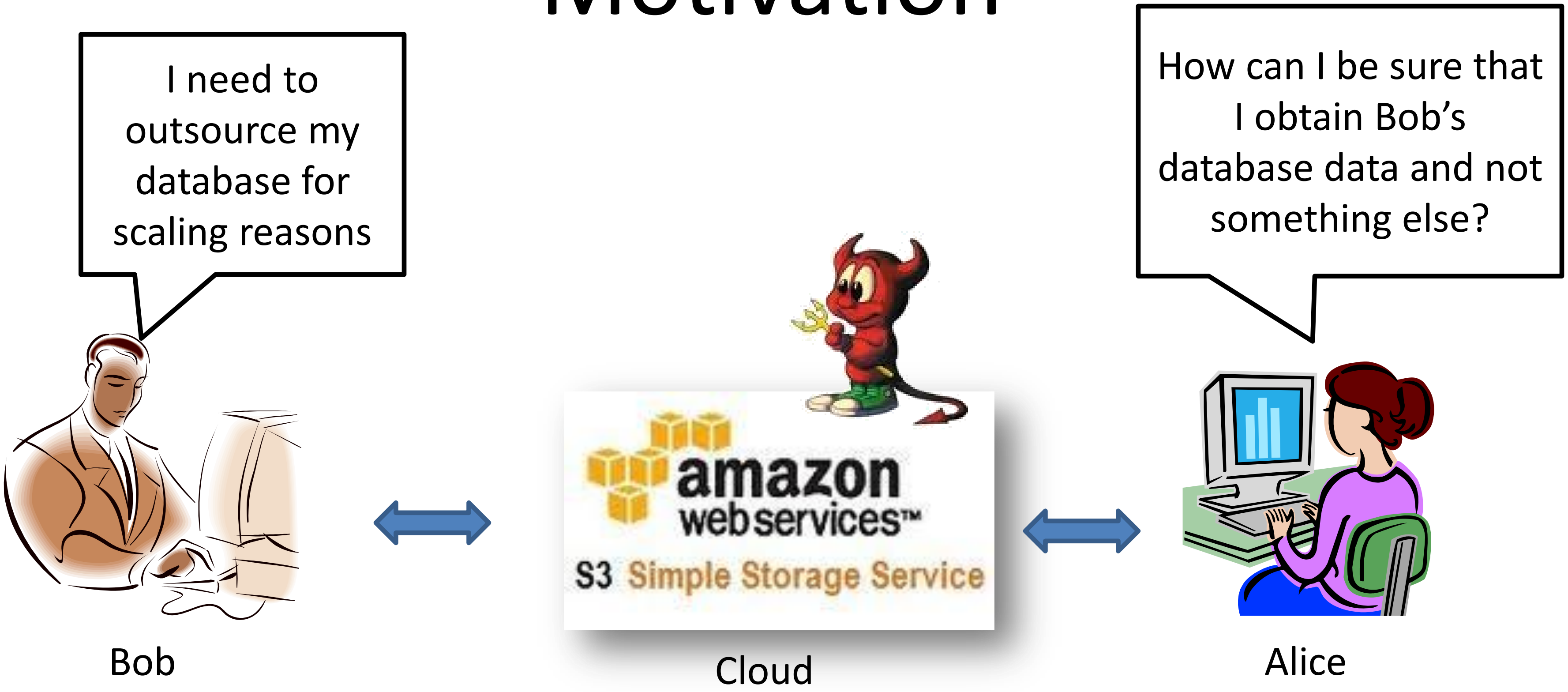
Encuentro de tesistas - 13 of november 2012 – Valparaiso

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Predicate Preserving Collision-Resistant Hashing

Philippe Camacho

Motivation



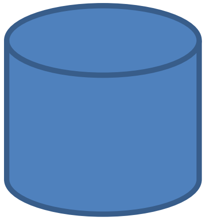
I need to outsource my database for scaling reasons

How can I be sure that I obtain Bob's database data and not something else?

Bob

Cloud

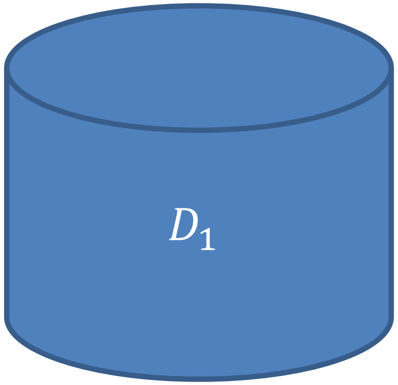
Alice



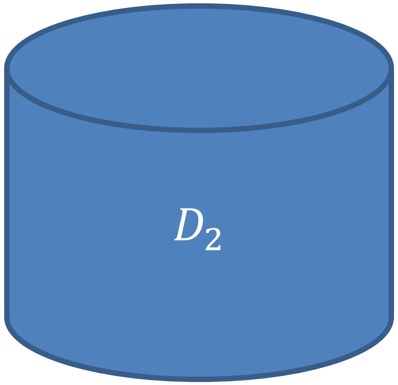
Database

We need to handle a short representation of the database.

Hash Functions (not cryptographic)



How do we check efficiently that the two databases are the same?



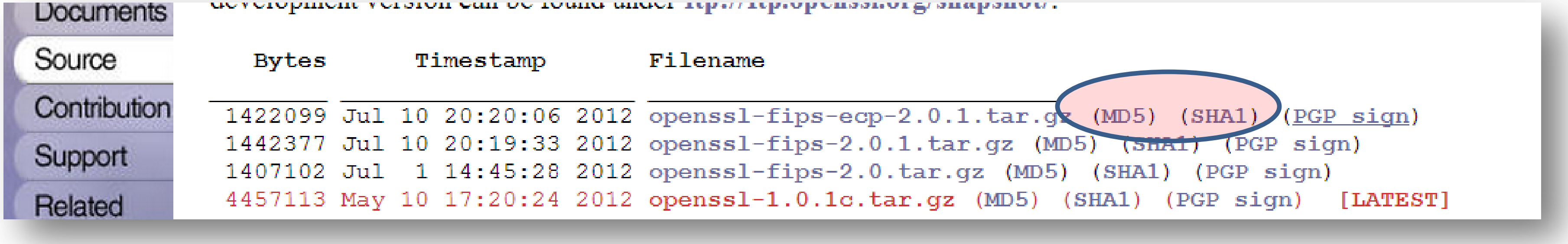
$H(D_1) \xrightarrow{\hspace{6em}}$

$\xleftarrow{\hspace{6em}} H(D_2)$

$H(D_1) = H(D_2)$?

$H(D_1) = H(D_2)$?

