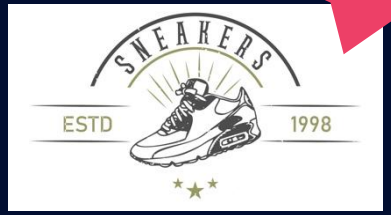
Bandwidth Efficient Partial Authorized PSI

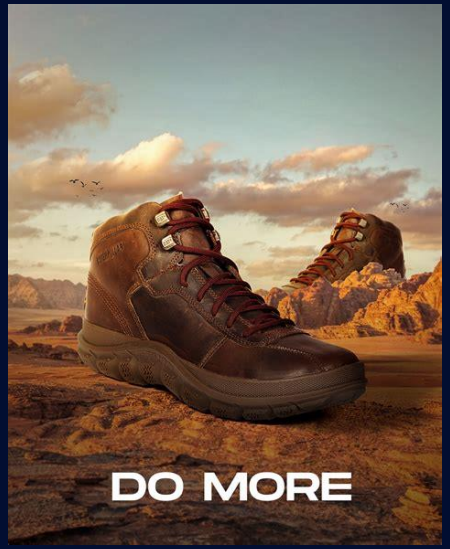
Tjitske Koster, Francesca Falzon, Lilika Markatou

TU Delft Delft University of Technology

Advertisements

We want to advertise shoes…

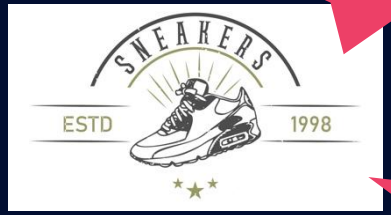




We'll sell you adds!

Advertisement strategy

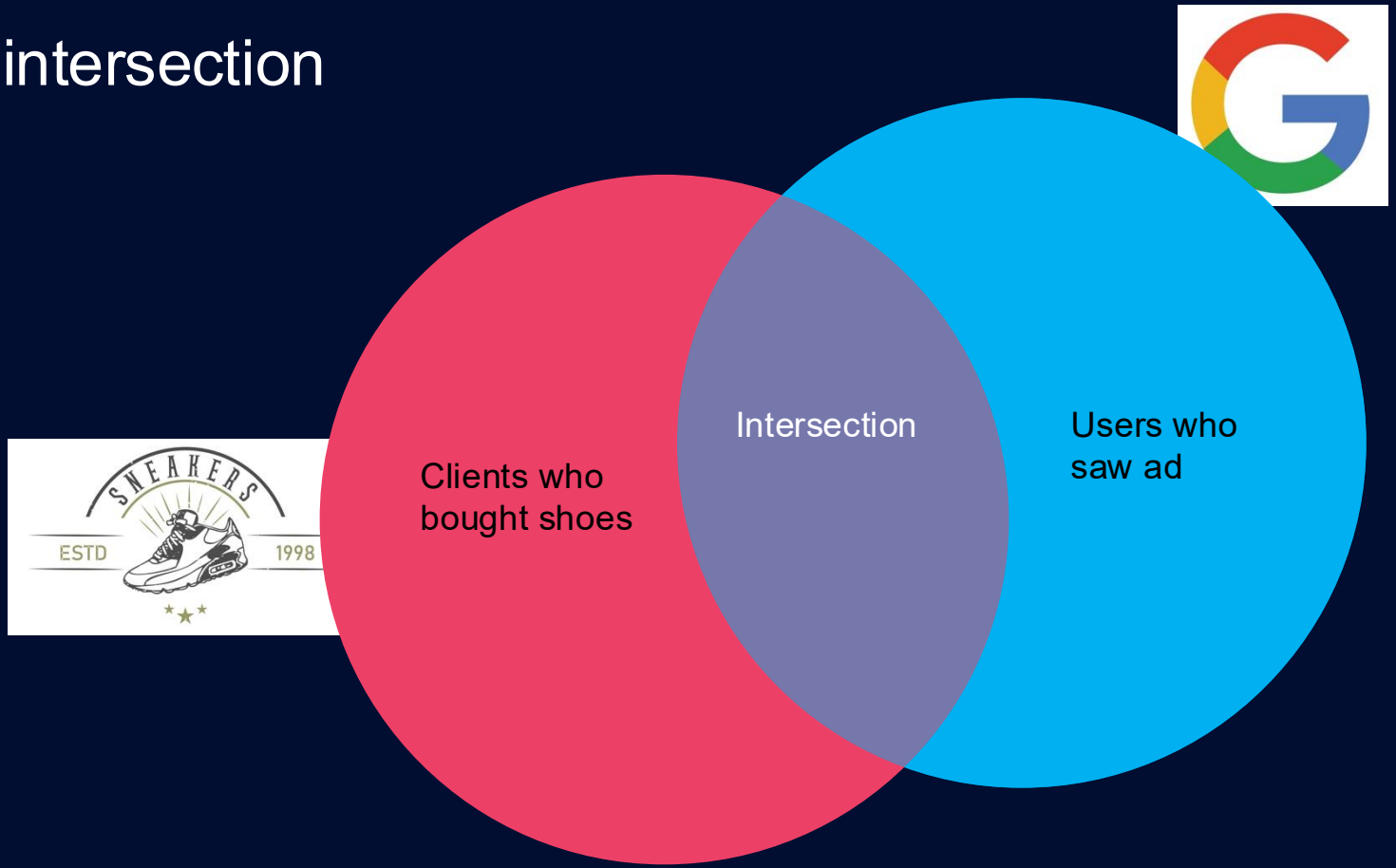
We sold shoes, but did they see the ad?

