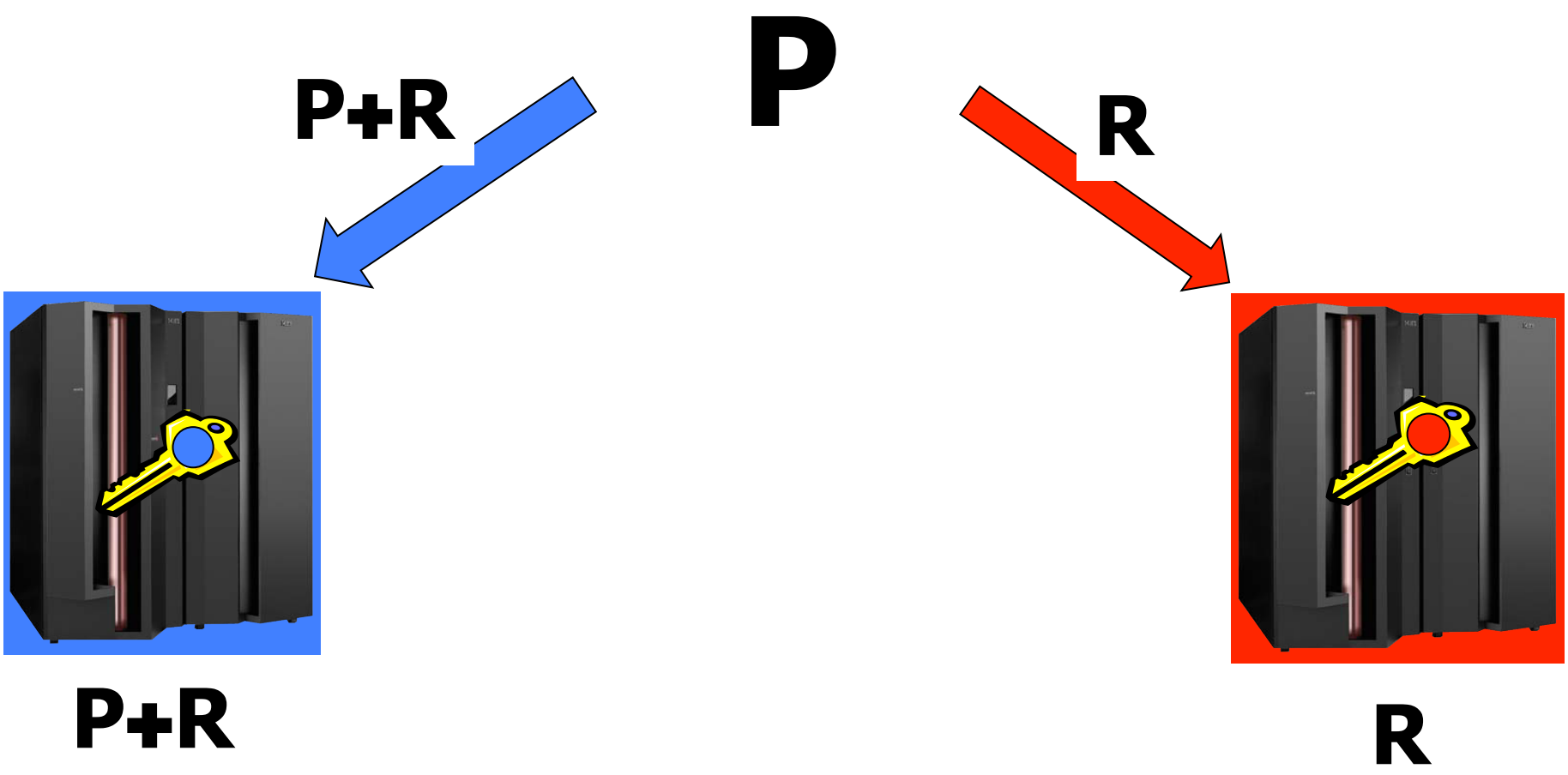
How DCP works

P

How DCP works

P

How DCP works



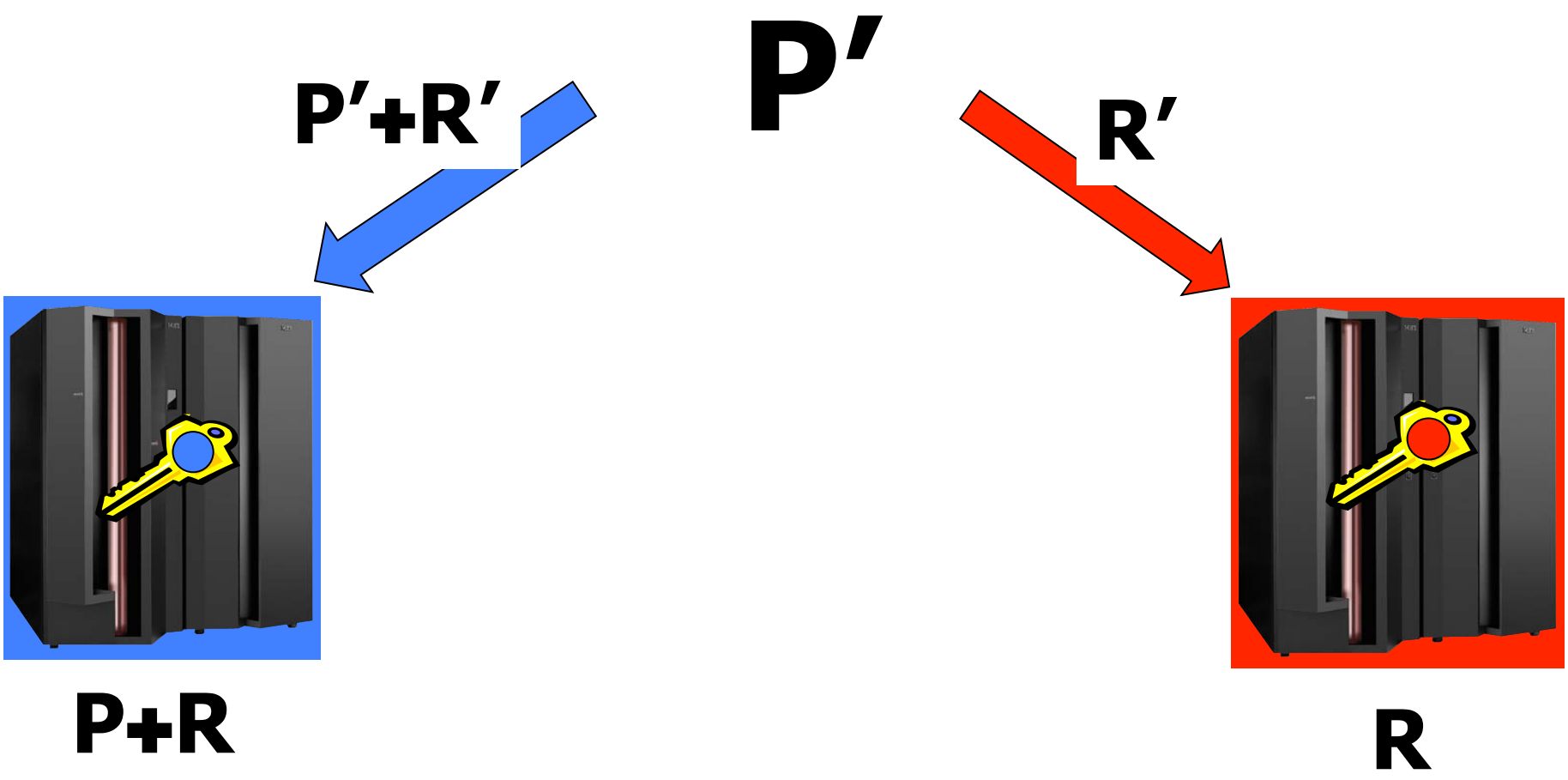
How DCP works

P′

P+R

R

How DCP works



How DCP works

P+R –

(P′+R′)

R –

R′

How DCP works

If P = P′

then…

~~P~~+R –

(~~P~~′+R′)

=

R –

R′

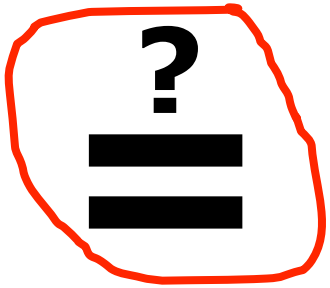
How DCP works

So core function is

Private Equality Test

P+R –

(P′+R′)