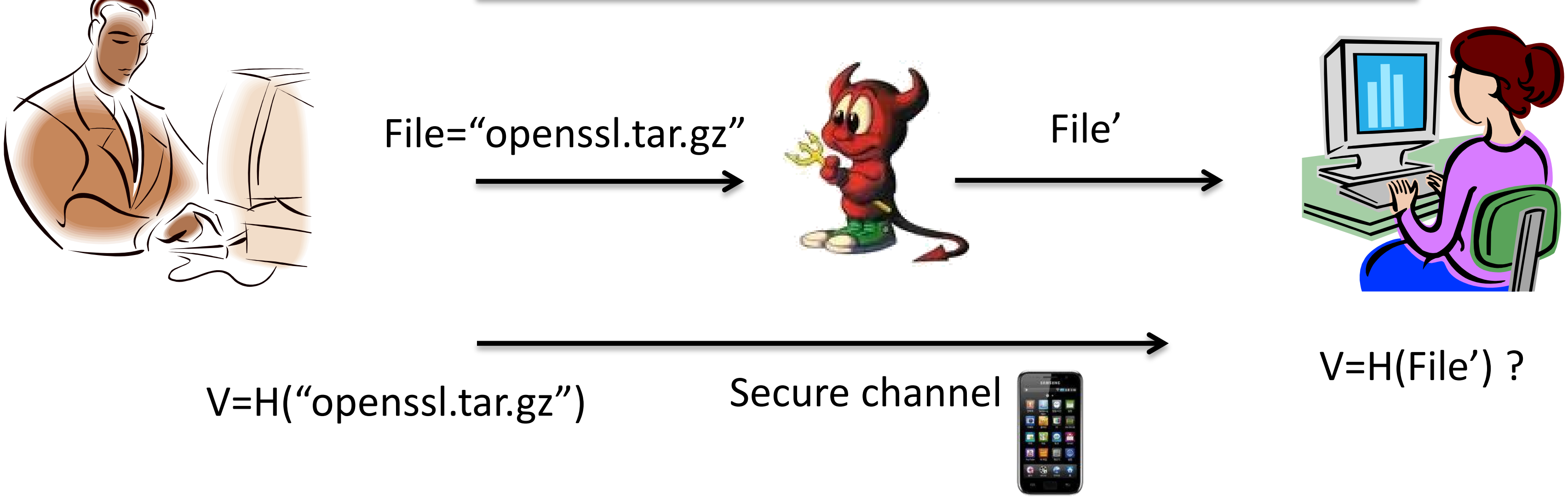
Collision-Resistant Hash Functions



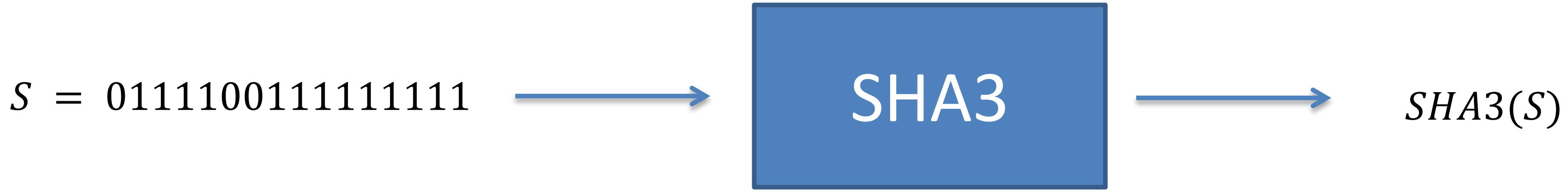
```
development version can be found under ftp://ftp.openssl.org/snapshot/.

   Bytes       Timestamp          Filename
 1422099 Jul 10 20:20:06 2012 openssl-fips-ecp-2.0.1.tar.gz (MD5) (SHA1) (PGP sign)
 1442377 Jul 10 20:19:33 2012 openssl-fips-2.0.1.tar.gz (MD5) (SHA1) (PGP sign)
 1407102 Jul  1 14:45:28 2012 openssl-fips-2.0.tar.gz (MD5) (SHA1) (PGP sign)
 4457113 May 10 17:20:24 2012 openssl-1.0.1c.tar.gz (MD5) (SHA1) (PGP sign)   [LATEST]
```

OpenSSL Source Code:
openssl.tar.gz

The Adversary should not be able to change the file (find a collision) without being detected.



Predicate Preserving Collision-Resistant Hashing



$S = 0111100111111111$

$SHA3(S)$

How to prove efficiently that S …

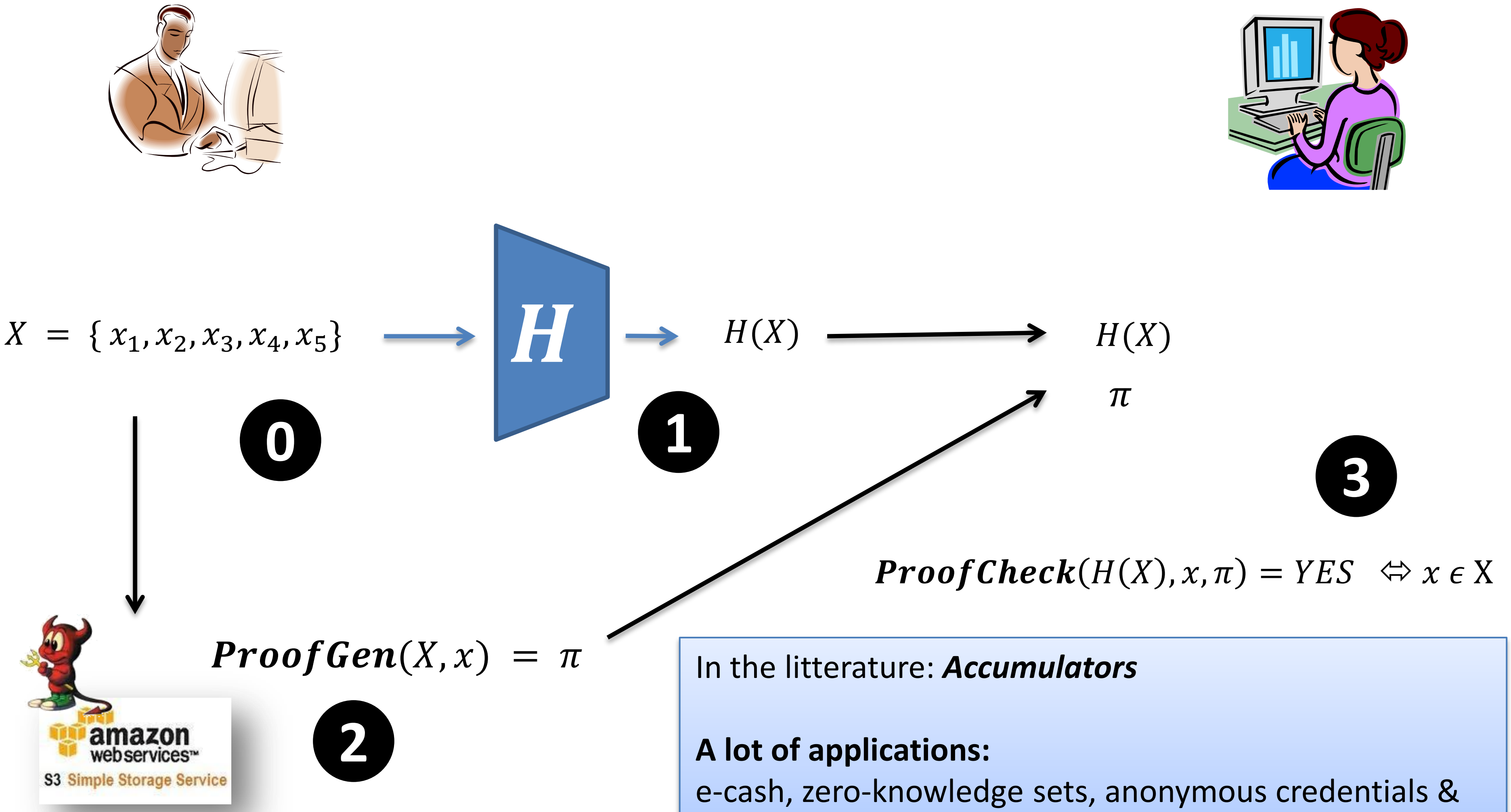
contains a 1 in position 5?

starts with 0111?

contains more 1’s than 0’s ?

