Cryptosystems: Theory and Practice

Introduction & Course Overview

Slides adapted from Raluca Ada Popa & Emily Stark

Cryptosystems

Cryptosystems

- Cryptosystems are secure computer systems that utilize advanced cryptographic techniques

Cryptosystems

- Cryptosystems are secure computer systems that utilize advanced cryptographic techniques
- What systems?

Cryptosystems

- Cryptosystems are secure computer systems that utilize advanced cryptographic techniques
- What systems?
  - Database, messaging, ML training & inference, blockchain

Cryptosystems

- Cryptosystems are secure computer systems that utilize advanced cryptographic techniques
- What systems?
  - Database, messaging, ML training & inference, blockchain
- What crypto?

Cryptosystems

- Cryptosystems are secure computer systems that utilize advanced cryptographic techniques
- What systems?
    - Database, messaging, ML training & inference, blockchain
- What crypto?
    - Homomorphic encryption, ORAM, PIR, MPC, differential privacy, zero-knowledge

Goals of this course

Goals of this course

- What important problems do these systems address?

Goals of this course

- What important problems do these systems address?
- Learn how cryptosystems are designed and implemented

Goals of this course

- What important problems do these systems address?
- Learn how cryptosystems are designed and implemented
  - Which cryptographic tools are used?

Goals of this course

- What important problems do these systems address?
- Learn how cryptosystems are designed and implemented
  - Which cryptographic tools are used?
  - How to use these cryptographic tools to build secure systems?

Goals of this course

- What important problems do these systems address?
- Learn how cryptosystems are designed and implemented
  - Which cryptographic tools are used?
  - How to use these cryptographic tools to build secure systems?
  - What systems techniques are used?

Goals of this course

- What important problems do these systems address?
- Learn how cryptosystems are designed and implemented
  - Which cryptographic tools are used?
  - How to use these cryptographic tools to build secure systems?
  - What systems techniques are used?
- Are these systems used in practice? What are difficulties in deploying these systems?

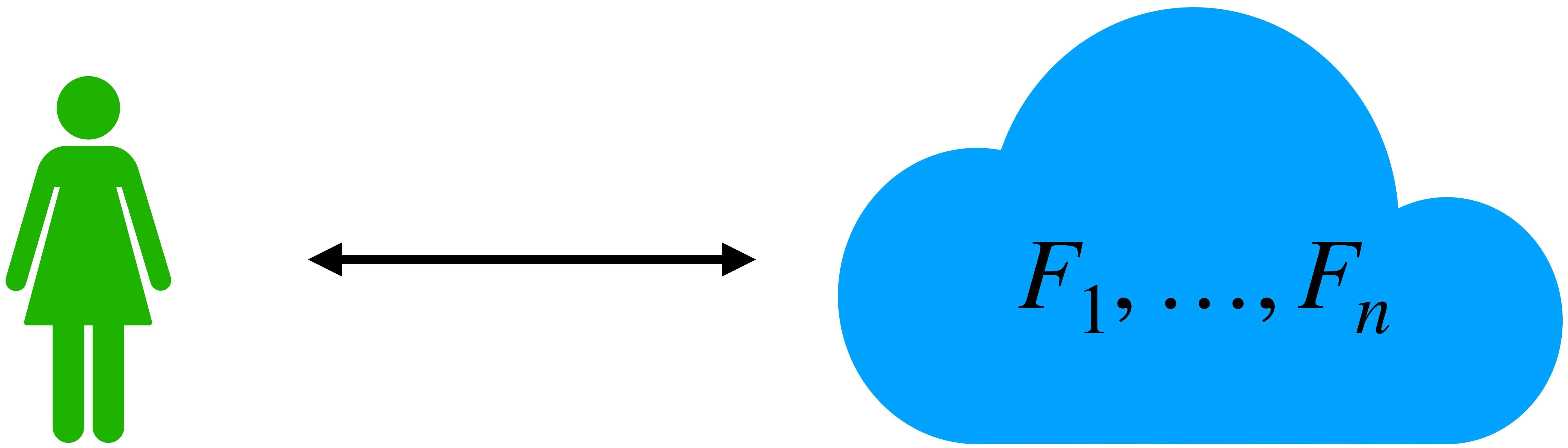
A scenario…

A scenario…

- Alice has files $F_1, \ldots, F_n$ and wants to store them in the cloud

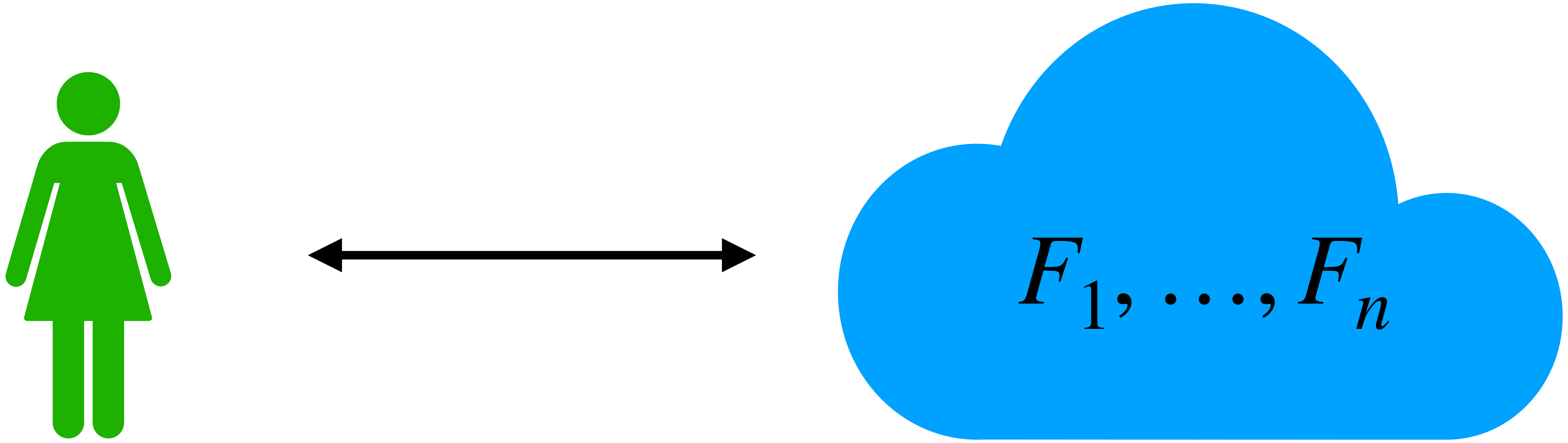
A scenario…

- Alice has files $F_1, \ldots, F_n$ and wants to store them in the cloud



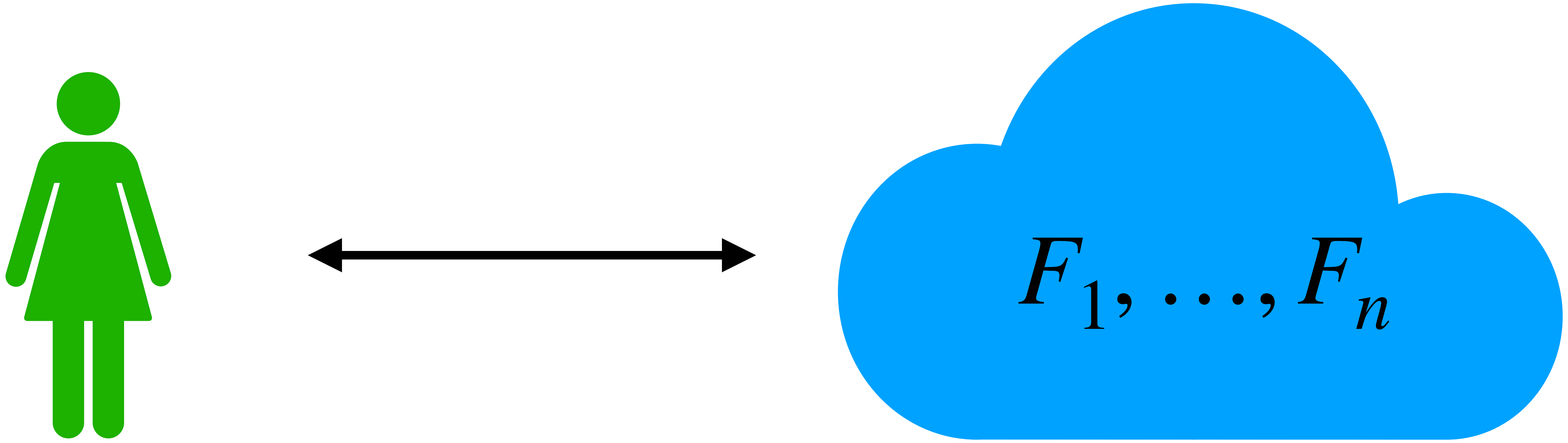
A scenario…

- Alice has files $F_1, \ldots, F_n$ and wants to store them in the cloud
- Alice does not entirely trust the cloud with respect to data integrity



A scenario…

- Alice has files $F_1, \ldots, F_n$ and wants to store them in the cloud
- Alice does not entirely trust the cloud with respect to data integrity
- When she retrieves file i, how can she verify that the untrusted cloud did not modify the file?



Collision resistant hash function (CRHF)

Collision resistant hash function (CRHF)

- $H : \{0,1\}^* \rightarrow \{0,1\}^m$ is a collision resistant hash function if for all PPT algorithms A, for all k sufficiently large:

$$Pr[(x, y) \leftarrow A(1^k) \ s.t. H(x) = H(y) \wedge x \neq y] \leq negl(k)$$

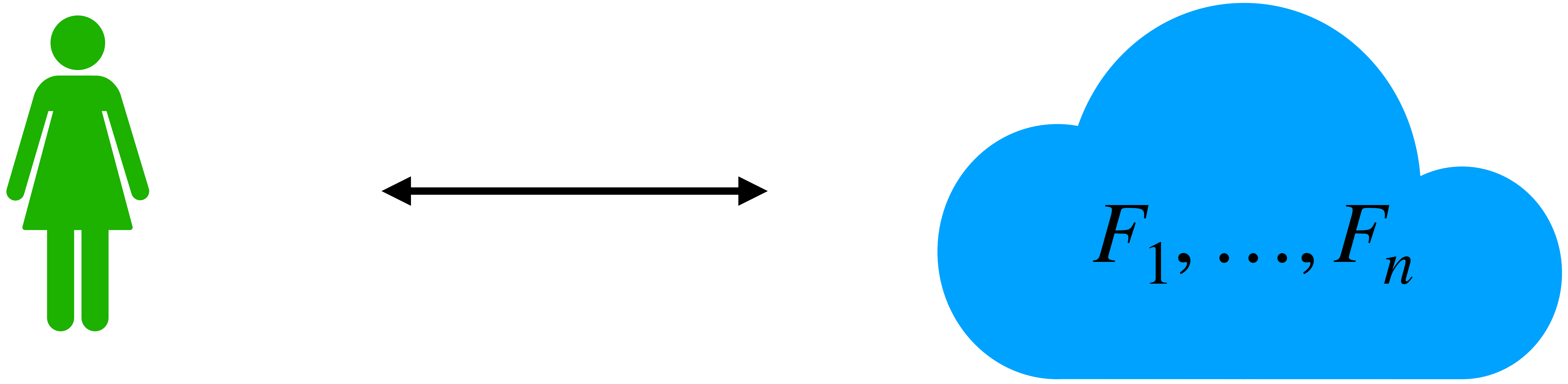
Collision resistant hash function (CRHF)

