Lecture 40: Certificate Transparency

https://cs161.org

Announcements

- Project 3 Part 2 - due Sunday 5/3, 11:59pm
- Final exam - regularly scheduled time

Warmup question

- Suppose Mallory tricks a certificate authority into issuing her a cert for anything she wants. What can she do to pretend to be google.com?

Design challenge

- How can we detect if a CA wrongly issues a certificate? Assume you can require CAs to take extra steps to help us verify that they're acting appropriately; what would you require?

- Hint: Bitcoin???

Transparency

- “Sunlight is the best disinfectant.”
  — Supreme Court Justice Louis Brandeis

Candidate design

- Design: Whenever a CA issues a certificate, it adds it to a public blockchain (the log). Site owners can monitor the log for illicit certificates for their site they didn't request.

- Advantages of this design? Disadvantages?

Today, Microsoft issued a Security Advisory warning that fraudulent digital certificates were issued by the Comodo Certificate Authority. This could allow malicious spoofing of high profile websites, including Google, Yahoo! and Windows Live.

The advisory states how 9 certificates were fraudulently issued by Comodo for the following names:

- login.live.com
- mail.google.com
- www.google.com
- login.yahoo.com (3 certificates)
- login.skype.com
- addons.mozilla.org
- “Global Trustee”

The attacker who penetrated the Dutch CA DigiNotar last year had complete control of all eight of the company's certificate-issuing servers during the operation and he may also have issued some rogue certificates that have not yet been identified. The final report from a security company commissioned to investigate the DigiNotar attack shows that the compromise of the now-bankrupt certificate authority was much deeper than previously thought.

A Dutch certificate authority that suffered a major hack attack this summer has been unable to recover from the blow and filed for bankruptcy this week.

Design challenge

- If Mallory compromises the CA, she could issue a bogus cert for google.com and then not add it to the public log. How could we detect that?

Candidate design

- Design: Whenever a CA issues a certificate, it adds it to a public blockchain (the log). Site owners can monitor the log. Whenever a browser receives a cert, it checks that it is in the log.

- Advantages of this design? Disadvantages?

Performance analysis

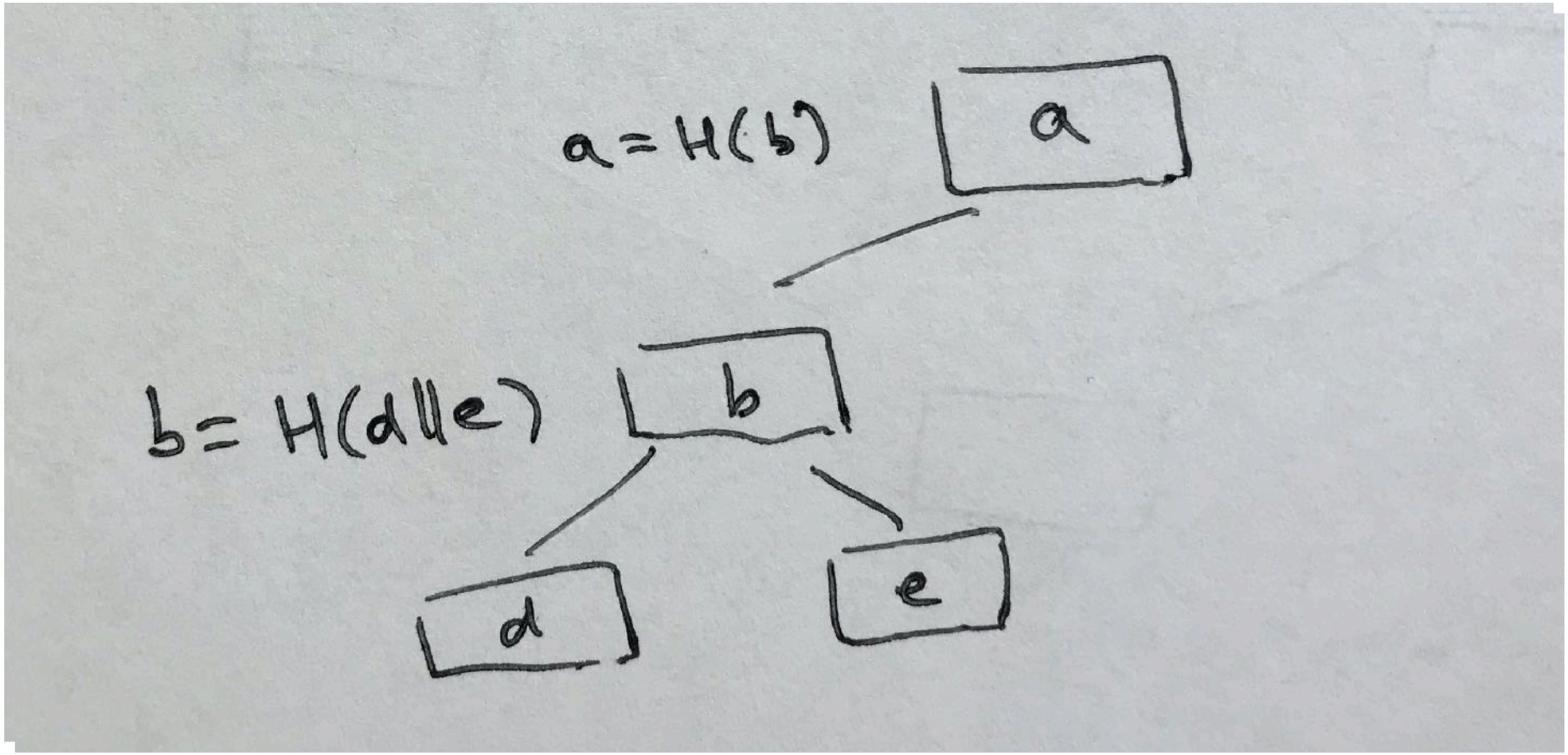
- Design: Whenever a CA issues a certificate, it adds it to a public blockchain (the log). Site owners can monitor the log. Whenever a browser receives a cert, it checks that it is in the log.