R –

R′

What distributed crypto achieves

Cryptographer's view

- Much stronger security
  - Mobile, active adversary
- Extensible to protection of variety of resources
  - E.g., keys, biometrics,…
- Extensible to *k*-out-of-*n*

Enterprise view

- Twice the equipment cost
- Diminished availability (or even higher cost)
  - "Five nines" ➔ "Four nines"
- Not solving real problem: phishing and social engineering, i.e., endpoint
- Breaching two systems as easy as breaching one
- Two distinct systems to maintain
- No obvious return on investment

What distributed crypto achieves

Enterprise view (new)

- Equipment is (fairly) cheap

Enterprise view (old)

- Twice the equipment cost
- Diminished availability (or even higher cost)
  - "Five nines" ➔ "Four nines"
- Not solving real problem: phishing and social engineering, i.e., endpoint
- Breaching two systems as easy as breaching one
- Two distinct systems to maintain
- No obvious return on investment

What distributed crypto achieves

Enterprise view (new)

- Equipment is (fairly) cheap
- At worst, the last '9' becomes an '8' (easy to bump back up)
  - E.g., 99.999% ➔ 99.998%

Enterprise view (old)

- Twice the equipment cost
- Diminished availability (or even higher cost)
  - "Five nines" ➔ "Four nines"
- Not solving real problem: phishing and social engineering, i.e., endpoint
- Breaching two systems as easy as breaching one
- Two distinct systems to maintain
- No obvious return on investment

What distributed crypto achieves

Enterprise view (new)

- Equipment is (fairly) cheap
- At worst, the last '9' becomes an '8' (easy to bump back up)
  - E.g., 99.999% ➔ 99.998%
- (1) Layered security; (2) Enterprises liable for breaches, not phishing

Enterprise view (old)

- Twice the equipment cost
- Diminished availability (or even higher cost)
  - "Five nines" ➔ "Four nines"
- Not solving real problem: phishing and social engineering, i.e., endpoint
- Breaching two systems as easy as breaching one
- Two distinct systems to maintain
- No obvious return on investment

What distributed crypto achieves

Enterprise view (new)

- Equipment is (fairly) cheap
- At worst, the last '9' becomes an '8' (easy to bump back up)
  - E.g., 99.999% ➔ 99.998%
- (1) Layered security; (2) Enterprises liable for breaches, not phishing
- Diversification, e.g., virtualize with distinct OSs

Enterprise view (old)

- Twice the equipment cost
- Diminished availability (or even higher cost)
  - "Five nines" ➔ "Four nines"
- Not solving real problem: phishing and social engineering, i.e., endpoint
- Breaching two systems as easy as breaching one
- Two distinct systems to maintain
- No obvious return on investment

What distributed crypto achieves

Enterprise view (new)

- Equipment is (fairly) cheap
- At worst, the last '9' becomes an '8' (easy to bump back up)
  - E.g., 99.999% ➔ 99.998%
- (1) Layered security; (2) Enterprises liable for breaches, not phishing
- Diversification, e.g., virtualize with distinct OSs
- Put one server in the cloud
  - Good for security too

Enterprise view (old)

- Twice the equipment cost
- Diminished availability (or even higher cost)
  - "Five nines" ➔ "Four nines"
- Not solving real problem: phishing and social engineering, i.e., endpoint
- Breaching two systems as easy as breaching one
- Two distinct systems to maintain
- No obvious return on investment

What distributed crypto achieves

Enterprise view (new)

- Equipment is (fairly) cheap
- At worst, the last '9' becomes an '8' (easy to bump back up)
  - E.g., 99.999% ➔ 99.998%
- (1) Layered security; (2) Enterprises liable for breaches, not phishing
- Diversification, e.g., virtualize with distinct OSs
- Put one server in the cloud
  - Good for security too
- Breaches are becoming commonplace

Enterprise view (old)

- Twice the equipment cost
- Diminished availability (or even higher cost)
  - "Five nines" ➔ "Four nines"
- Not solving real problem: phishing and social engineering, i.e., endpoint
- Breaching two systems as easy as breaching one
- Two distinct systems to maintain
- No obvious return on investment