…

Predicate: $\mathcal{P}(X, x) = True \Leftrightarrow x \in X$



$X = \{x_1, x_2, x_3, x_4, x_5\}$

$H(X)$

$H(X)$ π

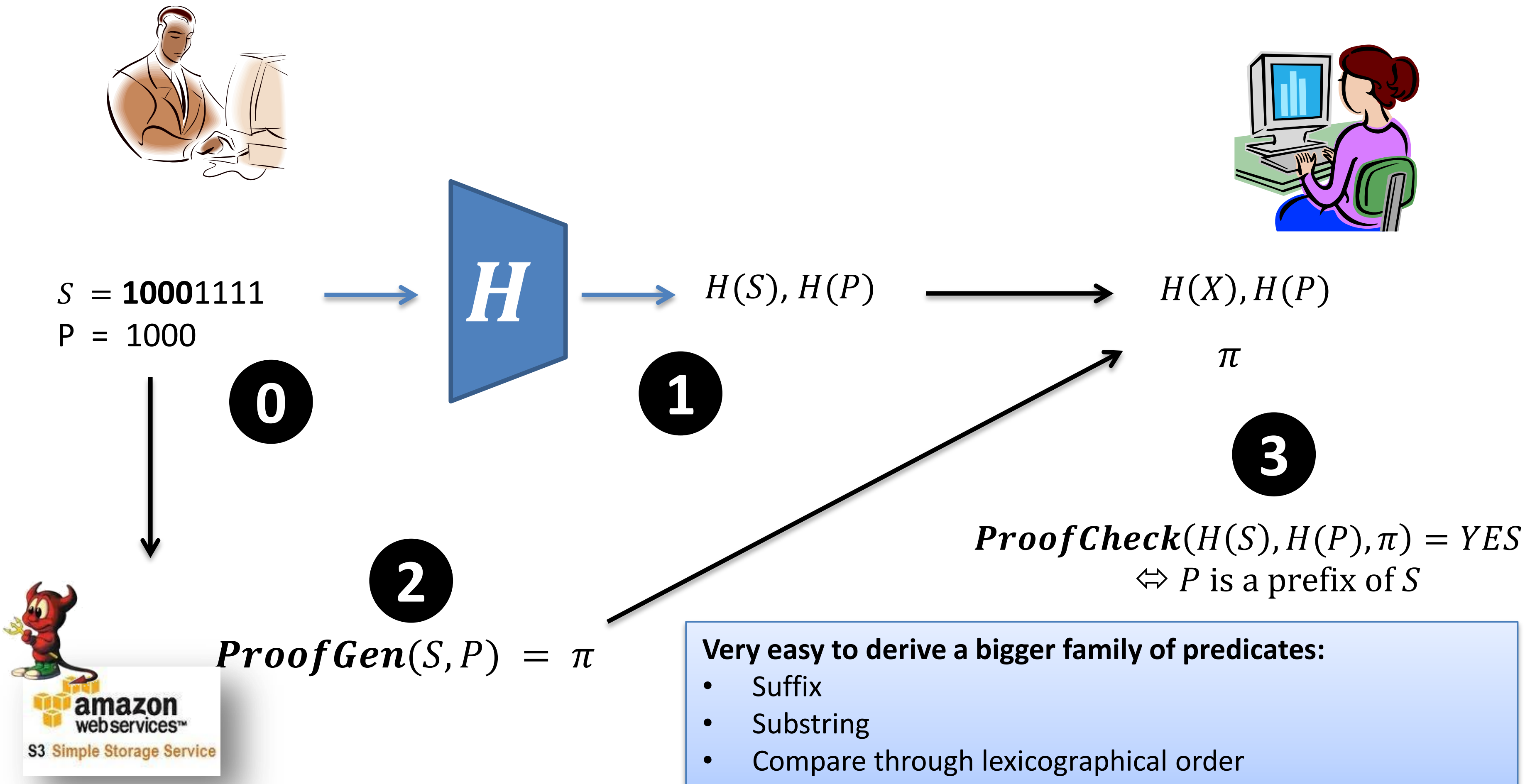
$\boldsymbol{ProofGen}(X, x) = \pi$

$\boldsymbol{ProofCheck}(H(X), x, \pi) = YES \Leftrightarrow x \in \mathrm{X}$

In the litterature: ***Accumulators***

A lot of applications:
e-cash, zero-knowledge sets, anonymous credentials & ring signatures, database authentication, …

Predicate: $\mathcal{P}(S,P) = True$
$\Leftrightarrow P$ is a prefix of S



S = **1000**1111

P = 1000

H

$H(S), H(P)$

$H(X), H(P)$

π

0

1

2

3

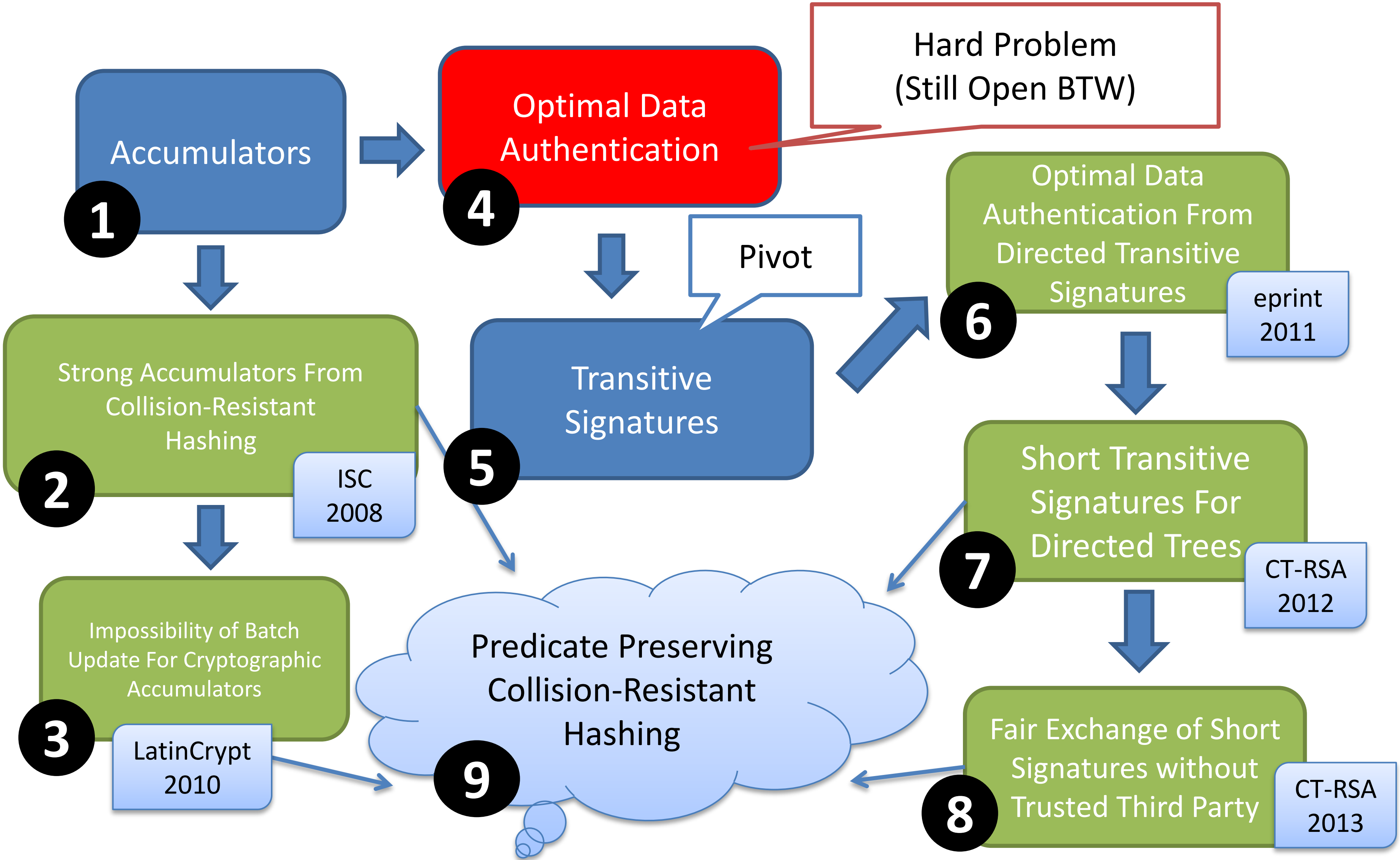
$\boldsymbol{ProofGen}(S,P) = \pi$

$\boldsymbol{ProofCheck}(H(S),H(P),\pi) = YES$
$\Leftrightarrow P$ is a prefix of S