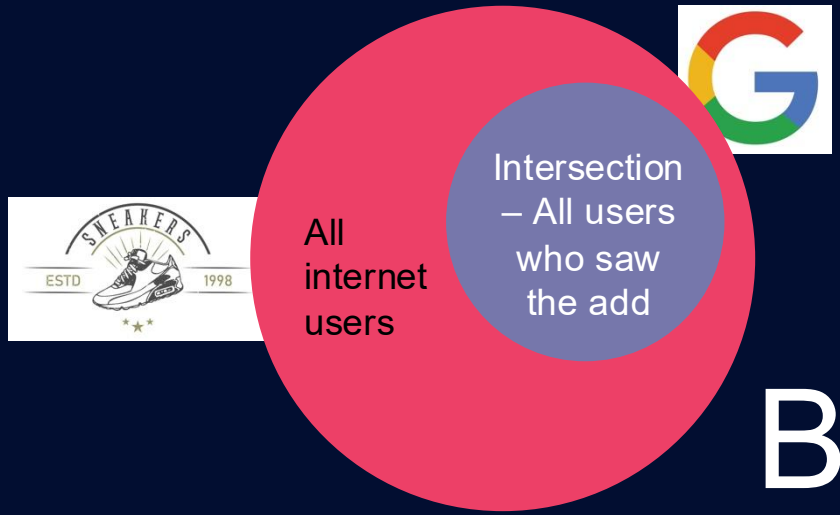


What if…

If you send your clients, I'll tell you!