What distributed crypto achieves

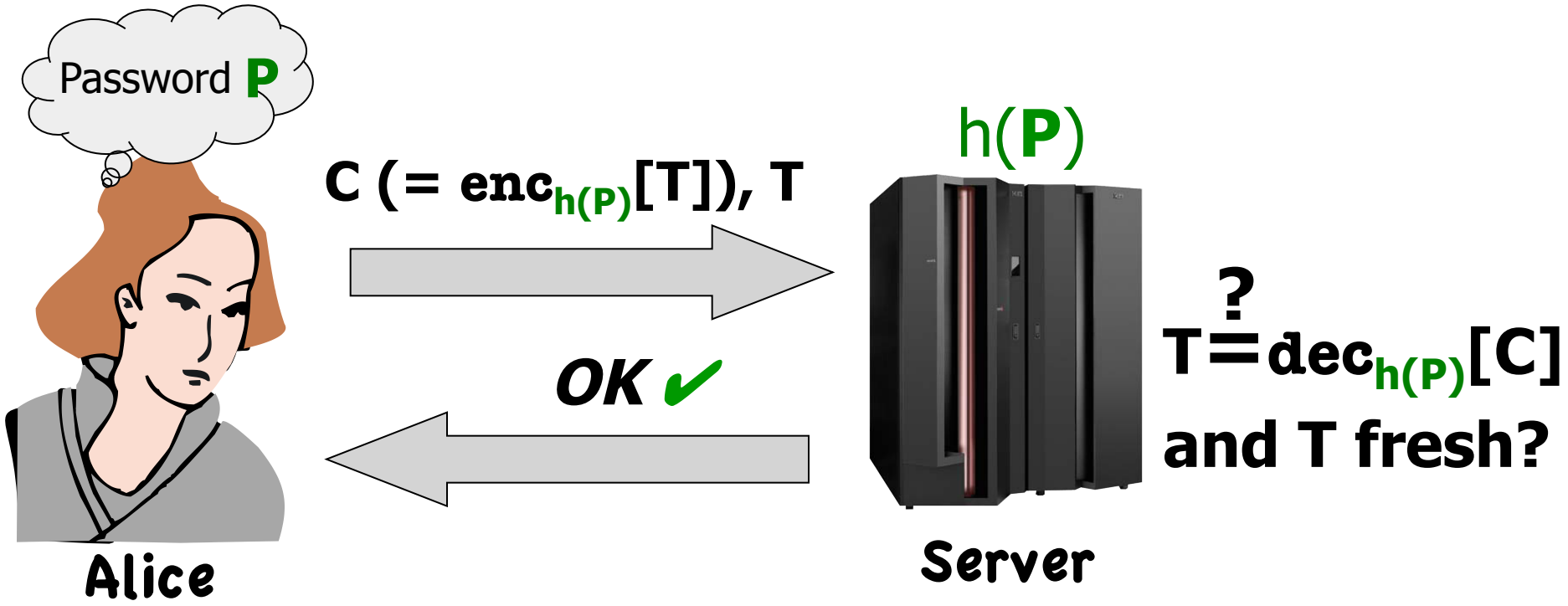
Perimeter security yielding in practice to assumption that adversary is always in the network

- Breaches are becoming commonplace

Practical research challenges

- How can we *efficiently* distribute other forms of authentication? E.g.,
  - Password-based Kerberos
  - One-time passcodes
  - Symmetric-key challenge-response
  - Biometrics
- How should we schedule proactivization epochs?
- What else should we distribute, beyond cryptography?
  - Access control?

Password-based Kerberos
(simplified)



There's no efficient way to distribute decryption function dec (e.g., AES) across two servers!

But we can cheat a little

h(**P**)

Server

But we can cheat a little

h(**P**)

h(**P**)