Very easy to derive a bigger family of predicates:

- Suffix
- Substring
- Compare through lexicographical order
- …

Map



1 Accumulators

2 Strong Accumulators From Collision-Resistant Hashing

ISC 2008

3 Impossibility of Batch Update For Cryptographic Accumulators

LatinCrypt 2010

4 Optimal Data Authentication

Hard Problem (Still Open BTW)

5 Transitive Signatures

Pivot

6 Optimal Data Authentication From Directed Transitive Signatures

eprint 2011

7 Short Transitive Signatures For Directed Trees

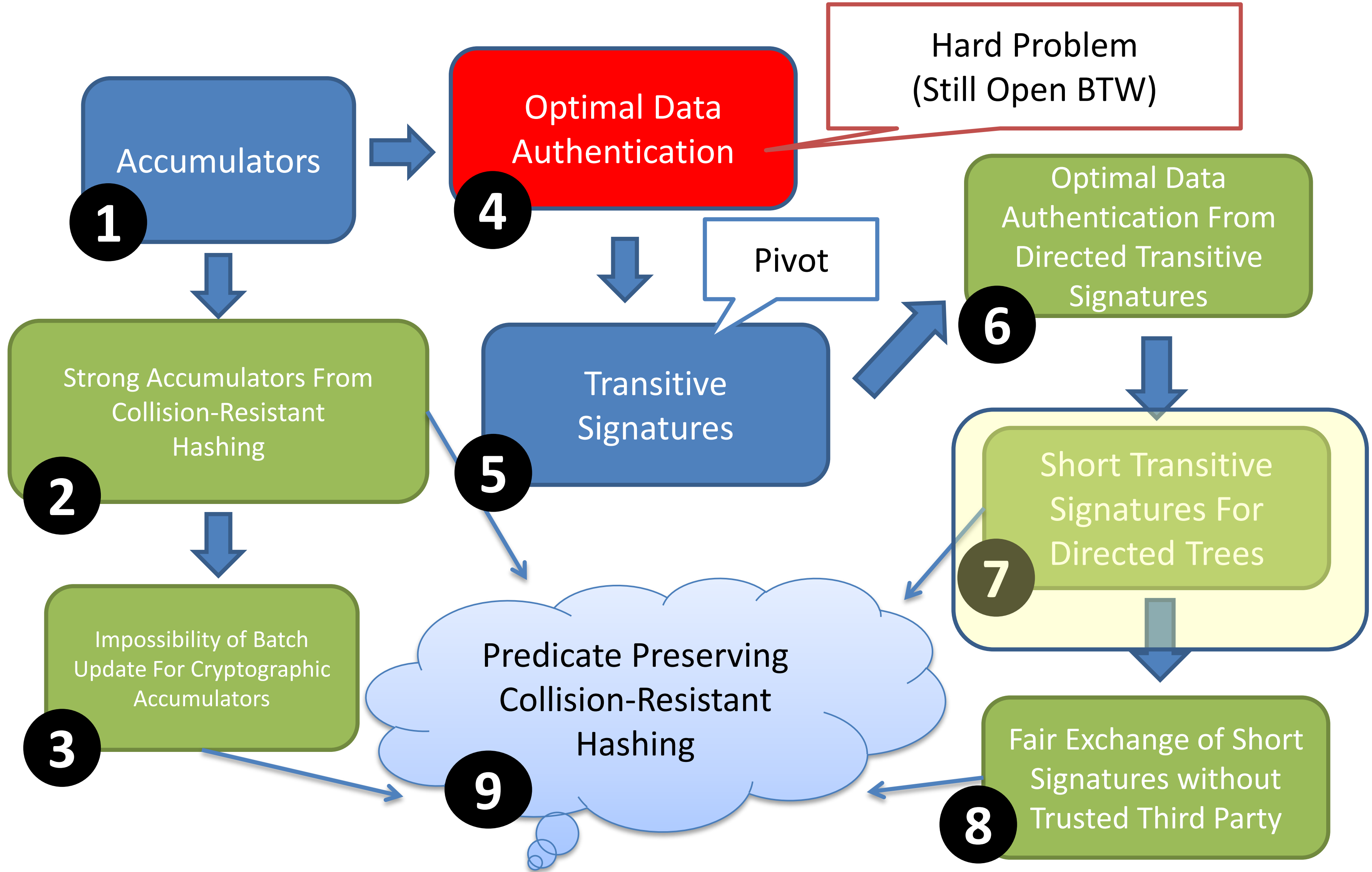
CT-RSA 2012

8 Fair Exchange of Short Signatures without Trusted Third Party

CT-RSA 2013

9 Predicate Preserving Collision-Resistant Hashing

Map



Accumulators

1

Optimal Data Authentication

4

Hard Problem
(Still Open BTW)

Strong Accumulators From Collision-Resistant Hashing

2

Pivot

Transitive Signatures

5

Optimal Data Authentication From Directed Transitive Signatures

6

Short Transitive Signatures For Directed Trees

7

Impossibility of Batch Update For Cryptographic Accumulators

3

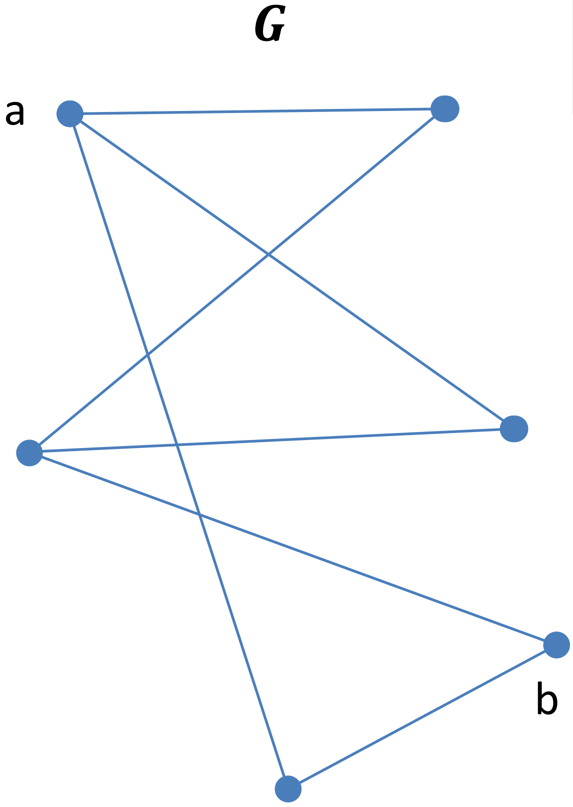
Predicate Preserving Collision-Resistant Hashing

9

Fair Exchange of Short Signatures without Trusted Third Party

8

How do we sign a graph?



Is there a path from a to b?