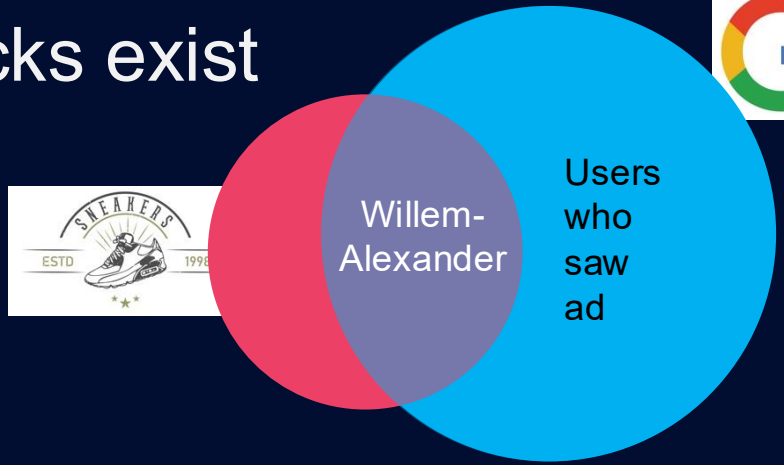
Private set intersection

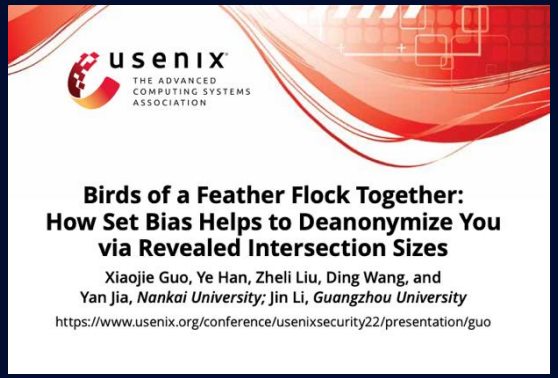




But…

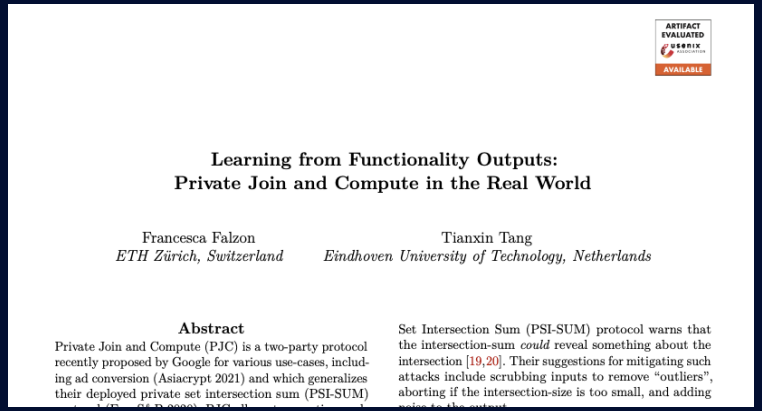
Many attacks exist

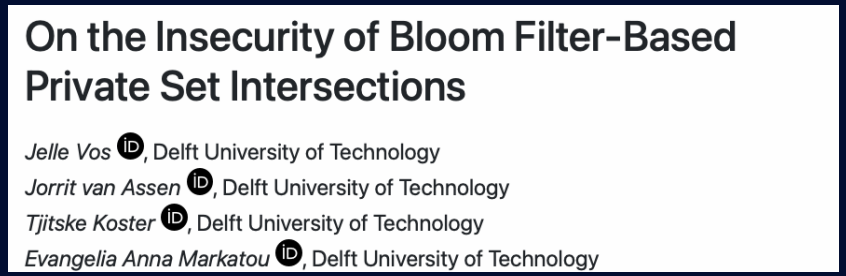




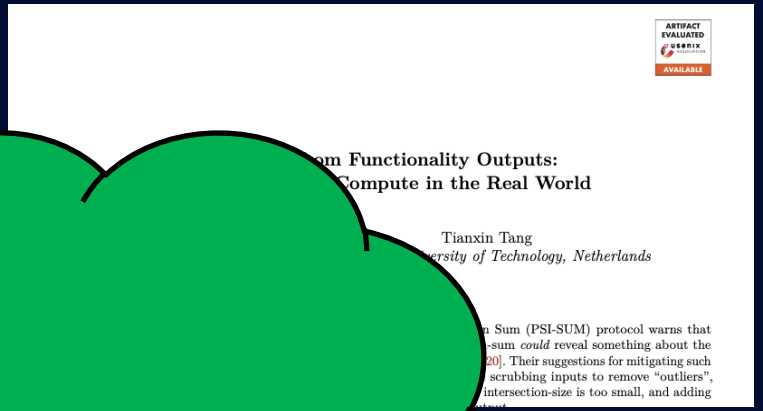
But…

Many attacks exist

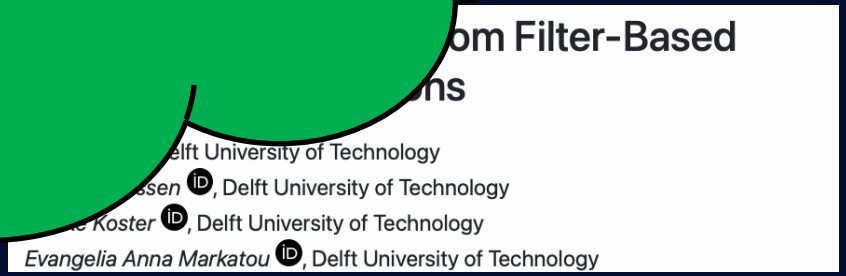




TU Delft Delft University of Technology



Introduce a Judge!

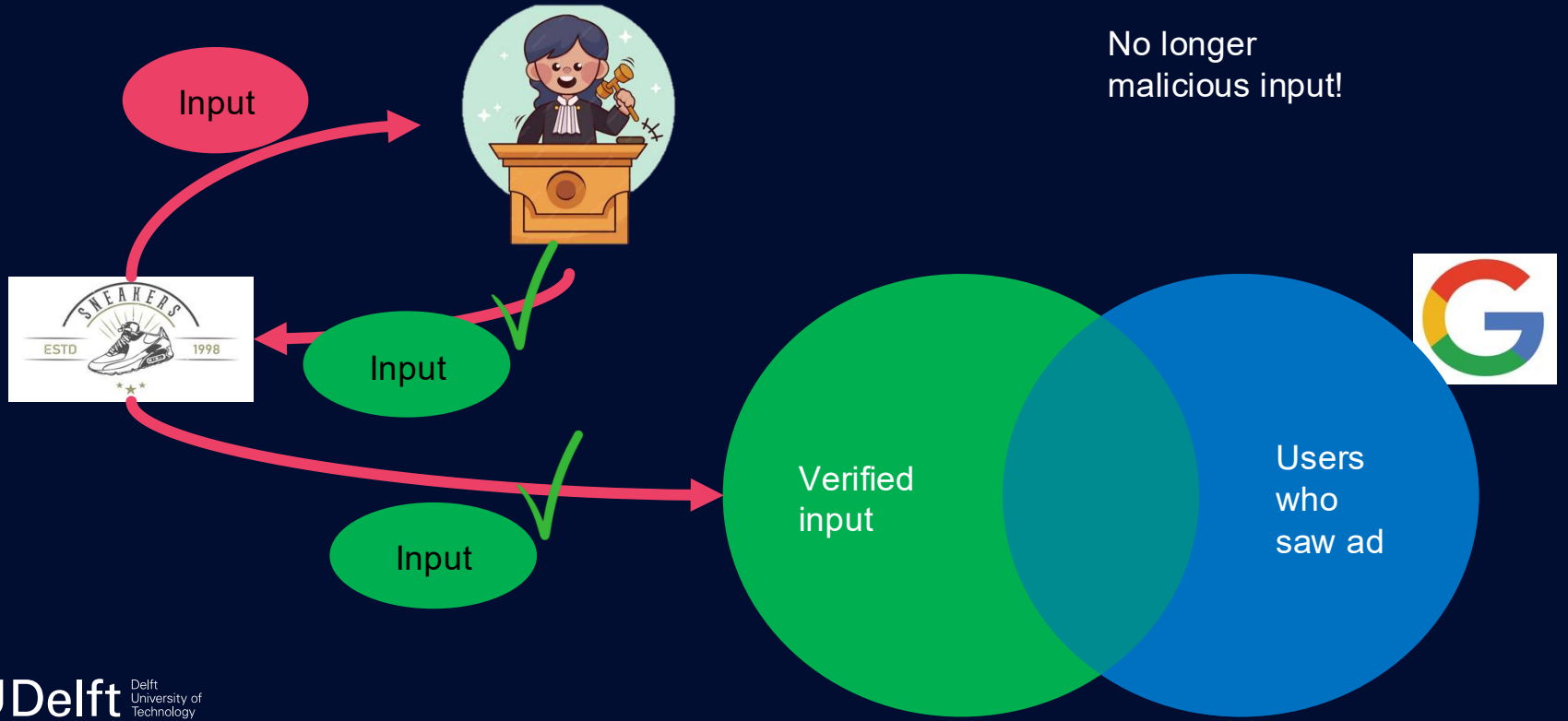


TU Delft Delft University of Technology

Today

- (Partial) Authorized PSI
- Faster Partial Authorized PSI
- How partial is partial? → Game theory

Authorized Private Set Intersection



But….

We have to fully trust the judge….