- $H : \{0,1\}^* \to \{0,1\}^m$ is a collision resistant hash function if for all PPT algorithms A, for all k sufficiently large:

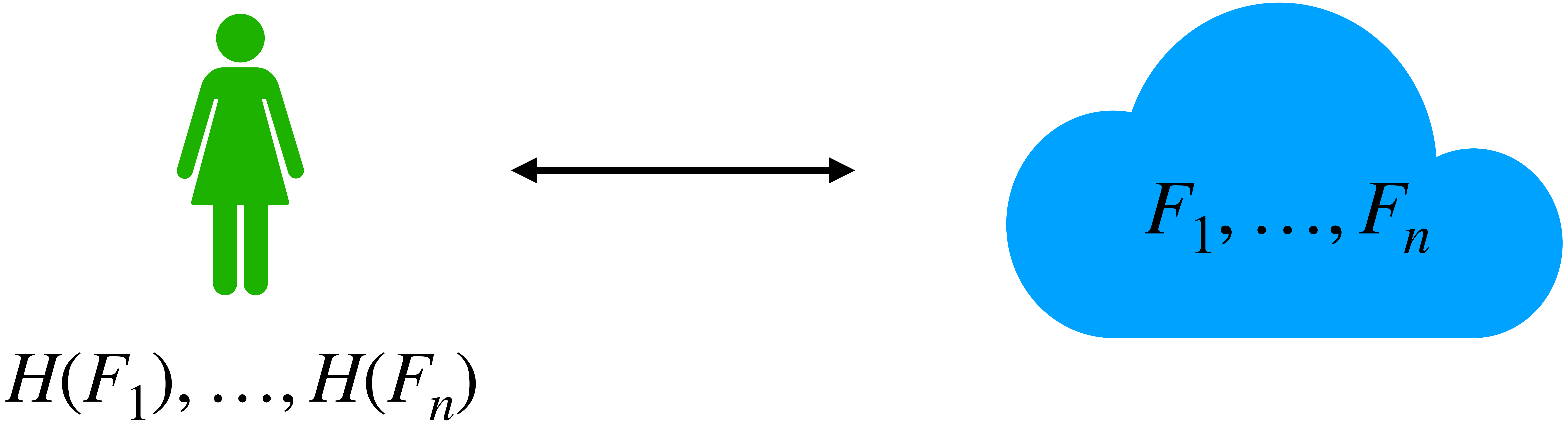
  $$Pr[(x, y) \leftarrow A(1^k) \ s.t. H(x) = H(y) \wedge x \neq y] \leq negl(k)$$

- In order words, it is computationally hard for the adversary to find two messages x and y such that their hashes are the same

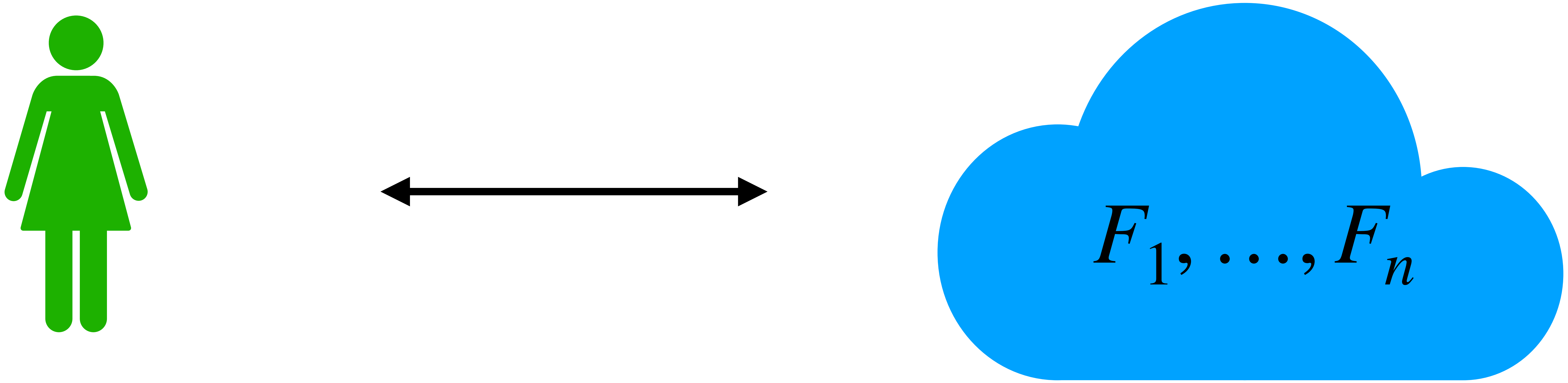
A first attempt



A first attempt



A first attempt



$H(F_1), \ldots, H(F_n)$

Problem: large amount of client storage

Merkle tree

- Invented by Ralph Merkle in 1979
- Used in many theoretic constructions and practical crypto systems

Merkle tree

Merkle tree

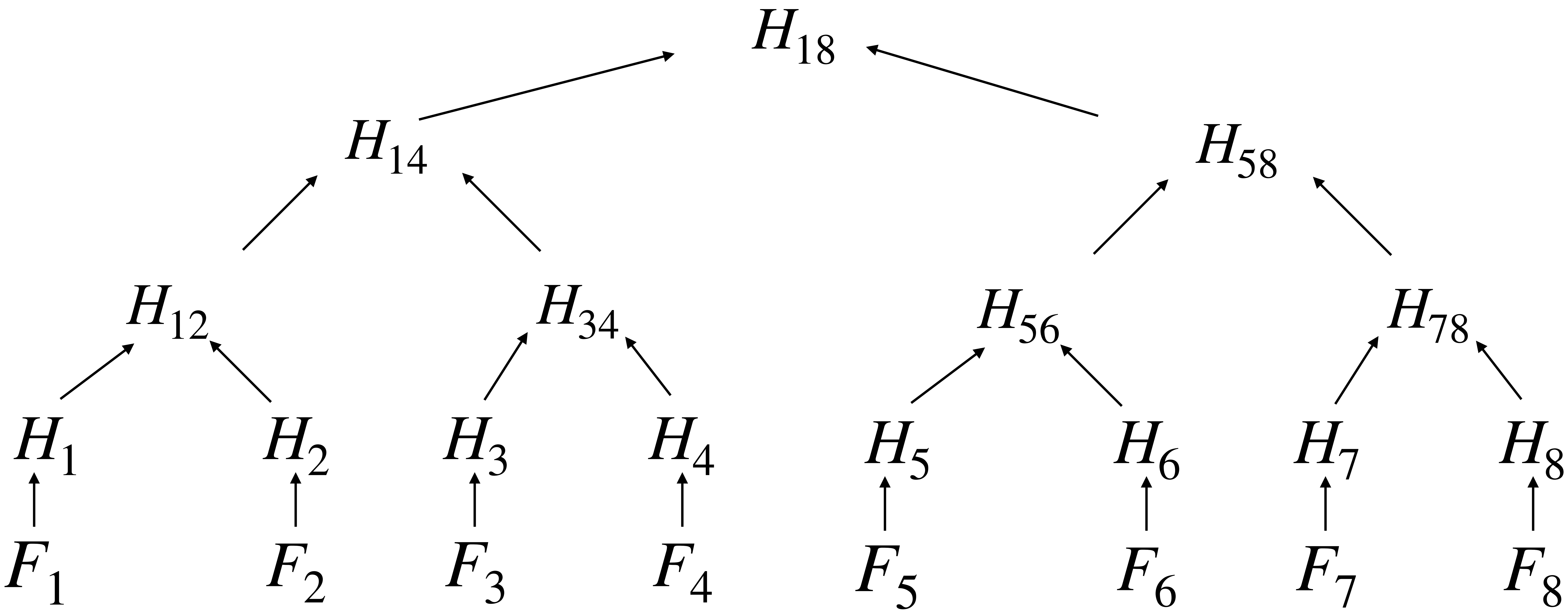
- A hash tree over a set of data values $F_1, \ldots, F_n$

Merkle tree

- A hash tree over a set of data values $F_1, \ldots, F_n$
- Each node is the hash of its two children

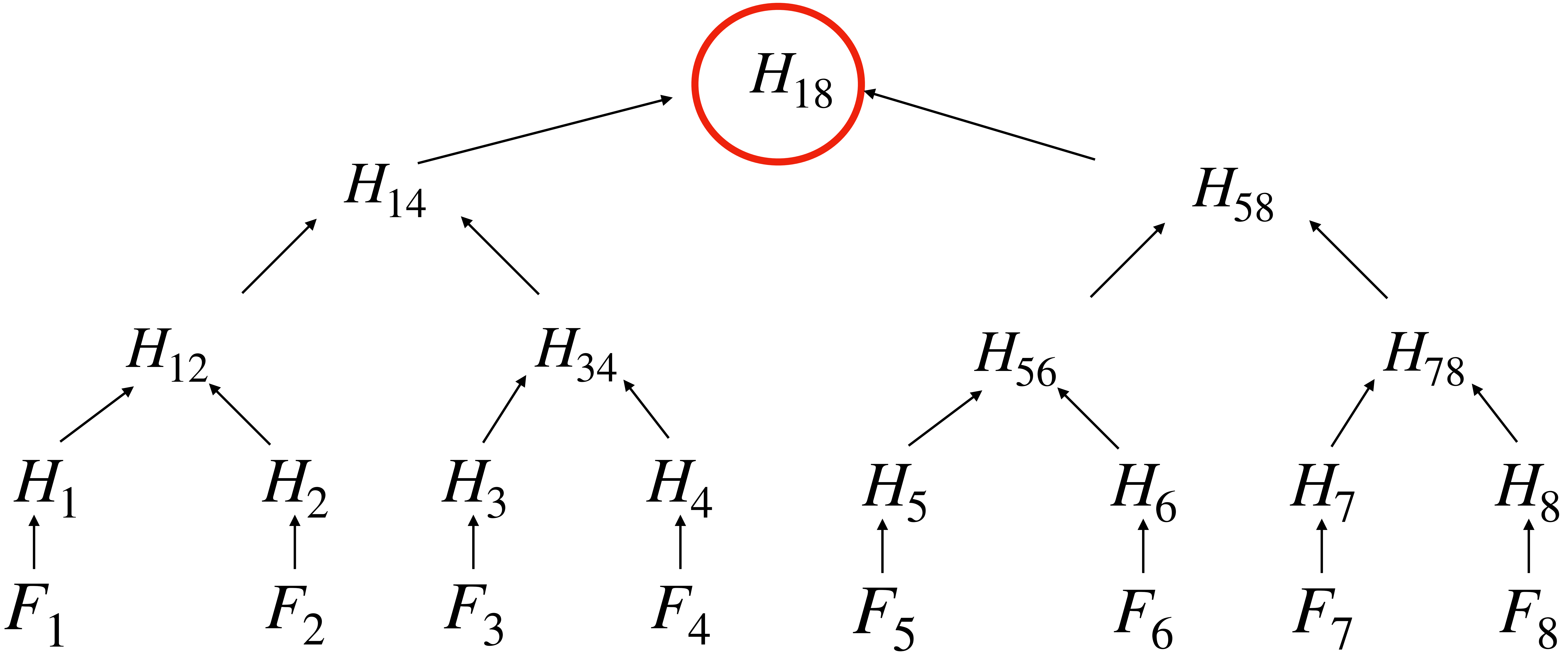
Merkle tree

- A hash tree over a set of data values $F_1, \ldots, F_n$
- Each node is the hash of its two children