Trivial solutions

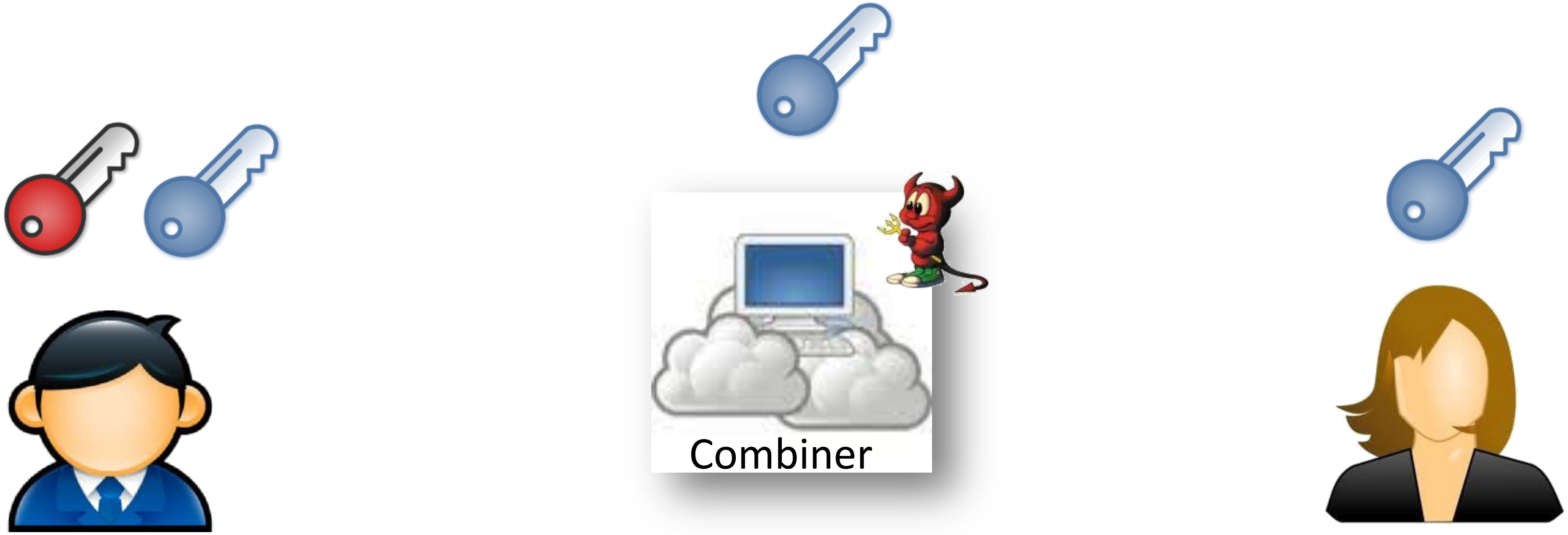
Let $\boldsymbol{n} = |\boldsymbol{G}|$, security parameter $\boldsymbol{\kappa}$

When adding a new node…

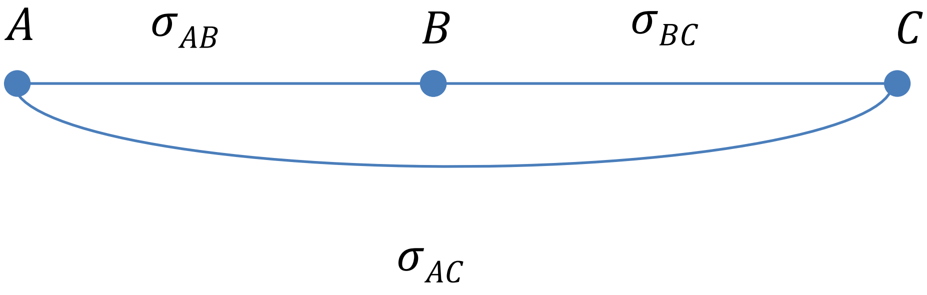
- Sign each edge
  - Time to sign: $\boldsymbol{O(1)}$
  - Size of signature: $\boldsymbol{O(n\kappa)}$ bits

- Sign each path
  - Time to sign (new paths): $\boldsymbol{O(n)}$
  - Size of signature: $\boldsymbol{O(\kappa)}$ bits

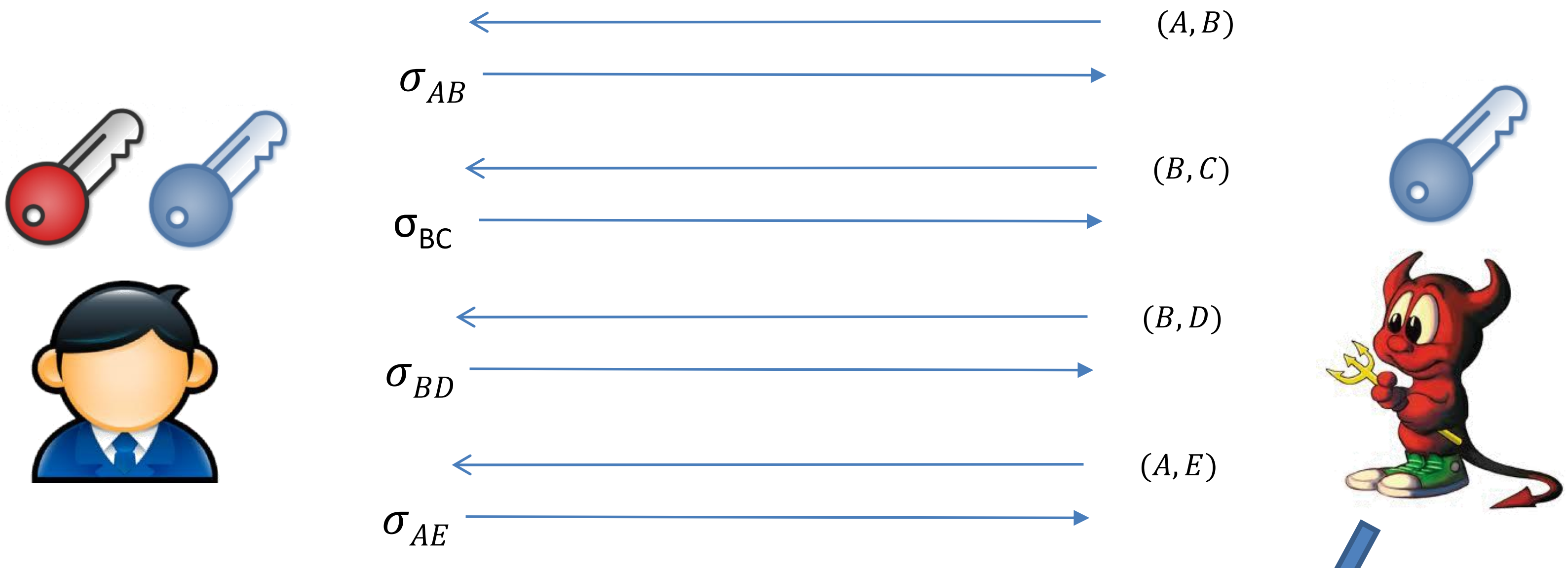
Transitive signature schemes **[MR02,BN05,SMJ05]**

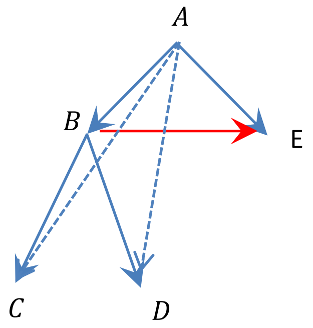


$\sigma_{XY} \leftarrow TSign(X, Y, \sigma_{XY},$ 🔑) $\sigma_{AC} \leftarrow Combine(\sigma_{AB}, \sigma_{BC},$ 🔑) ✓ $\leftarrow TVerify(A, C, \sigma_{AC},$ 🔑)



Security [MR02]





(σ^*, B, E):