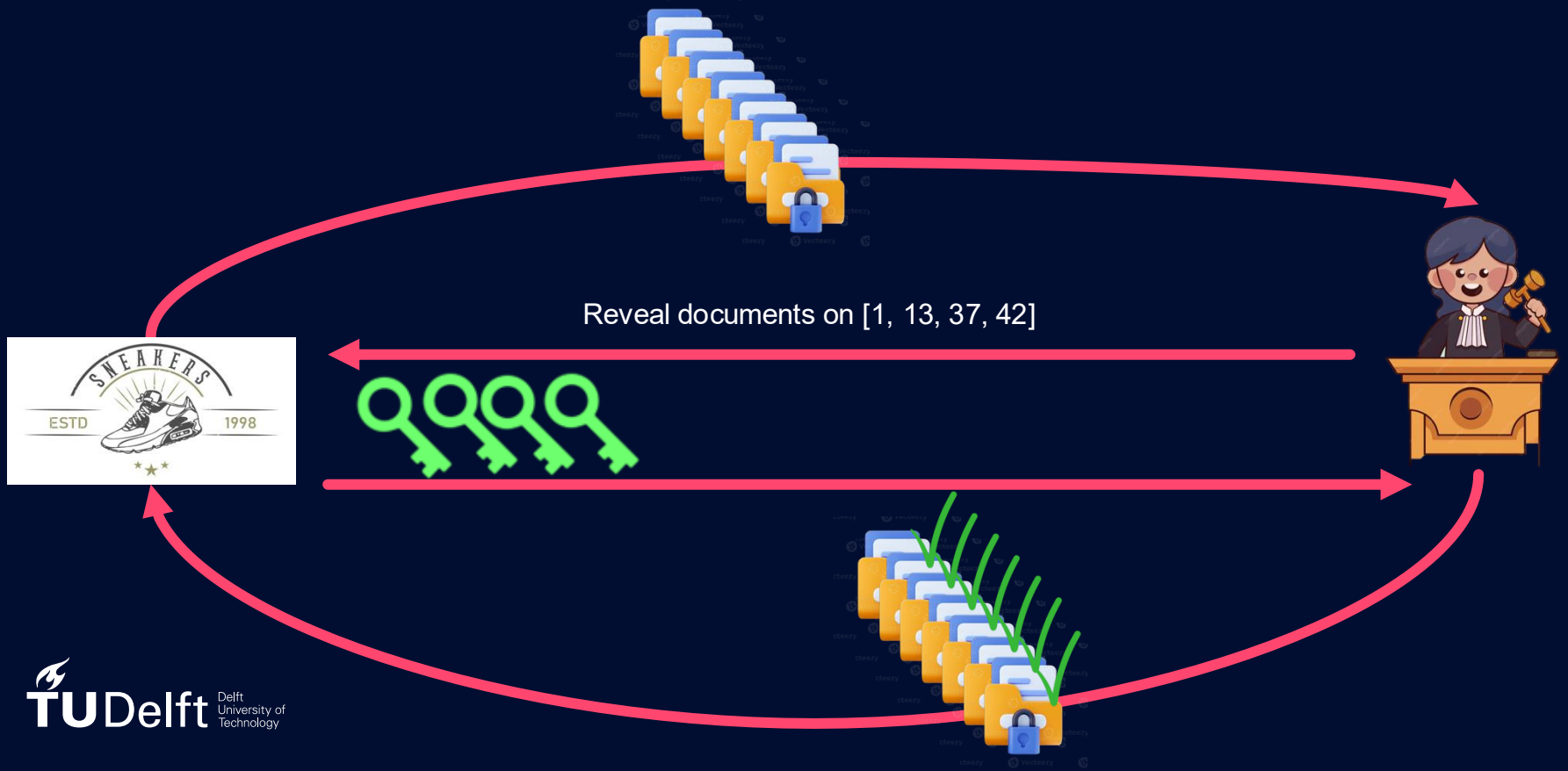
This problem was introduced and solved by:

Re-visiting Authorized Private Set Intersection: A New Privacy-Preserving Variant and Two Protocols

Francesca Falzon
ffalzon@ethz.ch
ETH Zürich
Zürich, Switzerland

Evangelia Anna Markatou
e.a.markatou@tudelft.nl
TU Delft
Delft, Netherlands

Partial Authorized Private Set Intersection



But….