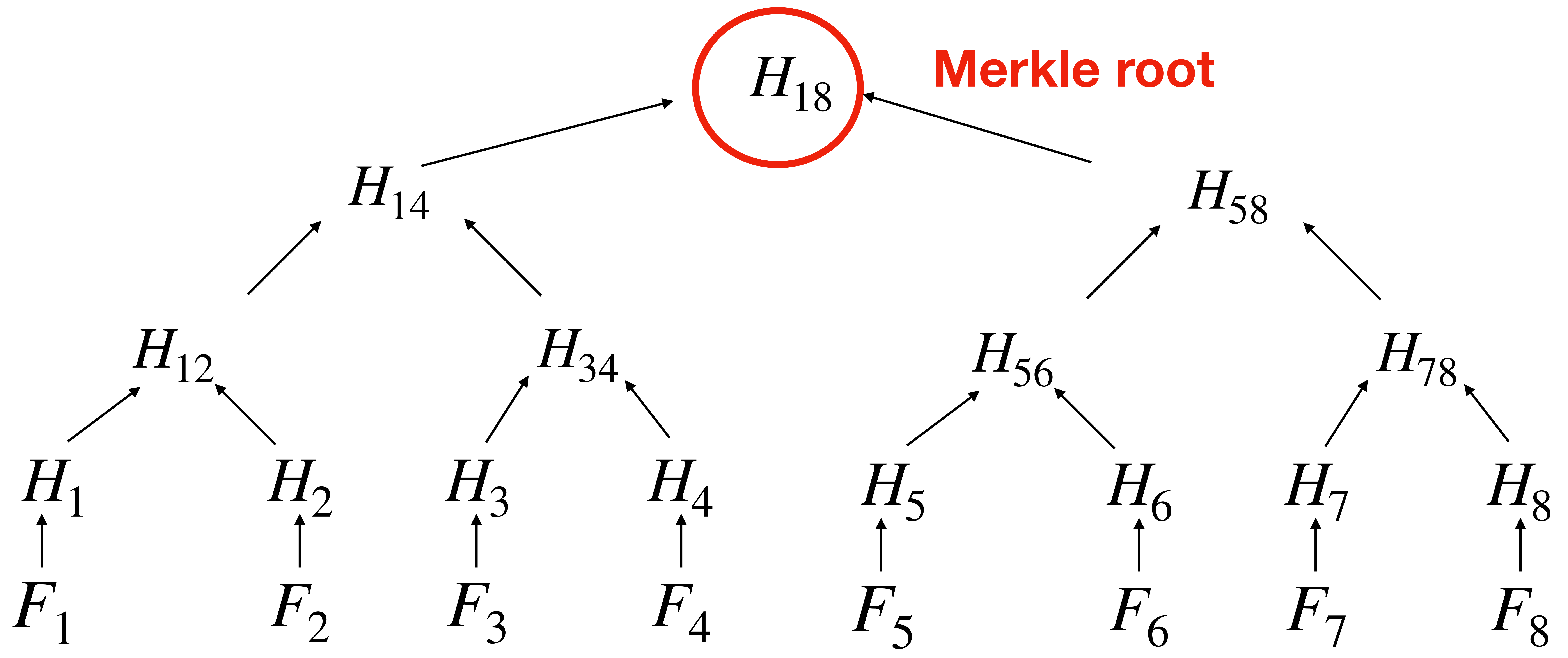
Merkle tree

- A hash tree over a set of data values $F_1, \ldots, F_n$
- Each node is the hash of its two children

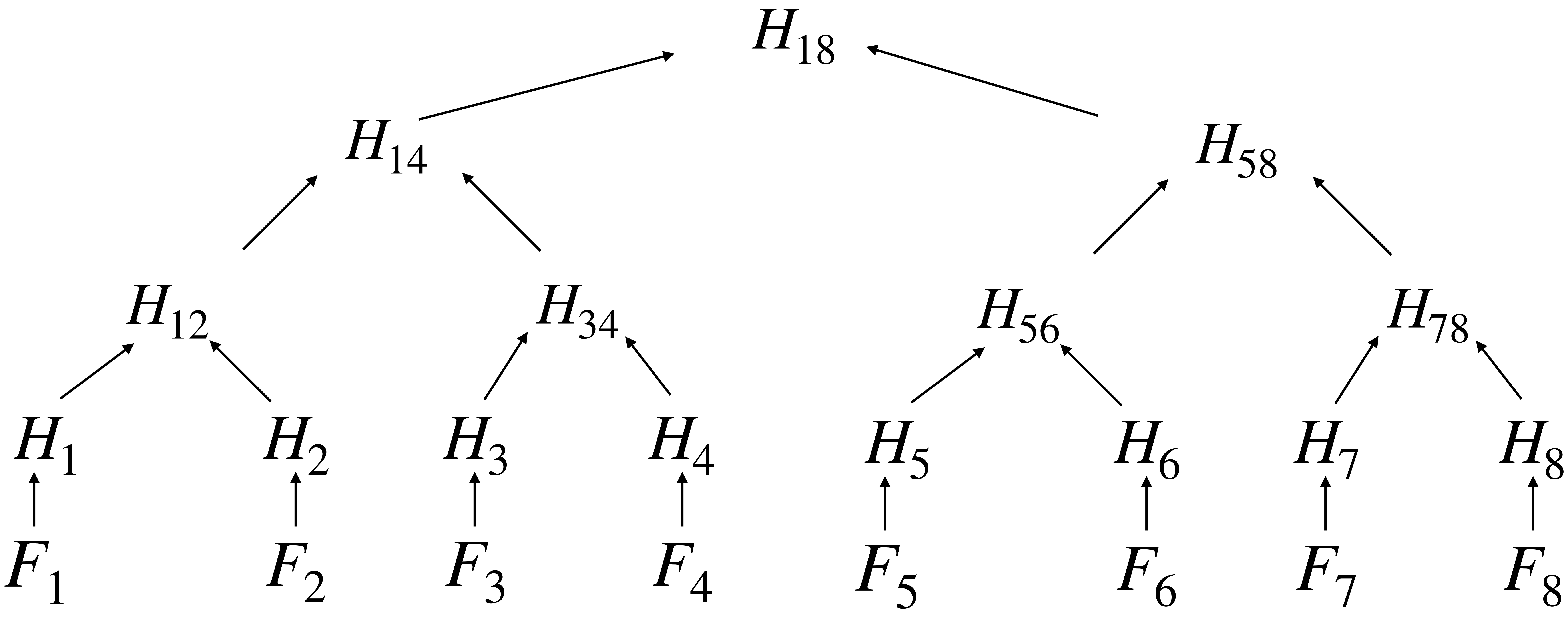


Merkle tree

- A hash tree over a set of data values $F_1, \ldots, F_n$
- Each node is the hash of its two children