✓ $\leftarrow TVerify(B, E, \sigma^*,$ 🔑 $)$ and

There is **no path** from ***B*** to ***E***

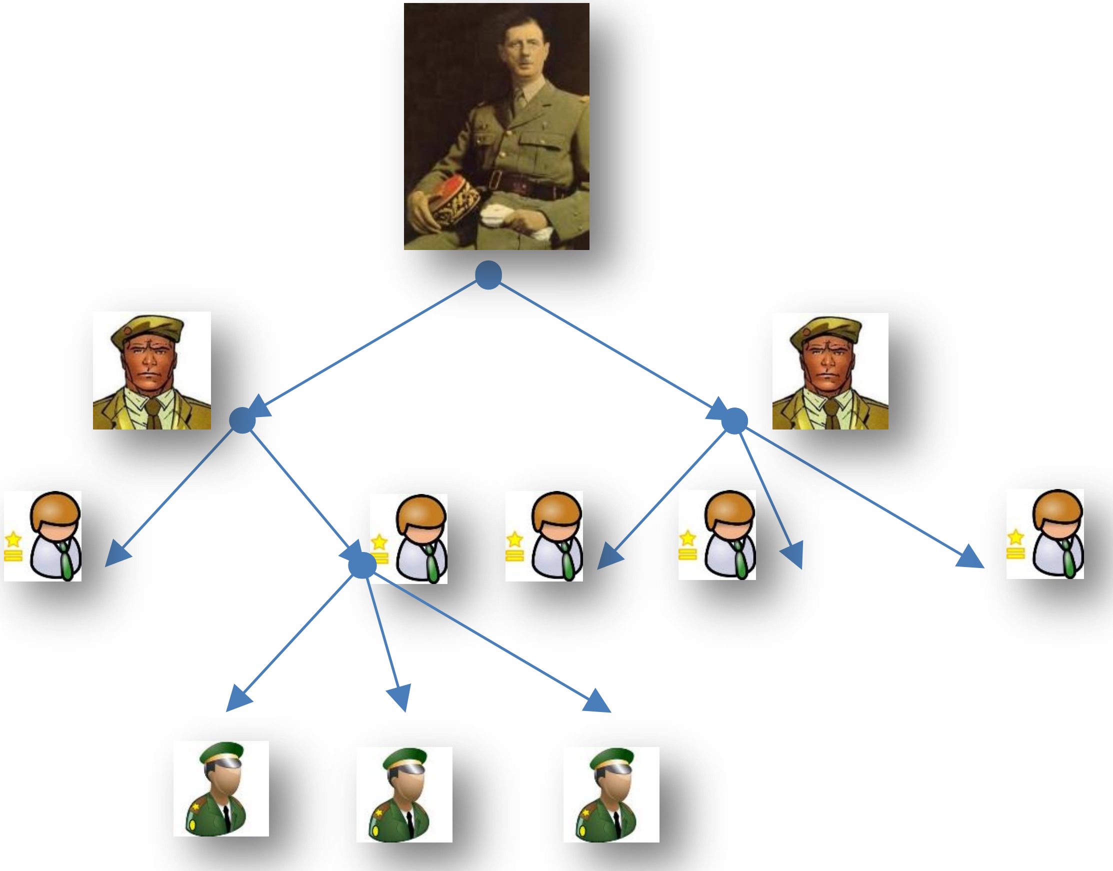
Sounds good, but…

- **[MR02,BN05,SMJ05]**
  for UNDIRECTED graphs ✓

- Transitive Signatures for
  Directed Graphs (DTS) still OPEN ?

- **[Hoh03]**
  DTS ⇒ Trapdoor Groups with
  Infeasible Inversion ⚠

Transitive Signatures for Directed Trees



Previous Work

- **[Yi07]**
  - Signature size: $n \log(n \log n)$ bits
    - Better than $O(n\kappa)$ bits for the trivial solution
  - RSA related assumption

- **[Neven08]**
  - Signature size: $n \log n$ bits
  - Standard Digital Signatures

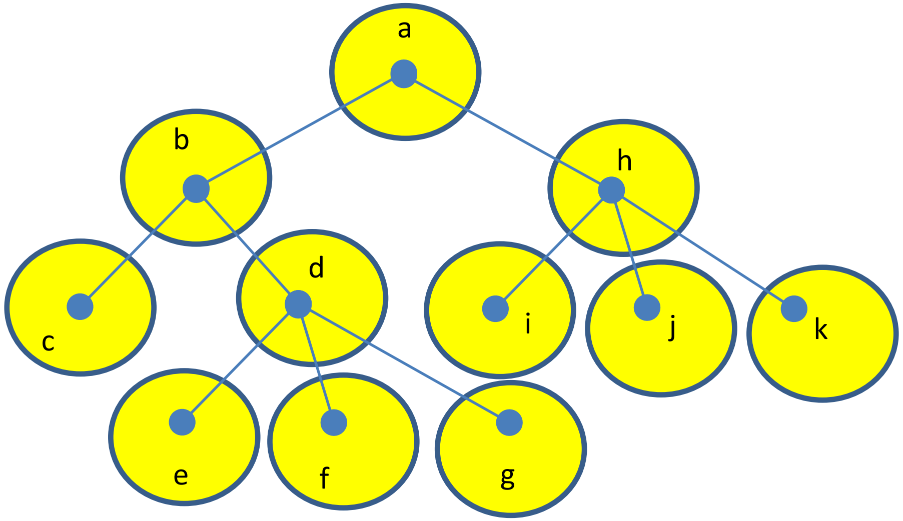
$O(n \log n)$ bits still impractical

Our Results

- For $\boldsymbol{\epsilon} \geq \mathbf{1}$
  - Time to sign edge / verify path signature: $\boldsymbol{O}(\boldsymbol{\epsilon})$
  - Time to compute a path signature: $\boldsymbol{O}(\boldsymbol{\epsilon}(\boldsymbol{n}/\boldsymbol{\kappa})^{1/\epsilon})$
  - Size of path signature: $\boldsymbol{O}(\boldsymbol{\epsilon\kappa})$ bits

Examples	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = \log(\boldsymbol{n})$
Time to sign edge / verify path signature	$\boldsymbol{O}(\mathbf{1})$	$\boldsymbol{O}(\mathbf{1})$	$\boldsymbol{O}(\log \boldsymbol{n})$
Time to compute a path signature	$\boldsymbol{O}(\boldsymbol{n}/\boldsymbol{\kappa})$	$\boldsymbol{O}(\sqrt{\boldsymbol{n}/\boldsymbol{\kappa}})$	$\boldsymbol{O}(\log \boldsymbol{n})$
Size of path signature	$\boldsymbol{O}(\boldsymbol{\kappa})$	$\boldsymbol{O}(\boldsymbol{\kappa})$	$\boldsymbol{O}(\boldsymbol{\kappa} \log \boldsymbol{n})$

Pre/Post Order Tree Traversal



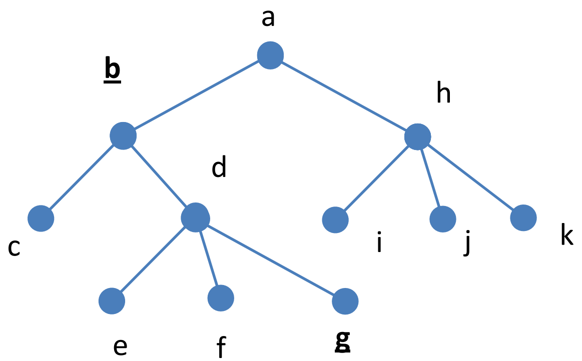
Pre order: a b c d e f g h i j k

Post order: c e f g d b i j k h a

Property of Pre/Post order Traversal

- **Proposition [Dietz82]**

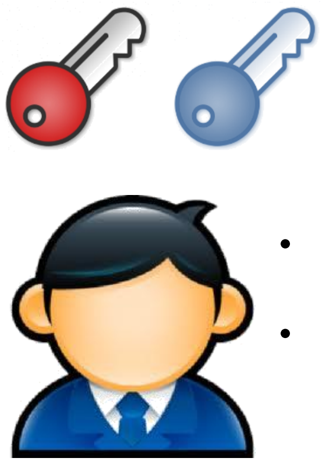
There is a path from $\boldsymbol{x}$ to $\boldsymbol{y}$ $\Leftrightarrow$ $\boldsymbol{pos}(\boldsymbol{x}) < \boldsymbol{pos}(\boldsymbol{y})$ in $\boldsymbol{Pre}$ $\boldsymbol{pos}(\boldsymbol{y}) < \boldsymbol{pos}(\boldsymbol{x})$ in $\boldsymbol{Post}$