That is a lot of bandwidth

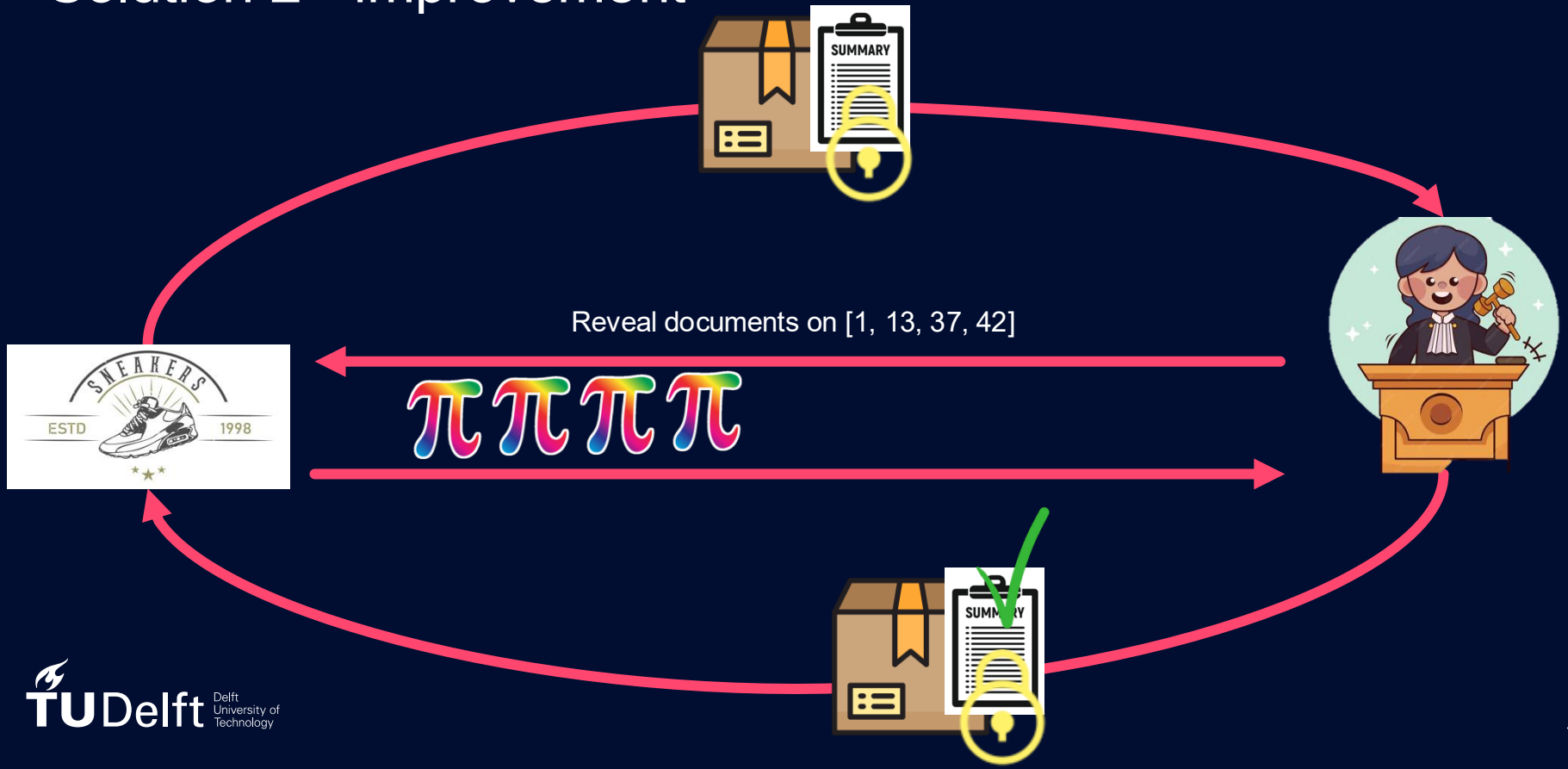
Solution 2

Paper 2025/2132

Bandwidth Efficient Partial Authorized PSI

Tjitske Ollie Koster, Delft University of Technology
Francesca Falzon, ETH Zurich
Evangelia Anna Markatou, Delft University of Technology

Solution 2 - improvement

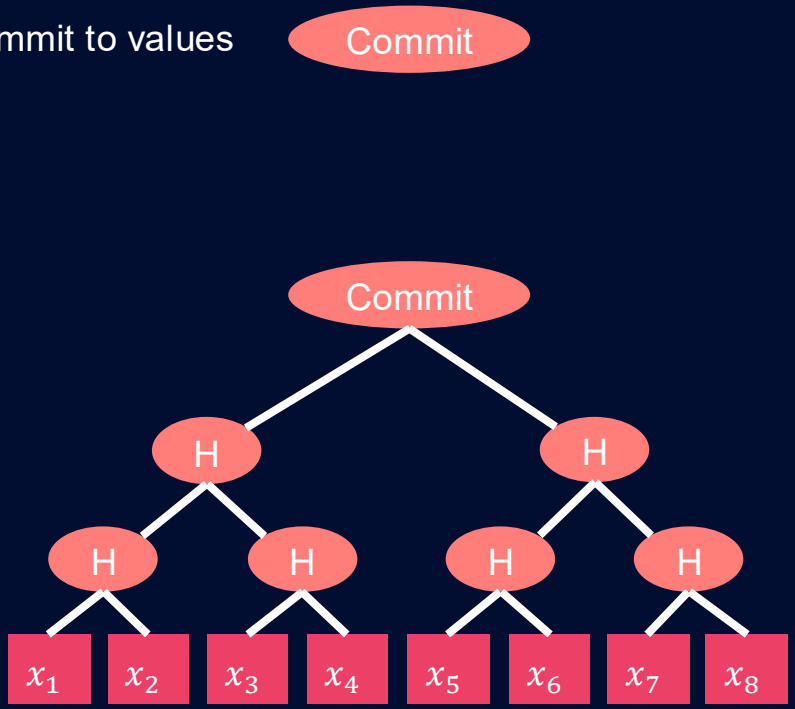


And then intersect a box?

What is a box?