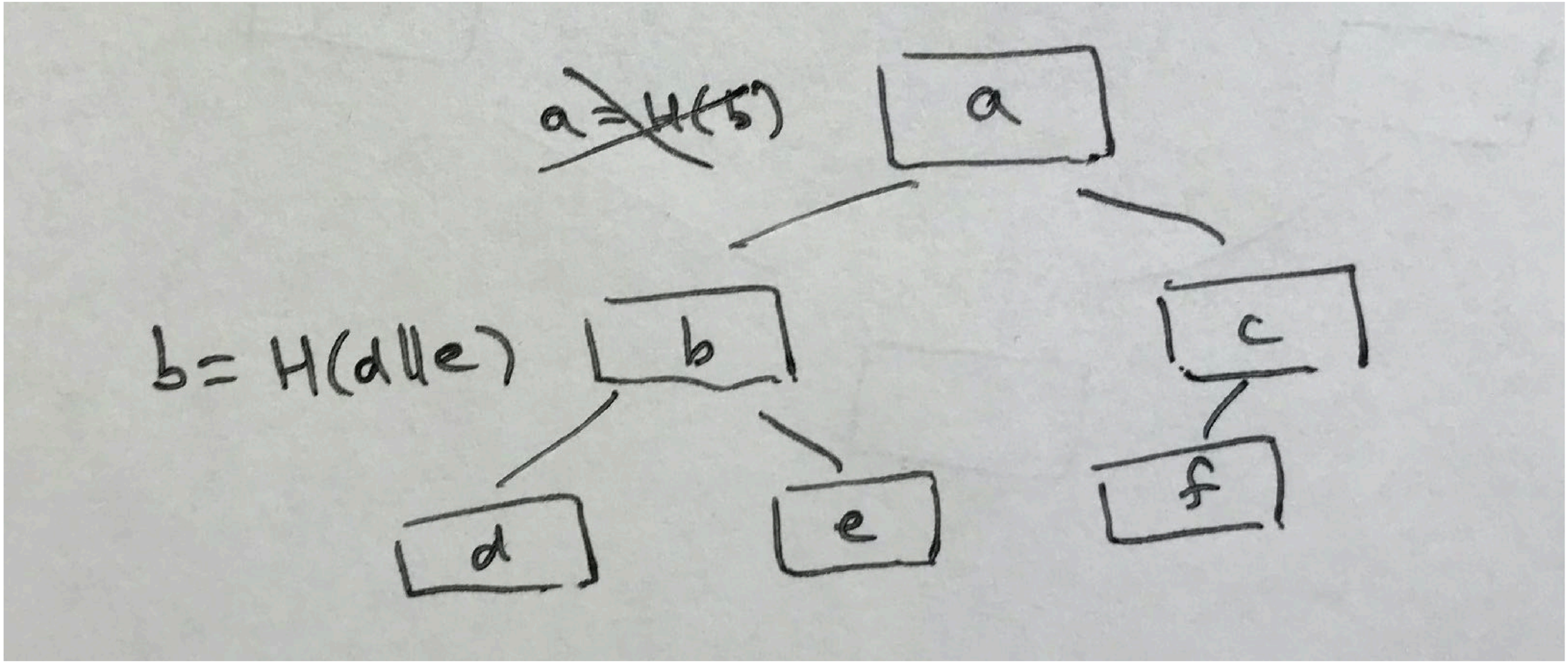
- Download the entire log: O(n) time
- Append a certificate to log: O(1)
- Check that a cert is in the log: O(n) time