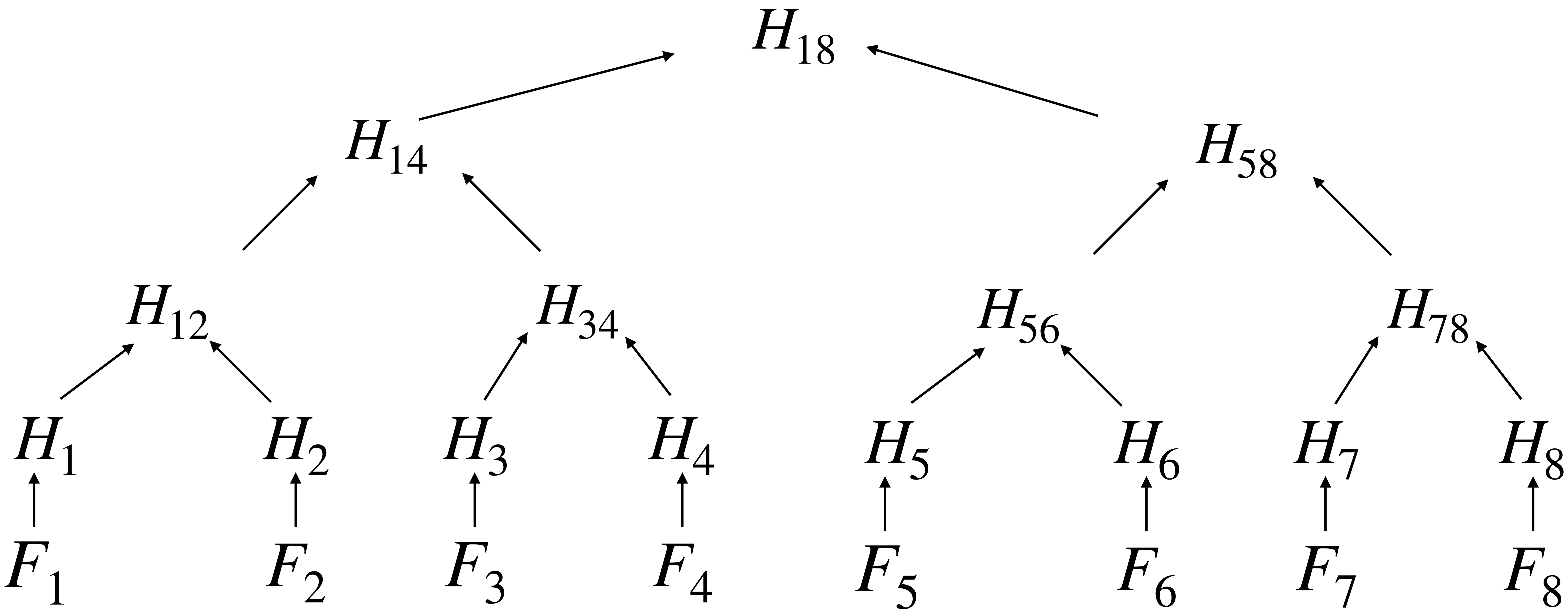


Merkle proof



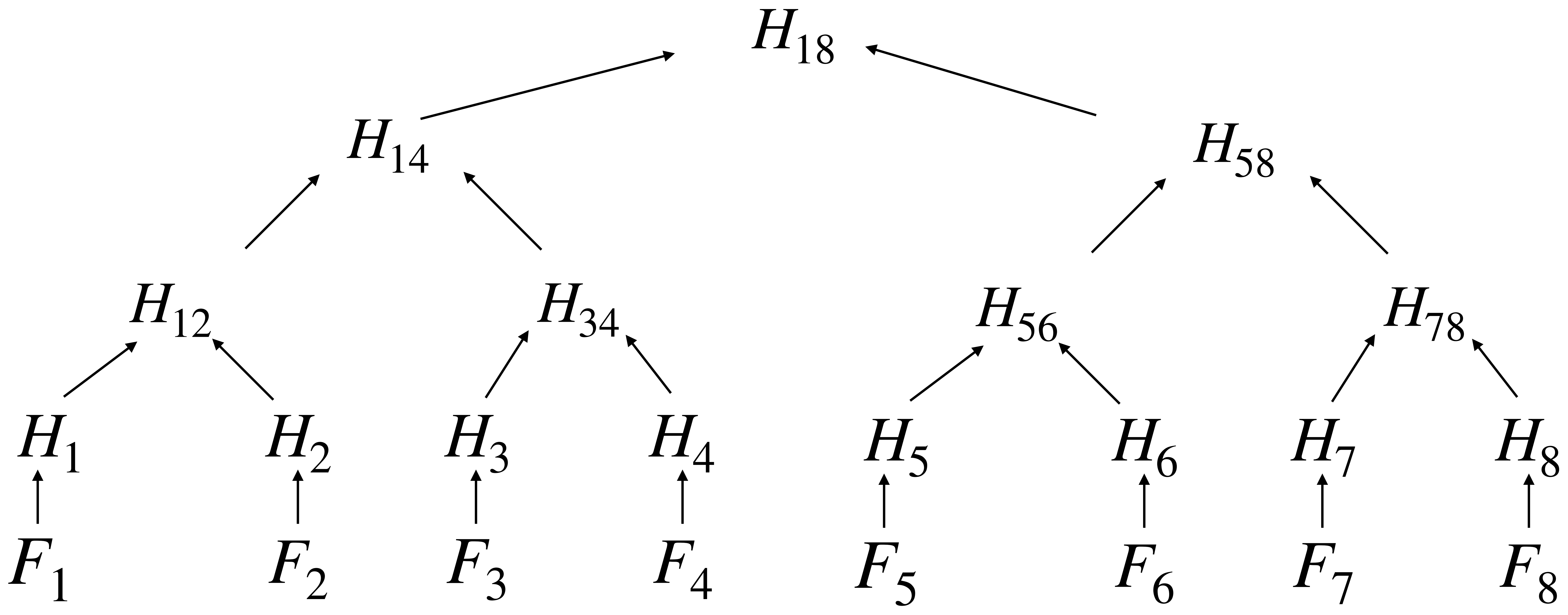
Merkle proof

- Alice wants to retrieve F_2

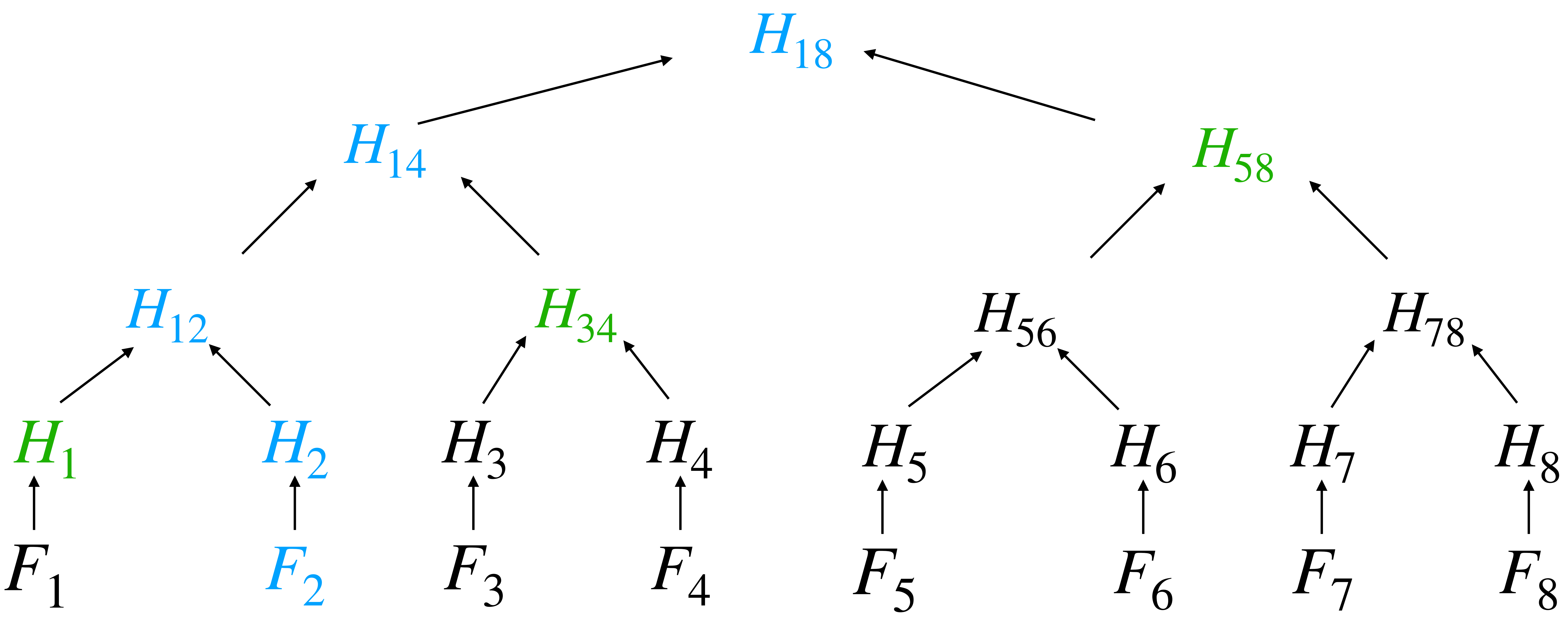


Merkle proof

- Alice wants to retrieve F_2
- Given this summary, how can F_2 be authenticated to Alice?

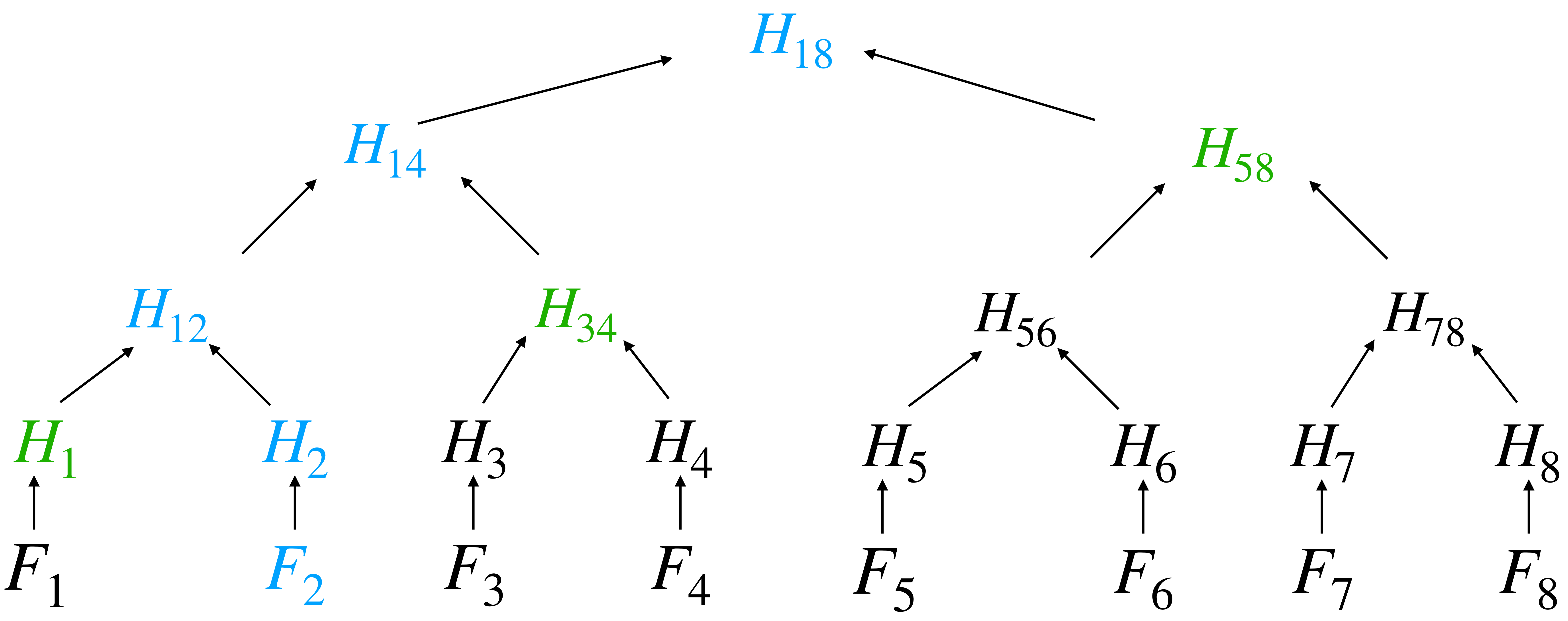


Merkle proof



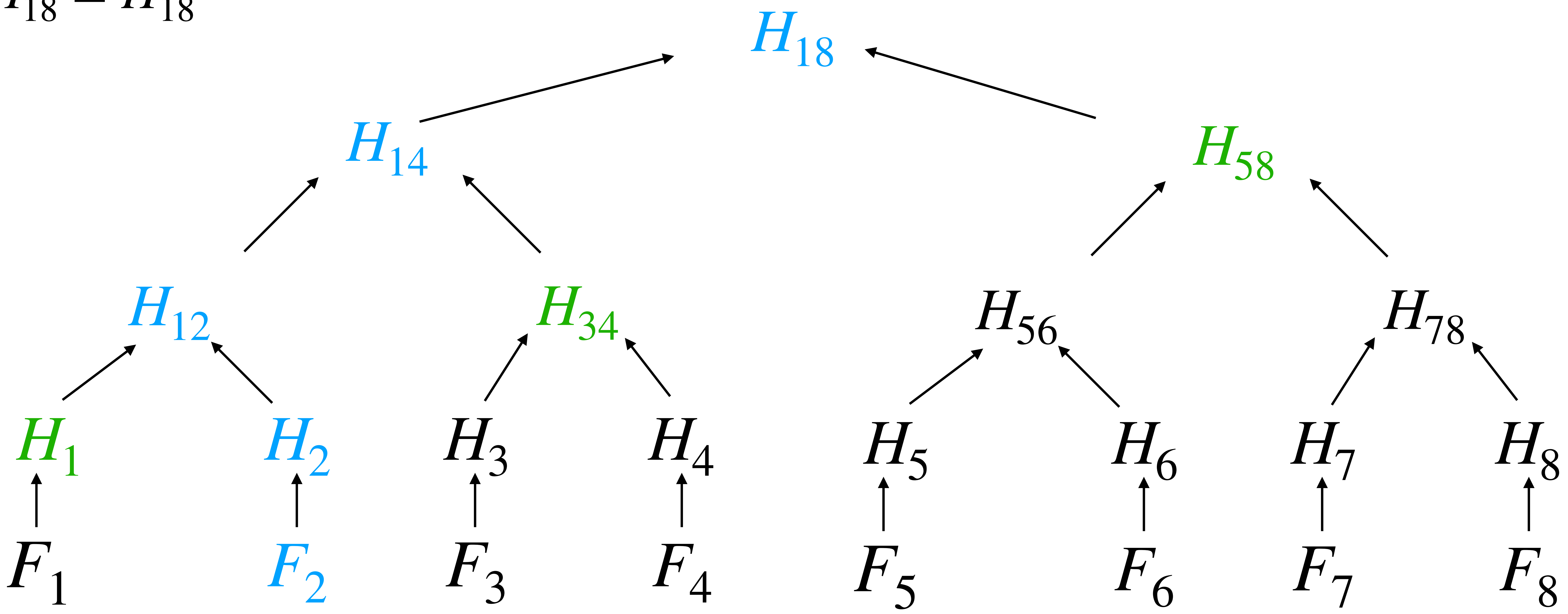
Merkle proof

- The server provides a Merkle proof, which are the siblings of nodes from F_2 to the root: H_1, H_{34}, H_{58}