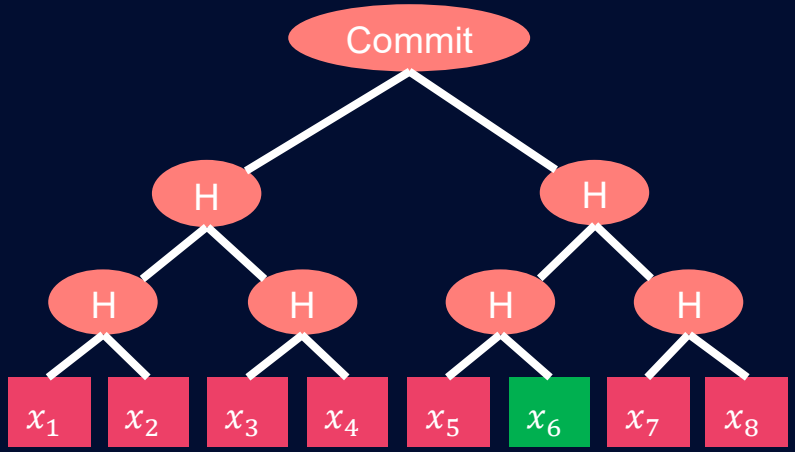
Vector commitment

- Commit to values



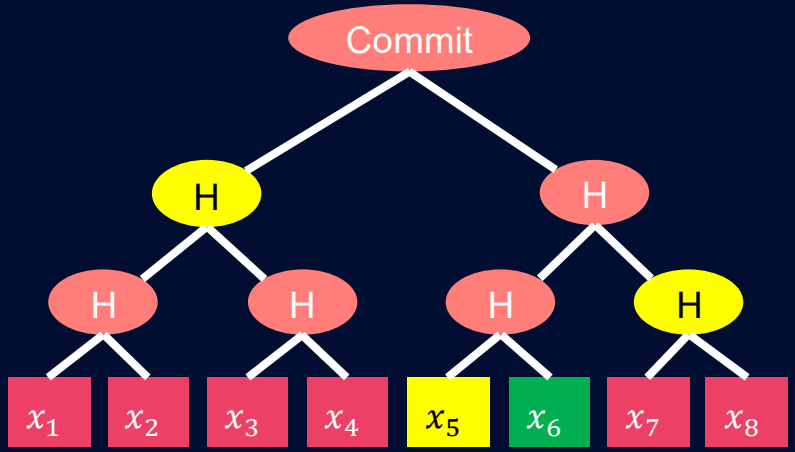
Vector commitment

- Commit to values
- Prove value x_i on position i of vector



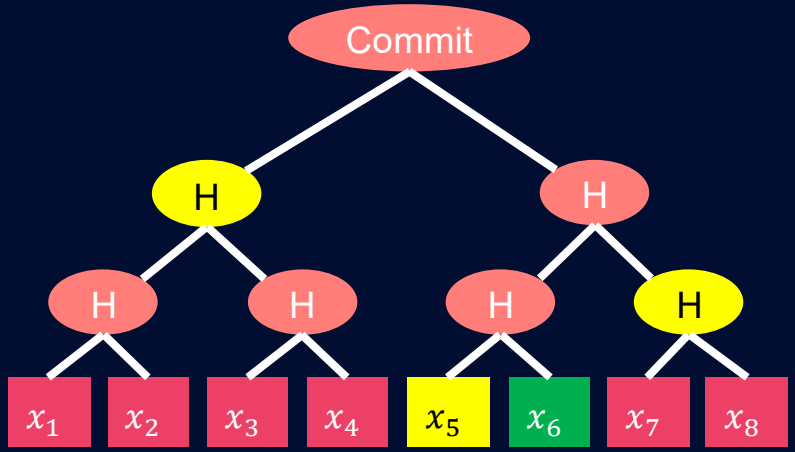
Vector commitment

- Commit to values
- Prove value x_i on position i of vector



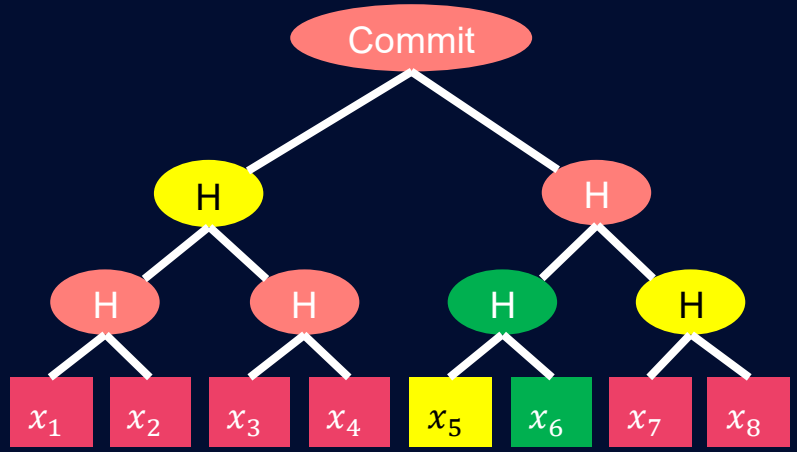
Vector commitment

- Commit to values
- Prove value x_i on position i of vector
- Verify, the proof π wrt



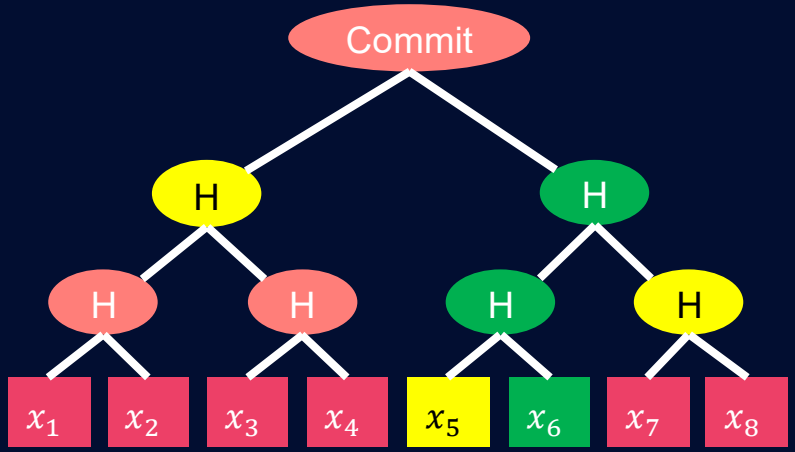
Vector commitment

- Commit to values
- Prove value x_i on position i of vector
- Verify, the proof π wrt



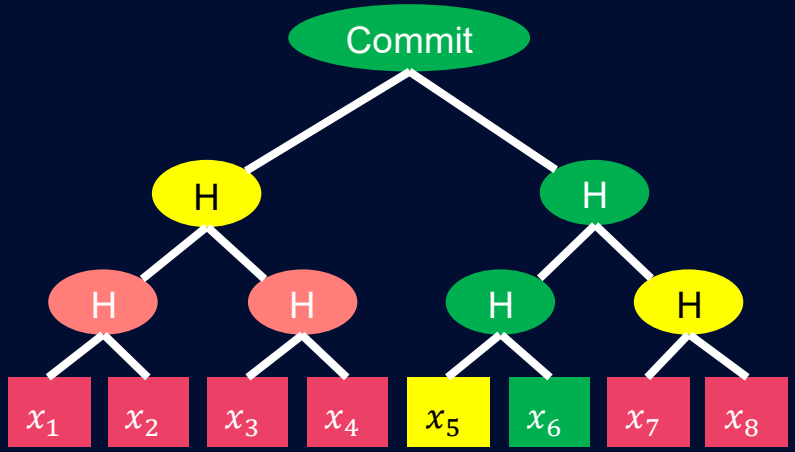
Vector commitment

- Commit to values
- Prove value x_i on position i of vector
- Verify, the proof π wrt