Pre order: a **b** c d e f **g** h i j k

Post order: c e f **g** d **b** i j k h a

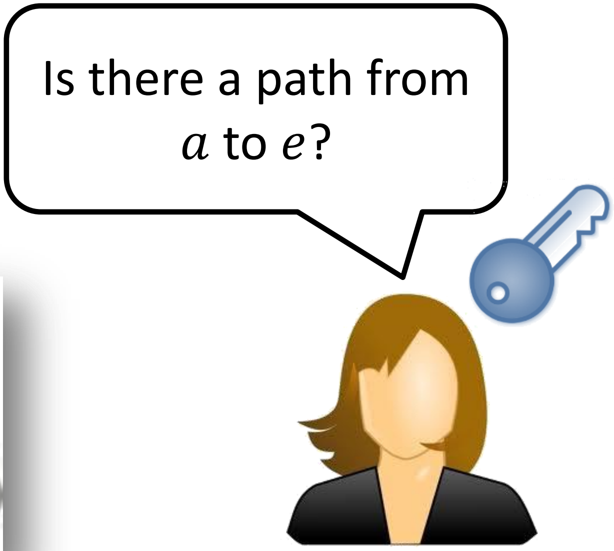
Idea



- Compute $\boldsymbol{pos}(\boldsymbol{x})$ in $\boldsymbol{Pre}$ and $\boldsymbol{Post}$
- E.g.: Sign $\boldsymbol{a}||\mathbf{1}||\mathbf{10}$

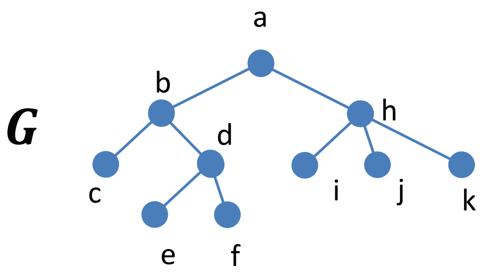
Signature of path $(\boldsymbol{a}, \boldsymbol{e})$:

- Signature of $\boldsymbol{a}||\mathbf{1}||\mathbf{10}$
- Signature of $\boldsymbol{e}||\mathbf{5}||\mathbf{2}$



- Check signatures
- Check

$$\mathbf{1} < \mathbf{5}$$
$$\mathbf{10} > \mathbf{2}$$



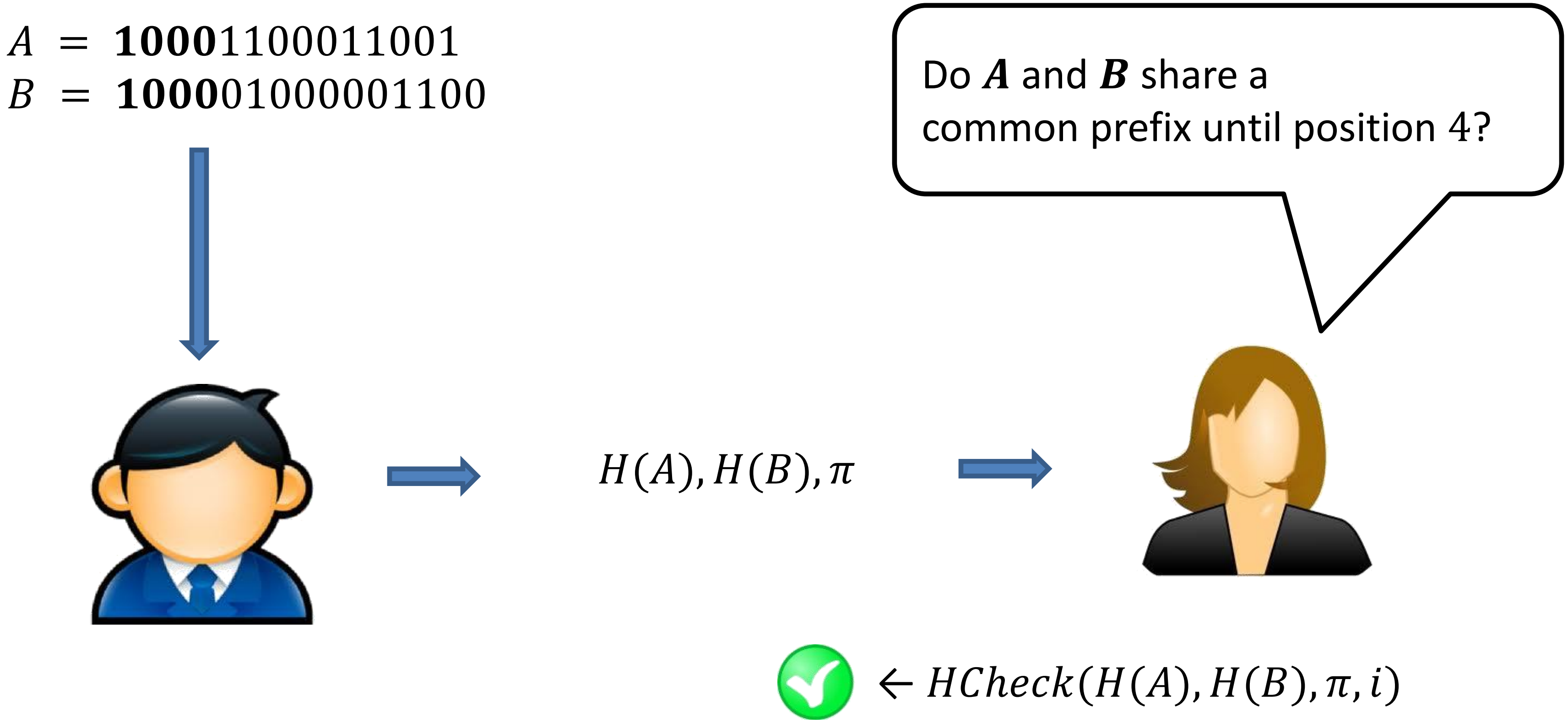
Position	1	2	3	4	5	6	7	8	9	10
Pre	a	b	c	d	e	f	h	i	j	k
Post	c	e	f	d	b	i	j	k	h	a

Challenge: handle changes.

Intuition: tricks to assign labels to the vertices so that these labels do not change.

Remaining task: compare efficiently large labels.

Idea



$A = \mathbf{1000}1100011001$
$B = \mathbf{1000}01000001100$

Do $\boldsymbol{A}$ and $\boldsymbol{B}$ share a common prefix until position 4?

$H(A), H(B), \pi$

✓ $\leftarrow HCheck(H(A), H(B), \pi, i)$

We want:

$\boldsymbol{H}$ collision resistant hash function + proofs

Security

$$\boldsymbol{HGen(1^{\kappa}, n) \rightarrow PK} \quad \Rightarrow$$

$$\Rightarrow \quad \boldsymbol{(A, B, i, \pi)}$$

$$\boldsymbol{Adv(A) = \Pr\begin{bmatrix} HCheck(H(A), H(B), \pi, i, PK) = \ True \\ \wedge \\ A[1..i] \neq B[1..i] \end{bmatrix}}$$

Bilinear maps (pairings)

- $(p, e, G, G_T, g) \leftarrow BMGen(1^k)$

- $|G| = |G_T| = p$
- $e: G \times G \rightarrow G_T$
- $e(g^a, g^b) = e(g, g)^{ab}$
- $e(g, g)$ generates G_T

AMAZING TOOL:

- Started in 2001
- Thousands of publications
- Dedicated Conference (Pairings)

n-BDHI assumption [BB04]

$\boldsymbol{e}: \boldsymbol{G} \times \boldsymbol{G} \rightarrow \boldsymbol{G_T}$