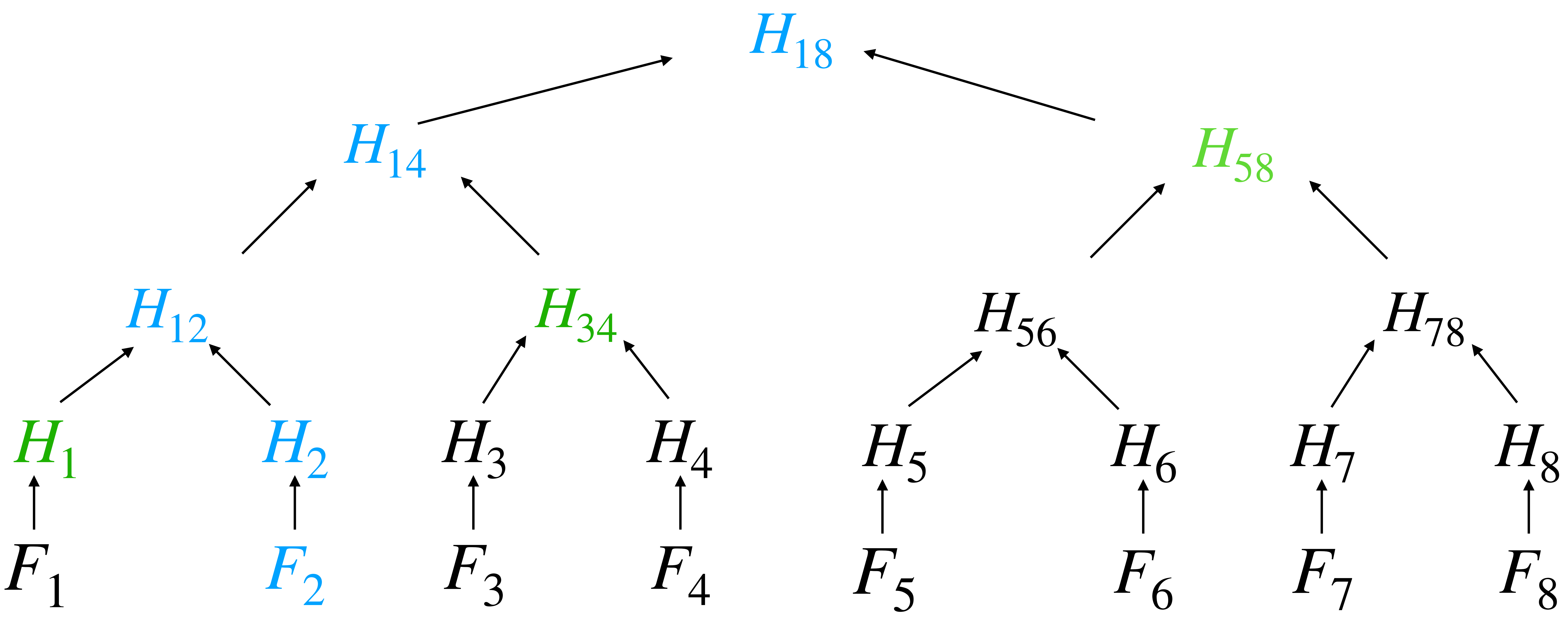
Merkle proof

- The server provides a Merkle proof, which are the siblings of nodes from F_2 to the root: H_1, H_{34}, H_{58}
- Alice verifies that $\hat{H}_2 \leftarrow H(F_2), \hat{H}_{12} = H(H_1, \hat{H}(2)), \hat{H}_{14} \leftarrow H(\hat{H}_{12}, H_{34}), \hat{H}_{18} = H(\hat{H}_{14}, H_{58})$ and that $H_{18} = \hat{H}_{18}$



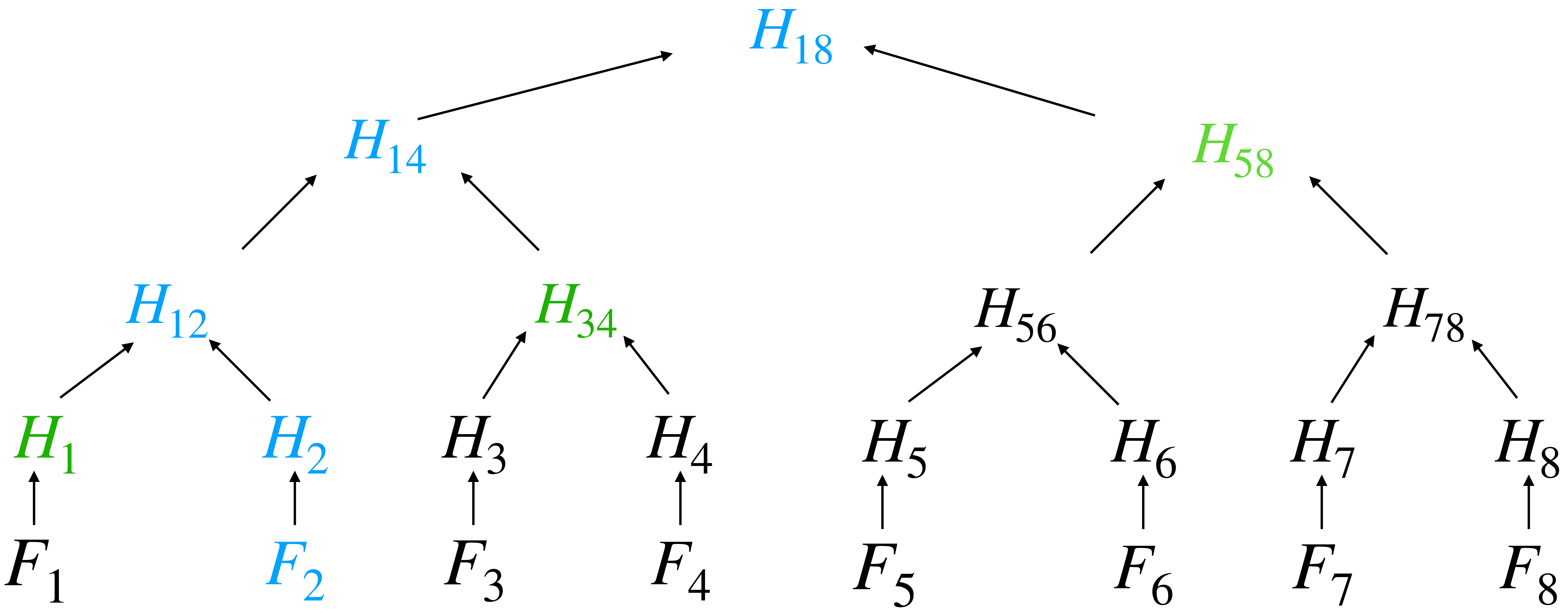
Proof of security?

- **Theorem (Merkle proof consistency):** It is infeasible to output a Merkle root h and two inconsistent proofs π_i and π_i' for two different inputs x_i and x_i' at the ith leaf in the tree of size n.



Proof of security?

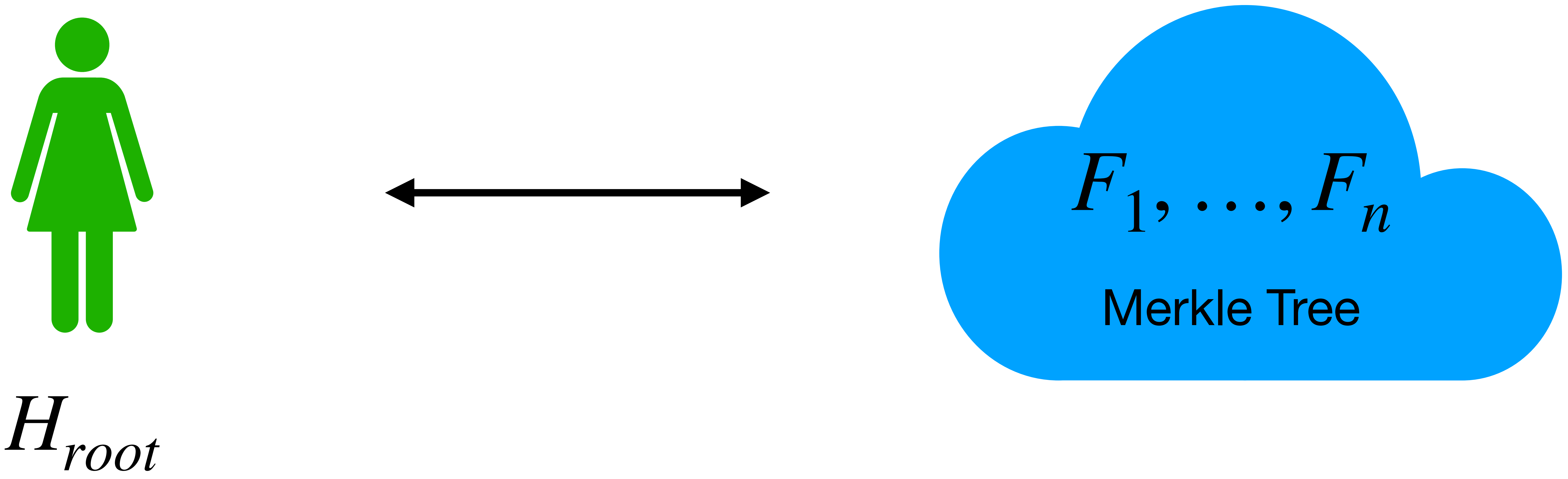
- If $F_2 \neq F_2'$ but the computed root hashes are the same, then there must exist some level $j \in [k]$ where there is a collision. But collision at level j implies a break in the collision-resistance of H



Asymptotics

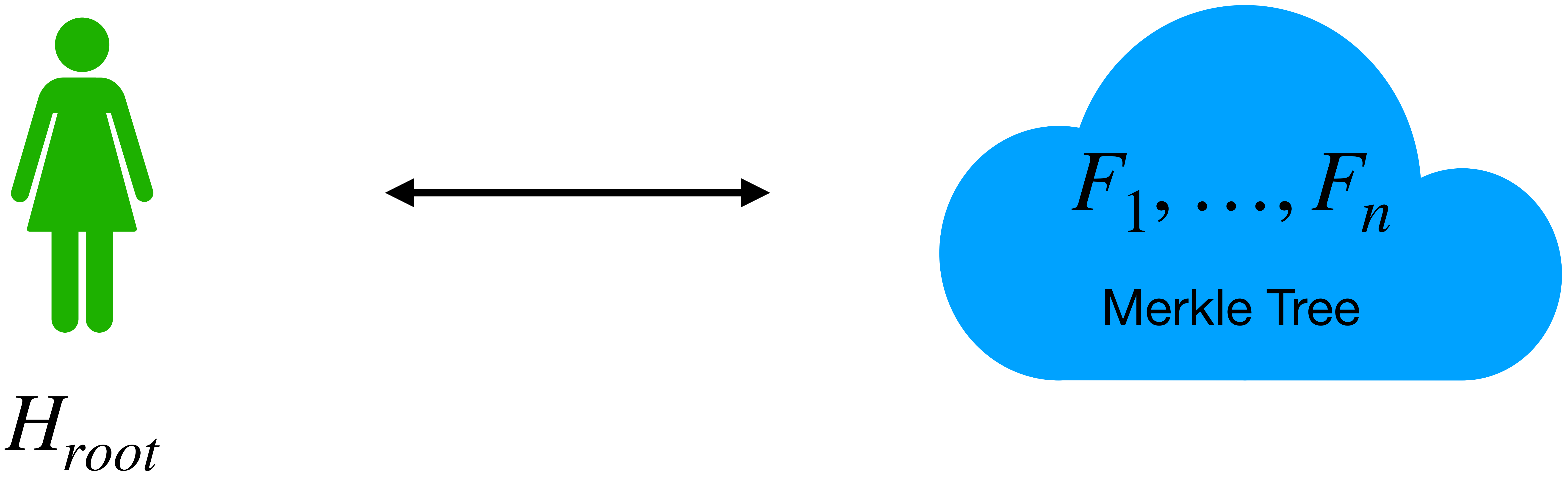
- n number of data items, m hash size
- Size of Merkle tree: $O(nm)$
- Size of Merkle root: $O(m)$
- Size of Merkle proof: $O(m \log n)$