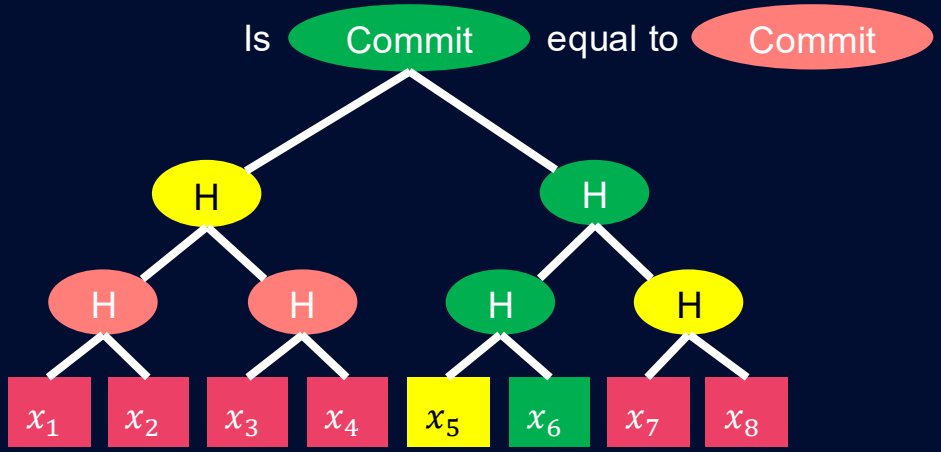
Vector commitment

- Commit to values Commit
- Prove value x_i on position i of vector 
- Verify, the proof  wrt Commit



Vector commitment

- Commit to values Commit
- Prove value x_i on position i of vector 
- Verify, the proof  wrt Commit



Vector commitment

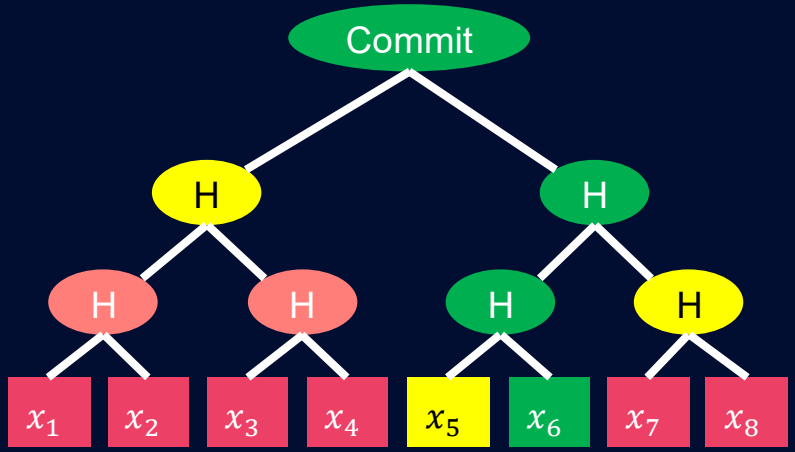
- Commit to values
- Prove value x_i on position i of vector
- Verify, the proof π wrt

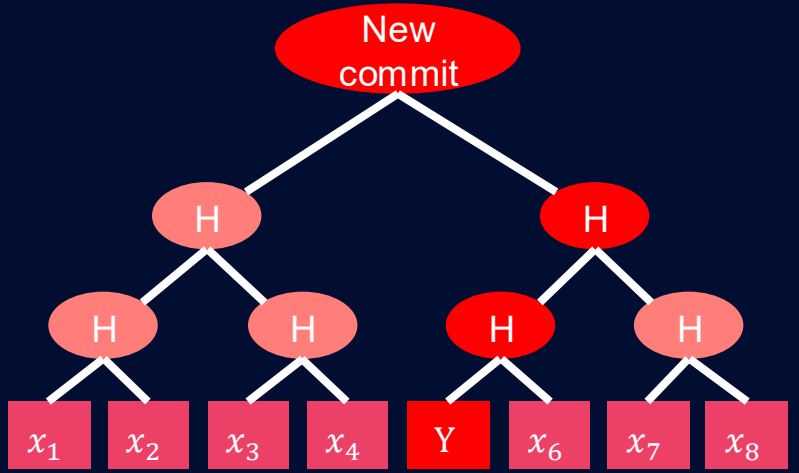
Commit

π

Commit

- Correctness
- No cheating





Vector commitment

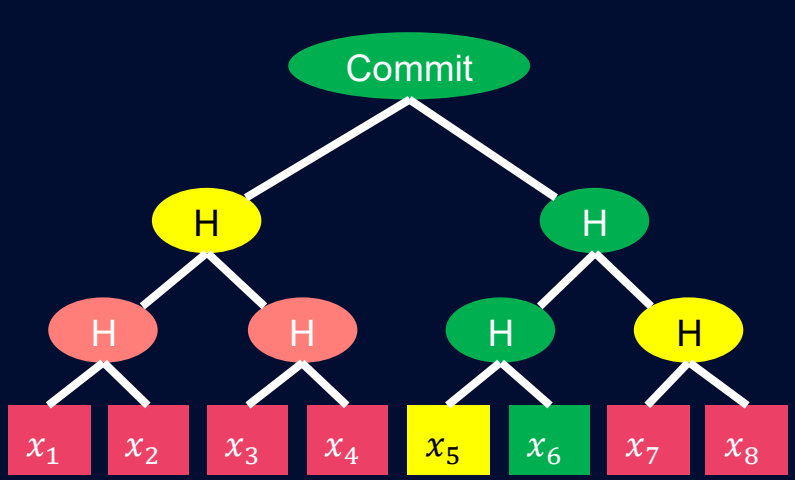
- Commit to values
- Prove value x_i on position i of vector
- Verify, the proof π wrt