Mix in *chaff* passwords

$h(P_1)$

$h(\mathbf{P})$

$h(P_2)$

$h(P_3)$

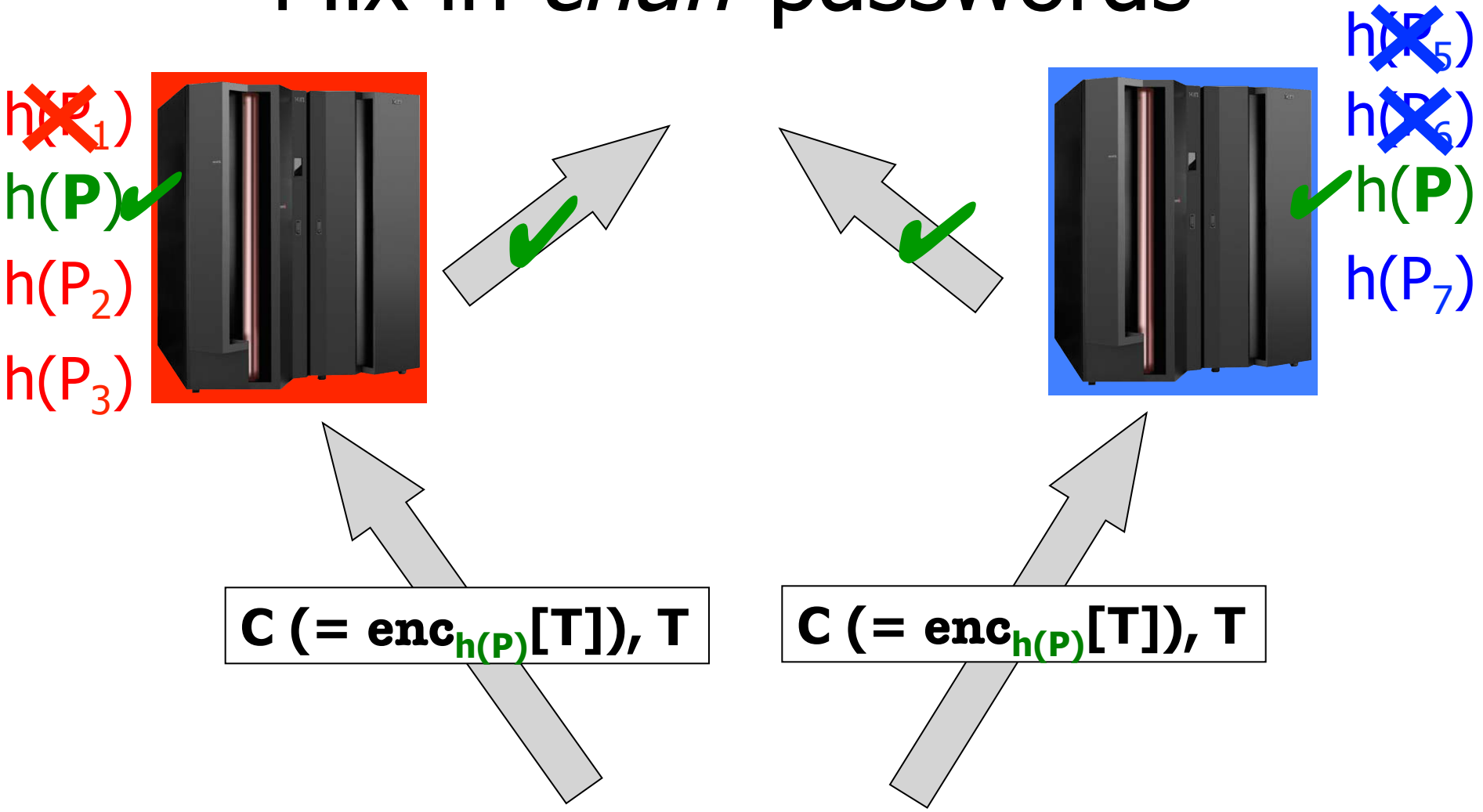
$h(P_5)$

$h(P_6)$

$h(\mathbf{P})$

$h(P_7)$

Mix in *chaff* passwords



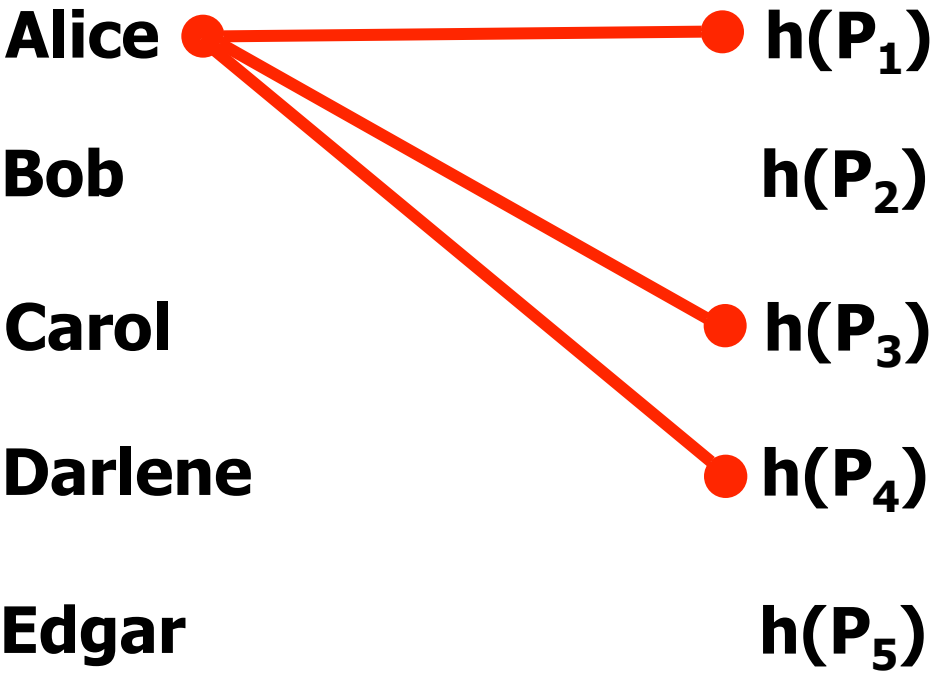
What does this buy us?

$h(P_1)$
$h(\mathbf{P})$
$h(P_2)$
$h(P_3)$

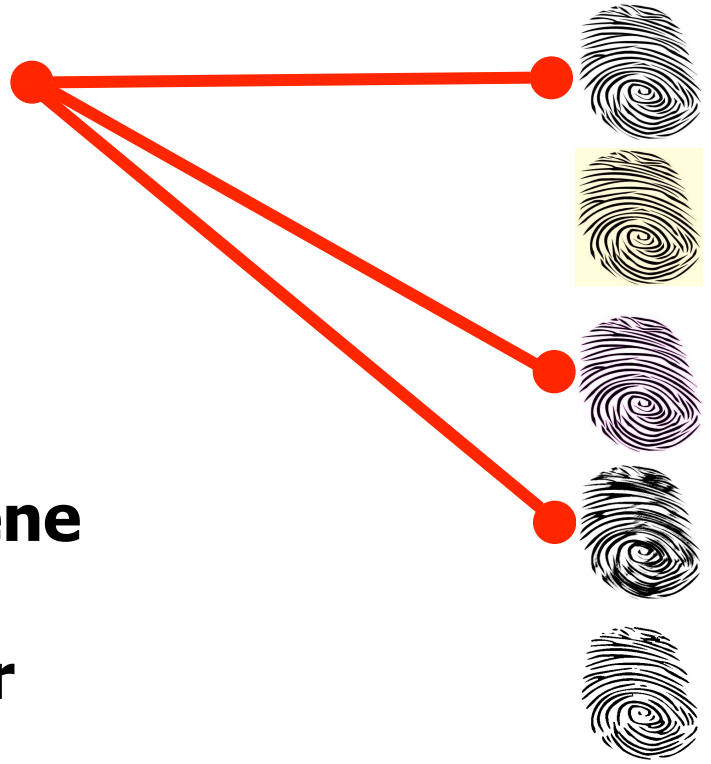
$h(P_5)$
$h(P_6)$
$h(\mathbf{P})$
$h(P_7)$

- **Basic principle: Real password is intersection of two, chaff-laden sets**
- If attacker breaches only one server, it doesn't learn h(P), only candidate set.
  - It can cause one server (w.p. 1) to say "yes," but not both.
- If attacker observes real authentication, it can learn h(P).
- Not as strong as true splitting, but much stronger than current approaches—and legacy-compatible

No need to fabricate chaff



Easy to extend to other credentials



Conclusion

- We may be on the cusp of broad deployment of distributed crypto.
- Distributed crypto for real systems is challenging (and interesting), and requires new / better tools.
  - Chaff?
  - Garbled circuits?
- Lots of other impactful "real-world" questions
  - Proactivization regimes, retrofitting to existing protocols, etc., etc.

Questions?