A better attempt



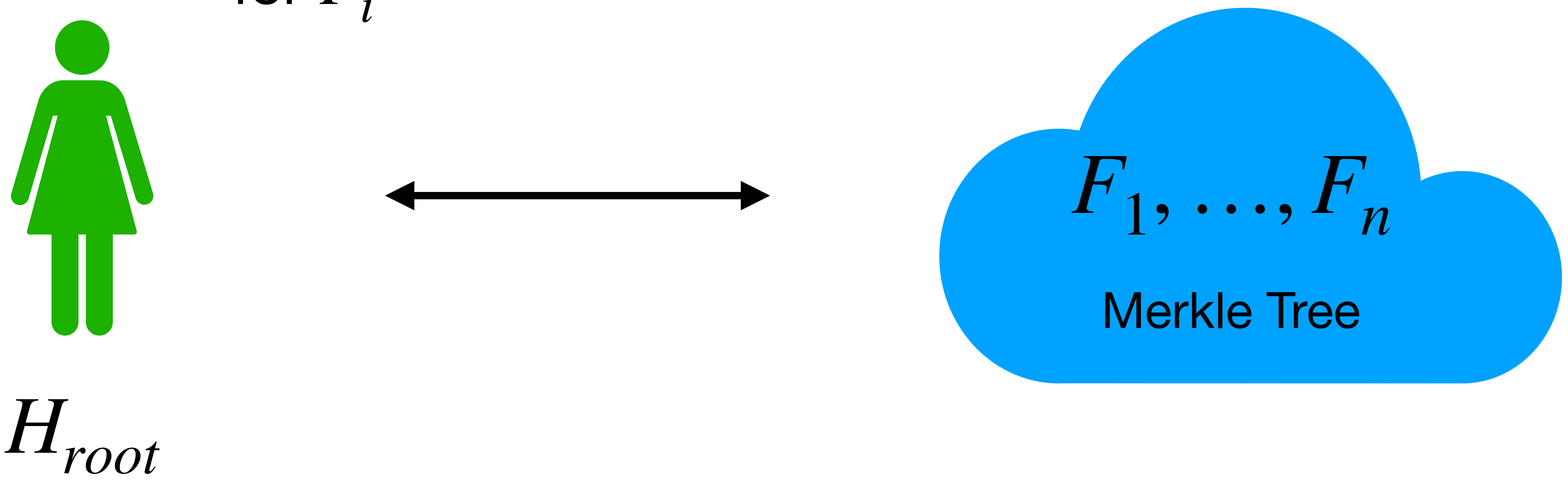
A better attempt

- Alice keeps the Merkle root H_{root}



A better attempt

- Alice keeps the Merkle root H_{root}
- Asks the server for a Merkle proof for F_i



Certificate transparency

Web certificates

Web certificates

- A website obtains a certificate of the form $sign_{CA}(PK_{bank}$, "*bank.com*", expiry), where CA is a certificate authority trusted by user browsers

Web certificates

- A website obtains a certificate of the form $sign_{CA}(PK_{bank}$, "*bank.com*", expiry), where CA is a certificate authority trusted by user browsers

- CAs have often been compromised

Web certificates

- A website obtains a certificate of the form $sign_{CA}(PK_{bank},$ “*bank.com*”, expiry), where CA is a certificate authority trusted by user browsers

- CAs have often been compromised

> 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.

Web certificates

- A website obtains a certificate of the form $sign_{CA}(PK_{bank},$ “*bank.com*”, expiry), where CA is a certificate authority trusted by user browsers
- CAs have often been compromised

Today, Microsoft issued a Security Advisory warning that fraudulent digital certificates were issued by the Comodo Certificate Authoı allow malicious spoofing of high profile websites, includin and Windows Live.

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.

Certificate transparency

Certificate transparency

- We may not be able to prevent attackers from using malicious certificates, but we may be able to make all certificates **discoverable**

Certificate transparency

- We may not be able to prevent attackers from using malicious certificates, but we may be able to make all certificates **discoverable**
- Append certificates to public logs that are hosted in a decentralized manner

Certificate transparency

- We may not be able to prevent attackers from using malicious certificates, but we may be able to make all certificates **discoverable**
- Append certificates to public logs that are hosted in a decentralized manner
- Independent parties (end users, domain owners, etc.) can be watch dogs for malicious certificates

Certificate transparency

- We may not be able to prevent attackers from using malicious certificates, but we may be able to make all certificates **discoverable**
- Append certificates to public logs that are hosted in a decentralized manner
- Independent parties (end users, domain owners, etc.) can be watch dogs for malicious certificates
- Using Merkle trees, these independent parties can verify “summary” of the logged certificates and detect inconsistencies

Parties

Parties

- Log servers: stores certificates in logs

Parties

- Log servers: stores certificates in logs
- Auditors: audit the log is append-only

Parties

- Log servers: stores certificates in logs
- Auditors: audit the log is append-only
  - Anyone can be an auditor

Parties

- Log servers: stores certificates in logs
- Auditors: audit the log is append-only
  - Anyone can be an auditor
  - Untrusted except that at least one auditor should be honest and reachable

Parties

- Log servers: stores certificates in logs
- Auditors: audit the log is append-only
  - Anyone can be an auditor
  - Untrusted except that at least one auditor should be honest and reachable
- Domain owners: monitor their certificates in the log

Parties

- Log servers: stores certificates in logs
- Auditors: audit the log is append-only
  - Anyone can be an auditor
  - Untrusted except that at least one auditor should be honest and reachable
- Domain owners: monitor their certificates in the log
  - Trusted to monitor their own certificates

Parties

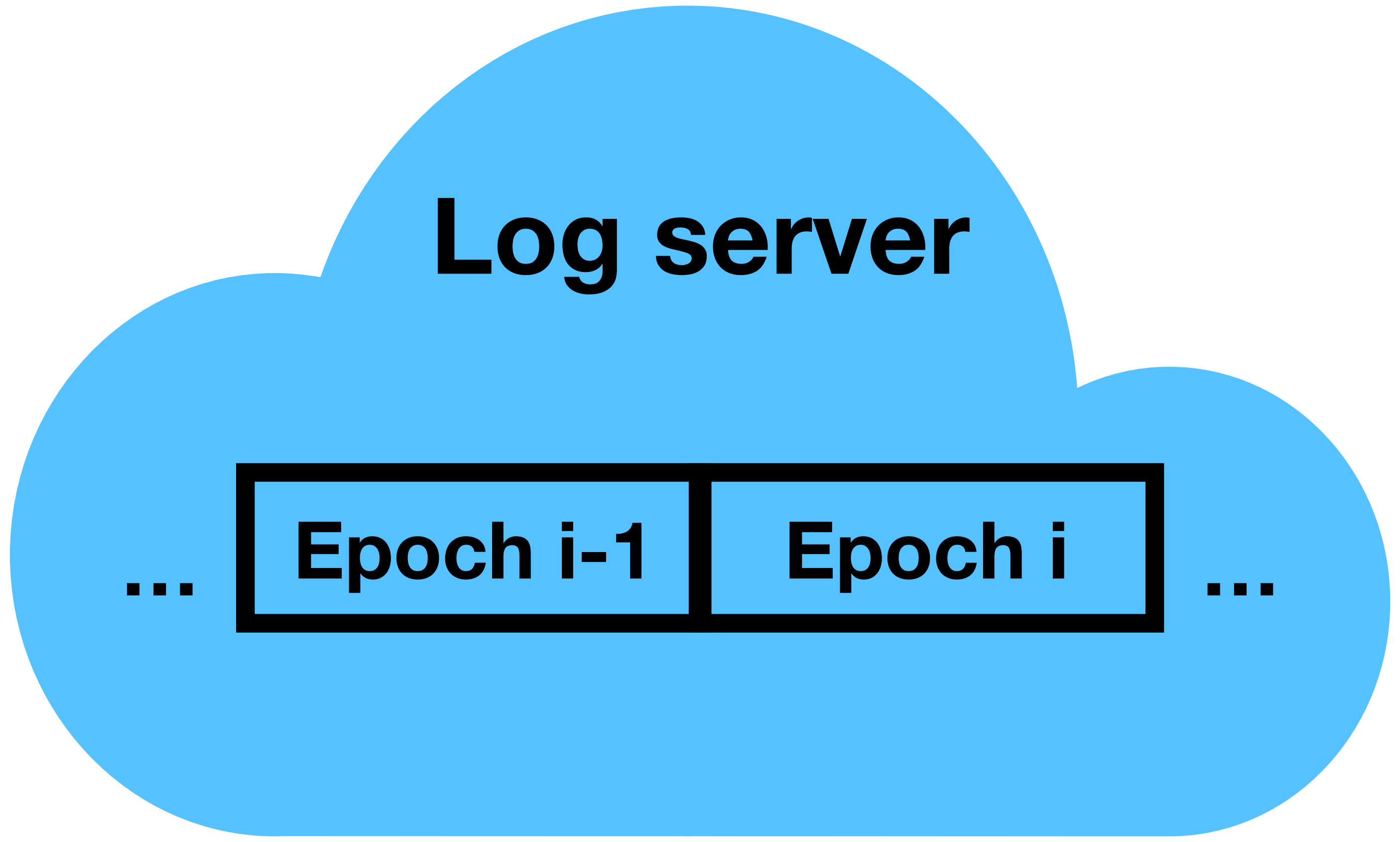
- Log servers: stores certificates in logs
- Auditors: audit the log is append-only
  - Anyone can be an auditor
  - Untrusted except that at least one auditor should be honest and reachable
- Domain owners: monitor their certificates in the log
  - Trusted to monitor their own certificates
- End users: check that certificates appear in the log

Parties

- Log servers: stores certificates in logs
- Auditors: audit the log is append-only
  - Anyone can be an auditor
  - Untrusted except that at least one auditor should be honest and reachable
- Domain owners: monitor their certificates in the log
  - Trusted to monitor their own certificates
- End users: check that certificates appear in the log
  - Trusted to check each certificate that it receives