$\boldsymbol{s} \leftarrow \boldsymbol{Z_p}$

$\boldsymbol{g}$ generator of $\boldsymbol{G}$

$(\boldsymbol{g^s}, \boldsymbol{g^{s^2}}, \dots, \boldsymbol{g^{s^n}})$

$\boldsymbol{e(g,g)^{1/s}}$

The hash function

- $\boldsymbol{HGen(1^{\kappa}, n)}$

  $(\boldsymbol{p}, \boldsymbol{G}, \boldsymbol{G_T}, \boldsymbol{e}, \boldsymbol{g}) \leftarrow BMGen(1^{\kappa})$

  $\boldsymbol{s} \leftarrow \boldsymbol{Z_p}$
  $\boldsymbol{T} := (\boldsymbol{g^s}, \boldsymbol{g^{s^2}}, \ldots, \boldsymbol{g^{s^n}})$

  return $\boldsymbol{PK} := (\boldsymbol{p}, \boldsymbol{G}, \boldsymbol{G_T}, \boldsymbol{e}, \boldsymbol{g}, \boldsymbol{T})$

- $\boldsymbol{HEval(M, PK)}$

$$\boldsymbol{H(M)} := \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{g^{M[i]s^i}}$$

Toy example: $\boldsymbol{M = 1001 \Rightarrow H(M) = g^s . g^{s^4}}$

Generating & Verifying Proofs

- $A = A[1..n] = 1000111001$
- $B = B[1..n] = 1000101100$

- $\Delta := \frac{H(A)}{H(B)} = \frac{g^{s}g^{s^5}g^{s^6}g^{s^7}g^{s^{10}}}{g^{s}g^{s^5}g^{s^7}g^{s^8}} = g^{s^6}\, g^{-s^8}\, g^{s^{10}}$

- $\Delta = \prod_{j=1}^{n} g^{C[j]s^j}$ with $C = [0, 0, 0, 0, 0, 1, 0, -1, 0, 1]$

Generating & Verifying Proofs

- $\boldsymbol{\Delta} = \prod_{j=1}^{n} g^{C[j]s^j}$ with $\boldsymbol{C} = [0, 0, 0, 0, 0, 1, 0, -1, 0, 1]$

- “Remove” factor $\mathbf{s}^{\mathbf{i+1}}$ in the exponent **without knowing $\mathbf{s}$**

$$\boldsymbol{\pi} := \boldsymbol{\Delta}^{\frac{1}{s^{i+1}}} = \prod_{j=i+1}^{n} g^{C[j]s^{j-i-1}} = g\, g^{-s^2} g^{s^4}$$

- Check the proof : $e(\boldsymbol{\pi}, g^{s^{i+1}}) = e(\boldsymbol{\Delta}, g)$

Security [CH12]

- **Proposition:**
  If the n-BDHI assumption holds then the previous construction is a CRHF that preserves the prefix predicate.

- Proof (idea)

$$\mathbf{A} = 100010$$
$$\mathbf{B} = 101001$$
$$\mathbf{i} = 3$$

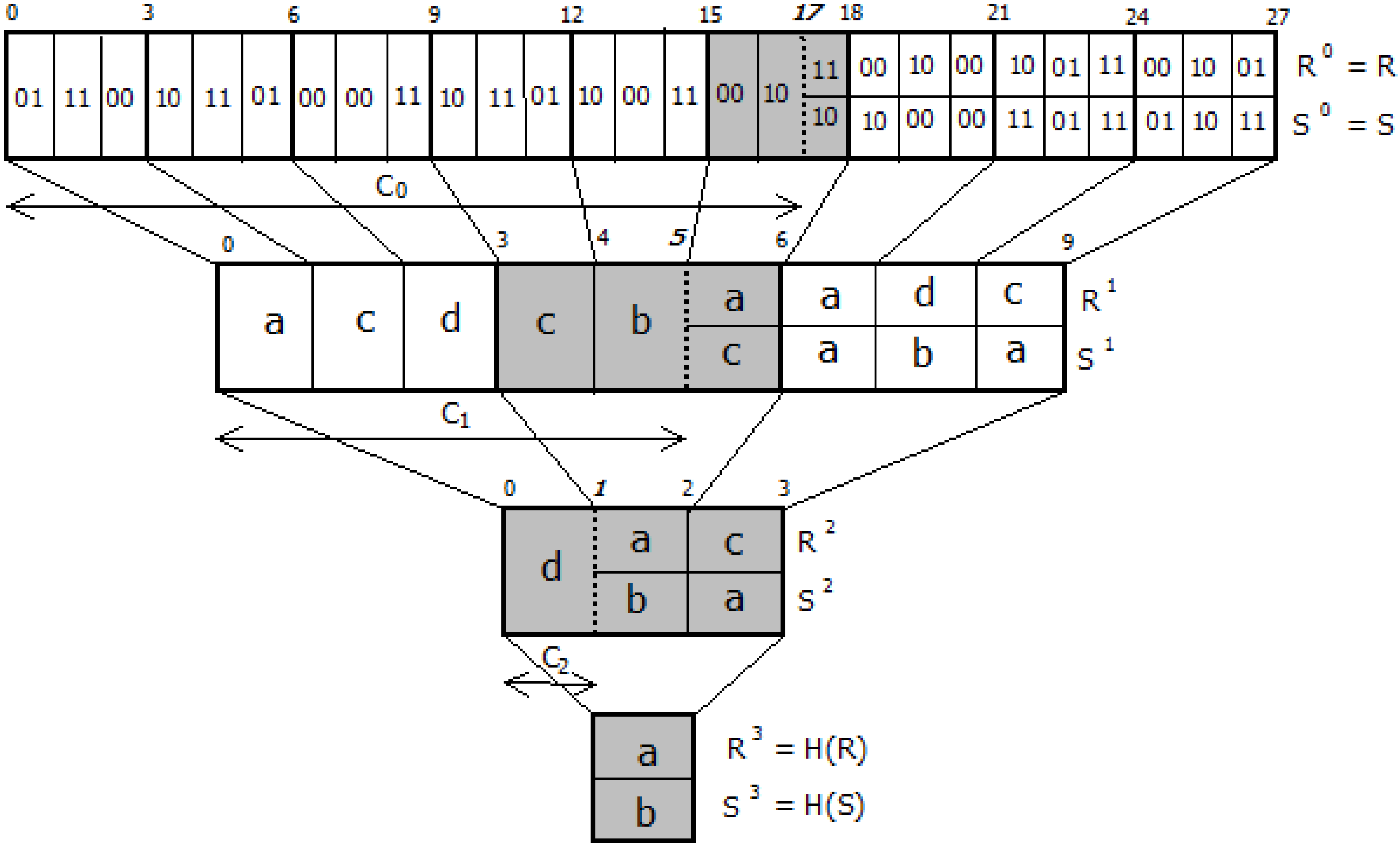
$$\mathbf{H(A)} = \mathbf{g}^{s}\,\mathbf{g}^{s^5}$$
$$\mathbf{H(B)} = \mathbf{g}^{s}\,\mathbf{g}^{s^3}\,\mathbf{g}^{s^6}$$
$$\boldsymbol{\Delta} = \frac{\mathbf{H(A)}}{\mathbf{H(B)}} = \mathbf{g}^{-s^3}\,\mathbf{g}^{s^5}\,g^{-s^6}$$
$$\boldsymbol{\pi} = \boldsymbol{\Delta}^{\frac{1}{s^4}} = \mathbf{g}^{-1/s}\,\mathbf{g}^{s}\,\mathbf{g}^{-s^2}$$

Trade off

$n = 54, \qquad \kappa = 2, \qquad \Sigma = \{a, b, c, d\}$

$n/\kappa = 54/2 = 27$

$\lambda = 3 \Rightarrow (n/\kappa)^{1/\lambda} = 3$



Conclusion

- We introduced the concept of Predicate Preserving Collision-Resistant Hashing
- Many open questions
  - Optimal Data Authentication
  - Relationship between predicate complexity and size for proofs
  - Apply these techniques to authenticated pattern matching
  - Find new applications…

Thank you!