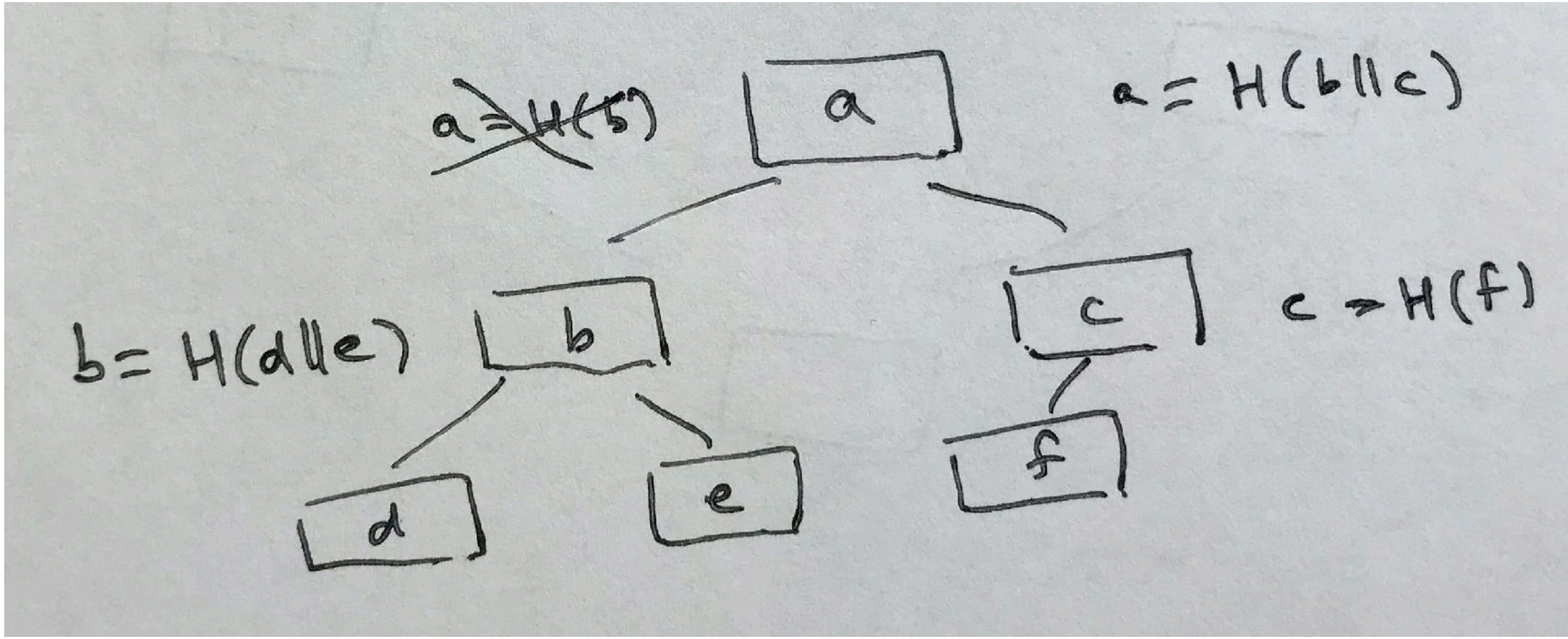
Merkle trees

- A data structure that lets you store a set of items

- Download the entire tree: O(N) time
- Append a certificate to tree: O(log N) time
- Check that a cert is in the tree: O(log N) time


a = H(b)
a
b = H(d||e)
b
d
e


a = H(b)
a
b = H(d||e)
b
c
d
e
f


a = H(b||c)
b = H(d||e)
c = H(f)
a
b
c
d
e
f

Certificate Transparency

- Public log of all issued certificates. Anyone can monitor. Browsers can check that each cert they receive is in log.

- Download the entire tree: O(N) time
- Append a certificate to tree: O(log N) time
- Check that a cert is in the tree: O(log N) time

Application: Database Integrity

- You are using a database stored in the cloud, and you're not sure if you trust the cloud server. How can you check that the responses from the server to queries are correct?
- Examples: SELECT username WHERE token=3092
  SELECT name WHERE age>65

Application: Encrypted Filesystems

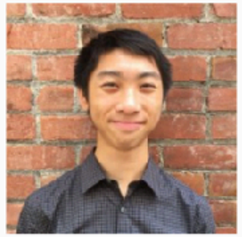
- You encrypt all data in the filesystem (like in Project 2). Now you want to detect rollback attacks (which were out of scope for Project 2). How do you do it?

Takeaways

- Security is about the balance between attacks and defenses.
- Sometimes we can achieve Fort Knox level security (e.g. cryptography). Sometimes we try to win the arms race (e.g., spam and fraud online).

Let’s thank the course staff!


(Head TA) Catherine Han


Allen Tong

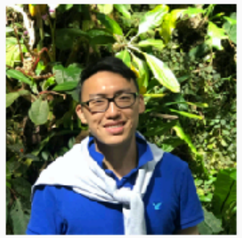

Andrew Law


Peyrin Kao


Sachit Shroff


Seung Jin Yang

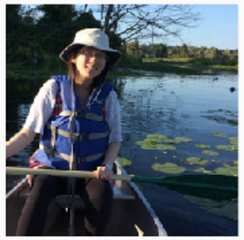

Cathy Lu


Eric Feng


Evan Corriere


Toby Chen


Victor Chan


Vivian Fang


Jason Li XiangJun


Keahooi Hung


Nicholas Ward