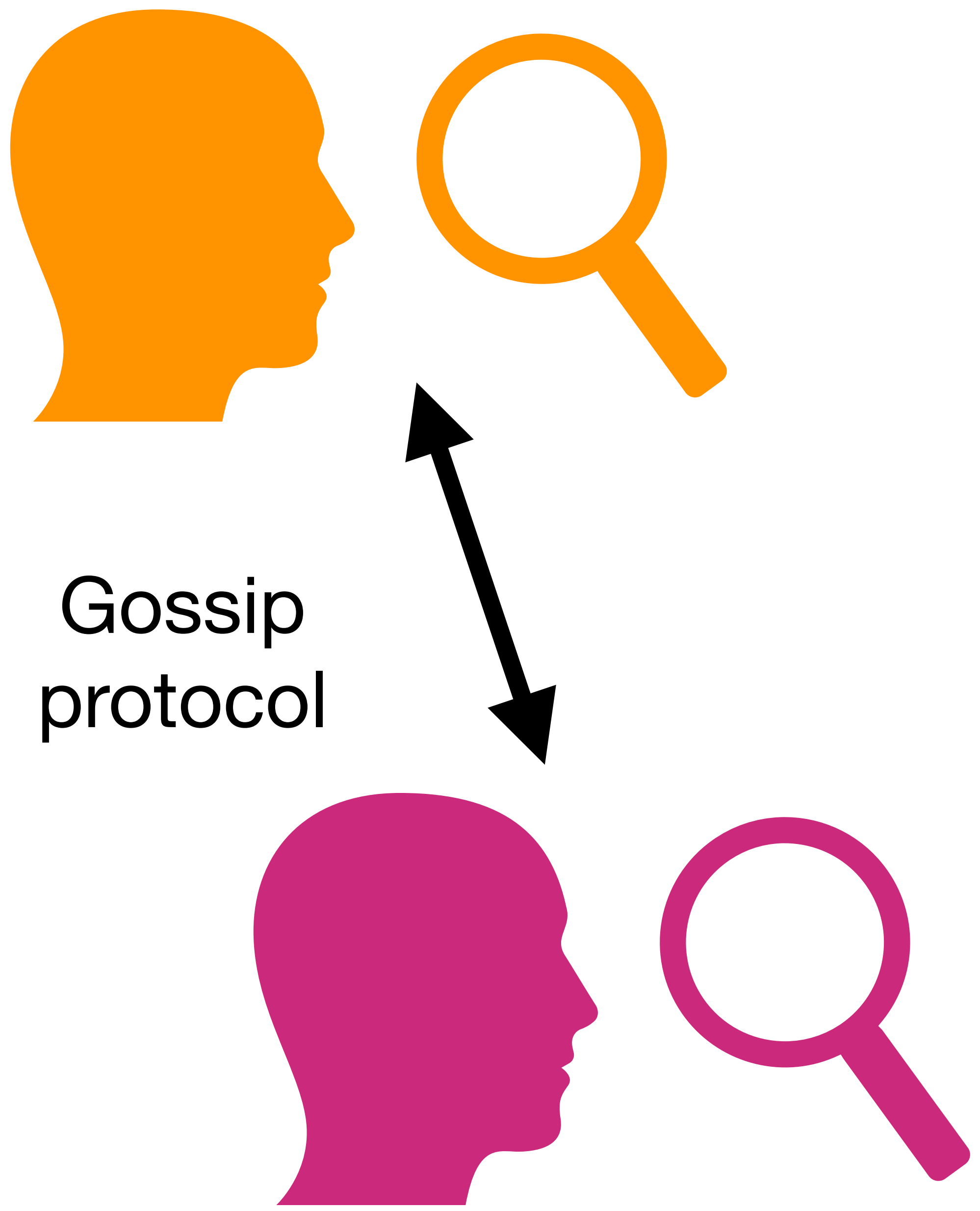
Certificate transparency

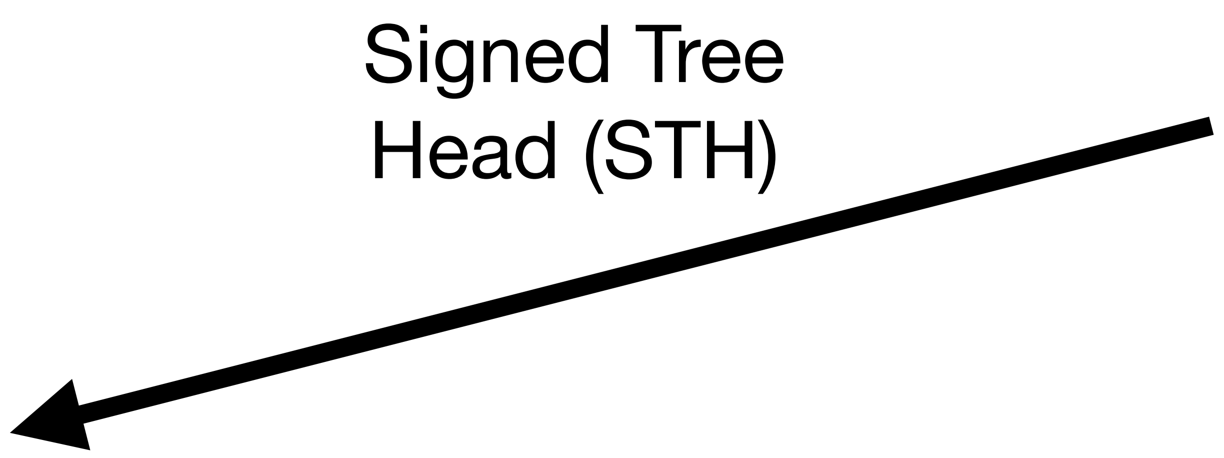
- Log servers store certificates in logs
- Construct Merkle trees over entire logs
- Divide certificates into epochs
- Periodically "checkpoint" and produce summaries of epochs (Signed Tree Head)

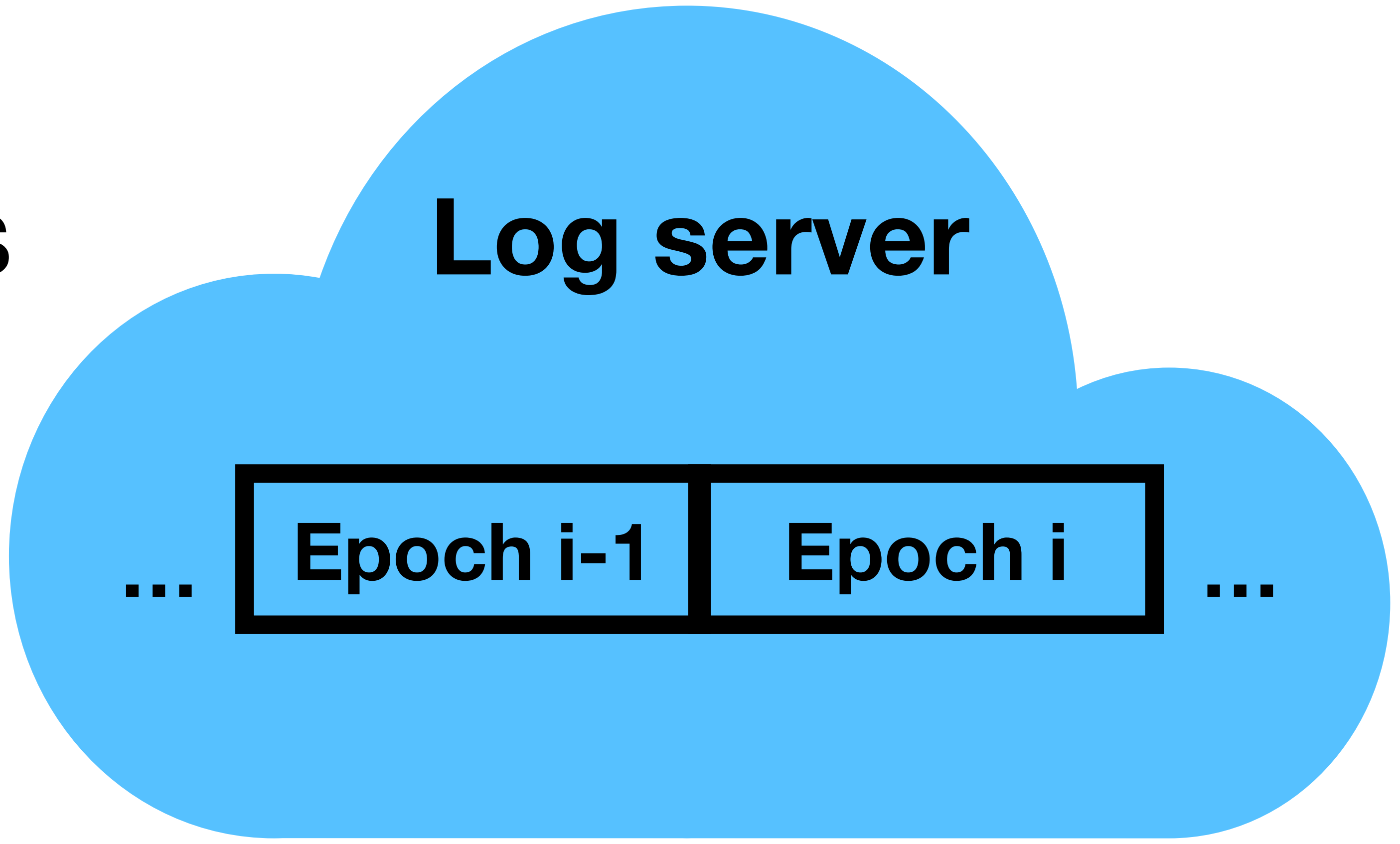


Certificate transparency

- Auditors check for extension proofs of tree nodes
- Auditors check consistency with each other to prevent “split view attacks” via gossip protocols





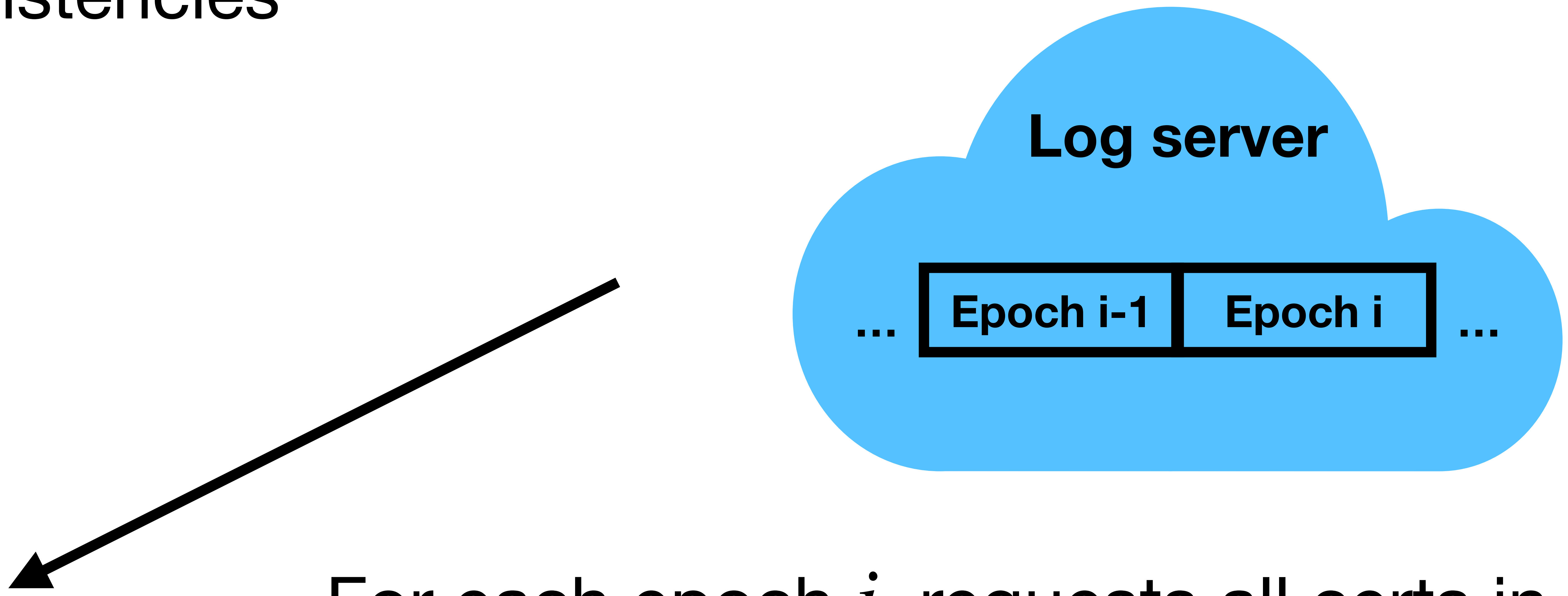


Consistency proof:

- Server proves that H^{i}_{root} is an extension of H^{i-1}_{root}

Certificate transparency

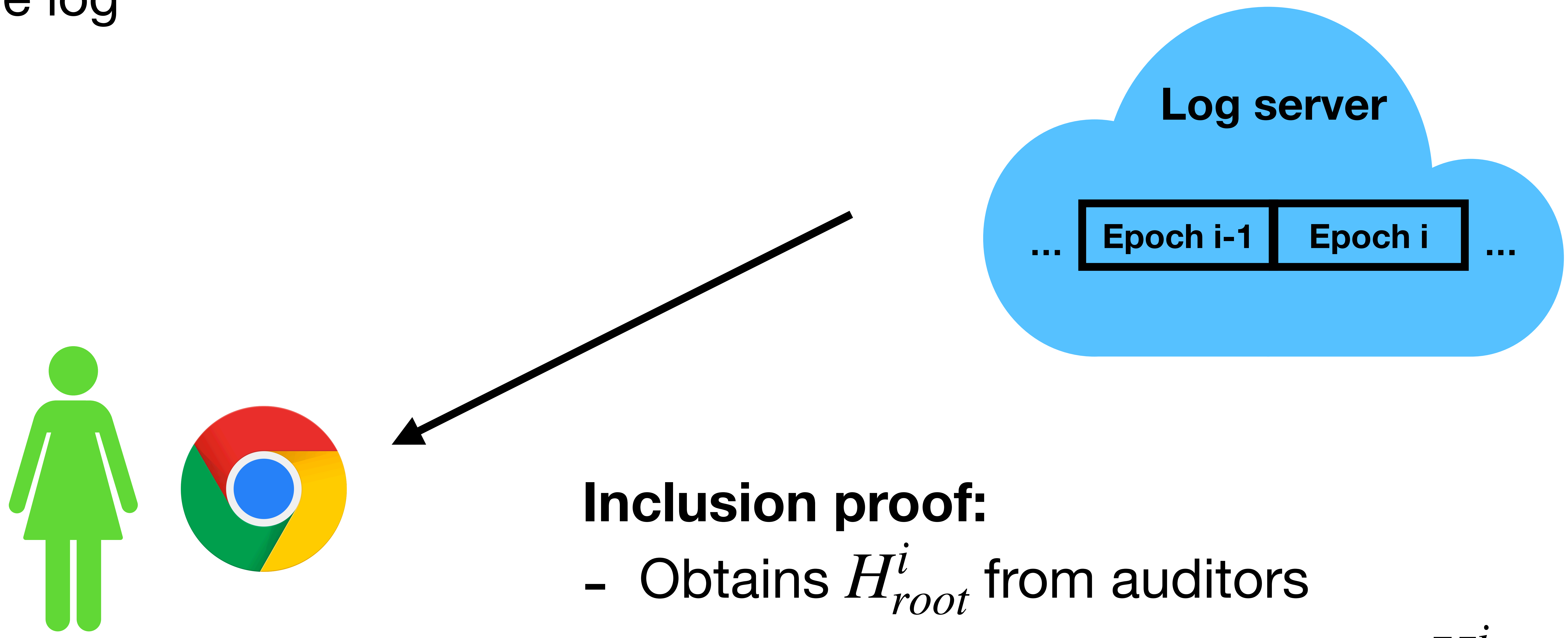
- Domain owners and independent monitors can verify logged certificates and detect inconsistencies



  - For each epoch i, requests all certs in the epoch from the log server
  - Checks them against H_{root}^{i} and H_{root}^{i-1} from the auditors
  - Checks that bank.com's certs are valid

Certificate transparency

- End users check that a website certificate is indeed valid and included in the log



Inclusion proof:

- Obtains H^{i}_{root} from auditors
- Log server proves that cert is in H^{i}_{root} by supplying the Merkle proof

Security analysis

Security analysis

- What if CA is compromised?

Security analysis

- What if CA is compromised?
- What if logs are compromised?

Security analysis

- What if CA is compromised?
- What if logs are compromised?
- What if auditors are compromised?

What’s currently deployed?

What’s currently deployed?

- As of May 2020, CT has publicly logged over 9.2 billion certificates

What’s currently deployed?

- As of May 2020, CT has publicly logged over 9.2 billion certificates
- Chrome requires web certificates issued after April 30, 2018 to appear in a CT log

What's currently deployed?

- As of May 2020, CT has publicly logged over 9.2 billion certificates
- Chrome requires web certificates issued after April 30, 2018 to appear in a CT log
- Taken from Emily Stark's blog post

What's currently deployed?

- As of May 2020, CT has publicly logged over 9.2 billion certificates
- Chrome requires web certificates issued after April 30, 2018 to appear in a CT log
- Taken from Emily Stark's blog post
    - "*Verifying that a given certificate is included in a summary is called SCT verification, and no major web browsers actually do it yet.*"

What’s currently deployed?

- As of May 2020, CT has publicly logged over 9.2 billion certificates
- Chrome requires web certificates issued after April 30, 2018 to appear in a CT log
- Taken from Emily Stark’s blog post
    - “*Verifying that a given certificate is included in a summary is called SCT verification, and no major web browsers actually do it yet.*”
        - Scalability issue: logs cannot handle the load of every end user contacting them for every TLS connection

What’s currently deployed?

- As of May 2020, CT has publicly logged over 9.2 billion certificates
- Chrome requires web certificates issued after April 30, 2018 to appear in a CT log
- Taken from Emily Stark’s blog post
  - “*Verifying that a given certificate is included in a summary is called SCT verification, and no major web browsers actually do it yet.*”
    - Scalability issue: logs cannot handle the load of every end user contacting them for every TLS connection
    - Privacy: inclusion proof reveals to the log which certificate/domain information, which is a violation of user’s privacy

Other resources on certificate transparency

- A Google talk on CT: https://www.youtube.com/watch?v=6PrAVzjZeOI
- RFC for CT: https://datatracker.ietf.org/doc/rfc6962/
- Log proofs: https://sites.google.com/site/certificatetransparency/log-proofs-work
- Challenges in SCT verification: https://www.agwa.name/blog/post/how_will_certificate_transparency_logs_be_audited_in_practice

Course logistics

Course logistics

- Class is MW 10:10 - 11:30 in GHC 4102

Course logistics

- Class is MW 10:10 - 11:30 in GHC 4102
- Office hours are Mondays 2 - 3 in GHC 9015

Course logistics

- Class is MW 10:10 - 11:30 in GHC 4102
- Office hours are Mondays 2 - 3 in GHC 9015
    - Can also email me at wenting@cmu.edu to set up meetings

Course logistics

- Class is MW 10:10 - 11:30 in GHC 4102
- Office hours are Mondays 2 - 3 in GHC 9015
  - Can also email me at wenting@cmu.edu to set up meetings
- Class website at https://wzheng.github.io/15-829/index.html

Course logistics

- Class is MW 10:10 - 11:30 in GHC 4102
- Office hours are Mondays 2 - 3 in GHC 9015
  - Can also email me at wenting@cmu.edu to set up meetings
- Class website at https://wzheng.github.io/15-829/index.html
- Guest speakers

Assignments

- Paper reviews (20%)
- Paper presentations + in-class discussions (30%)
- Research project (50%)

Paper reviews

Paper reviews

- Each class has one core reading

Paper reviews

- Each class has one core reading
- Everyone (except for the presenters) will write a paper review for this reading

Paper reviews

- Each class has one core reading
- Everyone (except for the presenters) will write a paper review for this reading
- Email to me at wenting@cmu.edu the day before lecture by 4 pm

Paper reviews

- Each class has one core reading
- Everyone (except for the presenters) will write a paper review for this reading
- Email to me at wenting@cmu.edu the day before lecture by 4 pm
  - Email title should be “[Paper Review] <date>”

Paper reviews

- Each class has one core reading
- Everyone (except for the presenters) will write a paper review for this reading
- Email to me at wenting@cmu.edu the day before lecture by 4 pm
    - Email title should be “[Paper Review] <date>”
    - Example: *[Paper Review] September 13*

Paper reviews

Paper reviews

- Similar to conference style reviews

Paper reviews

- Similar to conference style reviews
    - Paper summary, strengths, weaknesses, one discussion question

Paper reviews

- Similar to conference style reviews
  - Paper summary, strengths, weaknesses, one discussion question
  - *What important problem is it addressing?*

Paper reviews

- Similar to conference style reviews
  - Paper summary, strengths, weaknesses, one discussion question
  - *What important problem is it addressing?*
  - *What are the techniques used in the paper?*

Paper reviews

- Similar to conference style reviews
  - Paper summary, strengths, weaknesses, one discussion question
  - *What important problem is it addressing?*
  - *What are the techniques used in the paper?*
  - *What are the limitations & future work to be done?*

Paper presentations & discussion

Paper presentations & discussion

- Class presentation

Paper presentations & discussion

- Class presentation
    - Each student will present two classes this semester

Paper presentations & discussion

- Class presentation
  - Each student will present two classes this semester
  - Each class should have two student presenters

Paper presentations & discussion

- Class presentation
  - Each student will present two classes this semester
  - Each class should have two student presenters
  - Need to present the core paper, and may also need to present some optional material

Paper presentations & discussion

- Class presentation
  - Each student will present two classes this semester
  - Each class should have two student presenters
  - Need to present the core paper, and may also need to present some optional material
  - Presentation should focus on answering the previous two questions

Paper presentations & discussion

- Class presentation
  - Each student will present two classes this semester
  - Each class should have two student presenters
  - Need to present the core paper, and may also need to present some optional material
  - Presentation should focus on answering the previous two questions
  - Lead class discussion around the last discussion

Final project

Final project

- Teams of 2 - 3

Final project

- Teams of 2 - 3
    - Individual projects are ok if it’s a related, ongoing research project

Final project

- Teams of 2 - 3
  - Individual projects are ok if it’s a related, ongoing research project
- Should be a relevant topic to the class

Final project

Final project

- Deliverables & deadlines:

Final project

- Deliverables & deadlines:
    - September 20: topic & summary of literature review

Final project

- Deliverables & deadlines:
    - September 20: topic & summary of literature review
    - September 29: 1 - 2 page project proposal due

Final project

- Deliverables & deadlines:
  - September 20: topic & summary of literature review
  - September 29: 1 - 2 page project proposal due
  - Week of October 25: project check in

Final project

- Deliverables & deadlines:
    - September 20: topic & summary of literature review
    - September 29: 1 - 2 page project proposal due
    - Week of October 25: project check in
    - November 29, December 1: project presentations

Final project

- Deliverables & deadlines:
  - September 20: topic & summary of literature review
  - September 29: 1 - 2 page project proposal due
  - Week of October 25: project check in
  - November 29, December 1: project presentations
  - December 10: project write ups due; format is a 6-page, double column workshop paper

Today’s reading

- “How to Read a Paper” by S. Keshav

Today's reading

- "How to Read a Paper" by S. Keshav

Books are not scrolls.
Scrolls must be read like the Torah from one end to the other.
Books are random access – a great innovation over scrolls.
Make use of this innovation! Do NOT feel obliged to read a book from beginning to end.
Permit yourself to open a book and start reading from anywhere.
In the case of mathematics or physics or anything especially hard, try to find something anything that you can understand.
Read what you can.
Write in the margins. (You know how useful that can be.)
Next time you come back to that book, you'll be able to read more.
You can gradually learn extraordinarily hard things this way.

- Manuel Blum