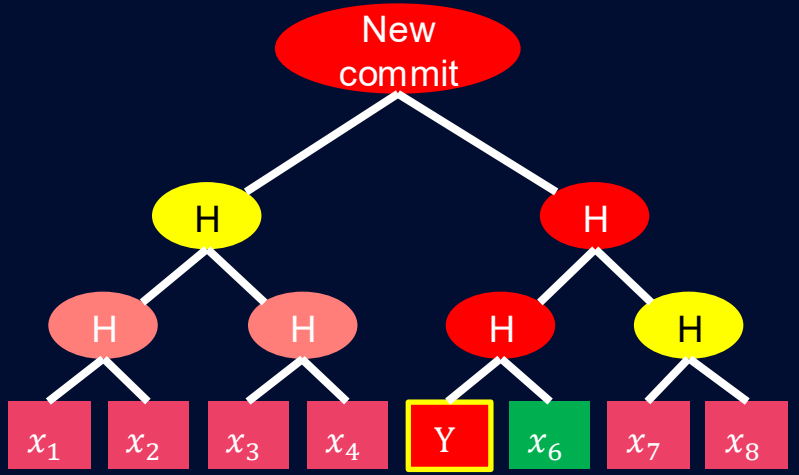
Commit

π

Commit

- Correctness
- No cheating





Vector commitment

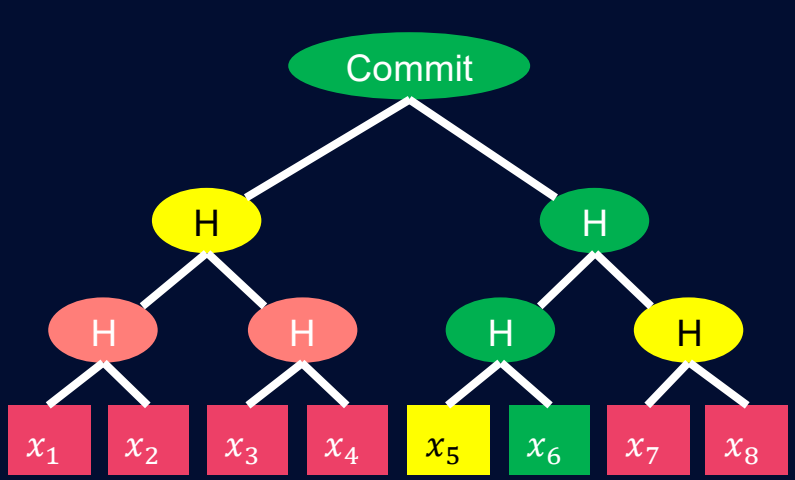
- Commit to values
- Prove value x_i on position i of vector
- Verify, the proof π wrt

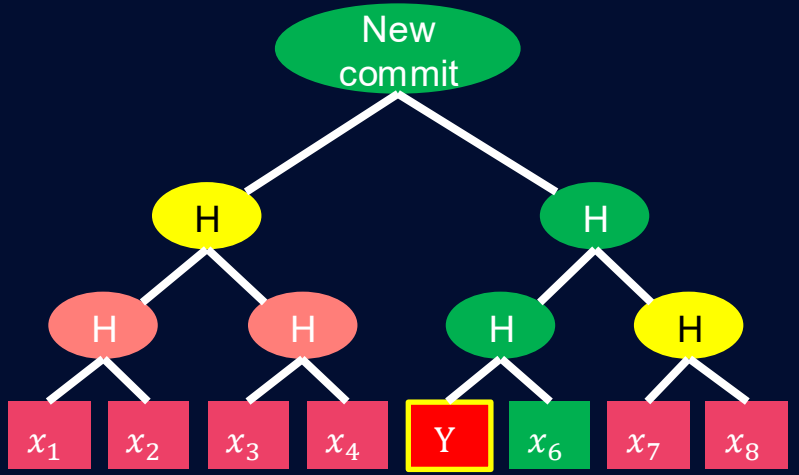
Commit

π

Commit

- Correctness
- No cheating





Vector commitment

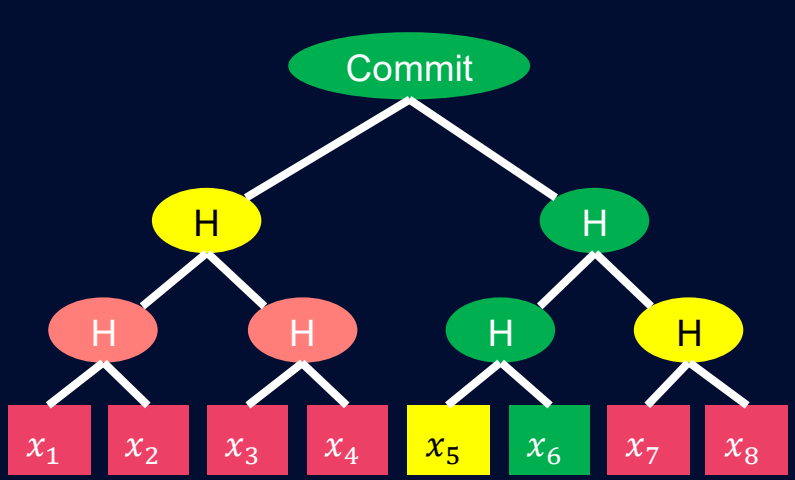
- Commit to values
- Prove value x_i on position i of vector
- Verify, the proof π wrt

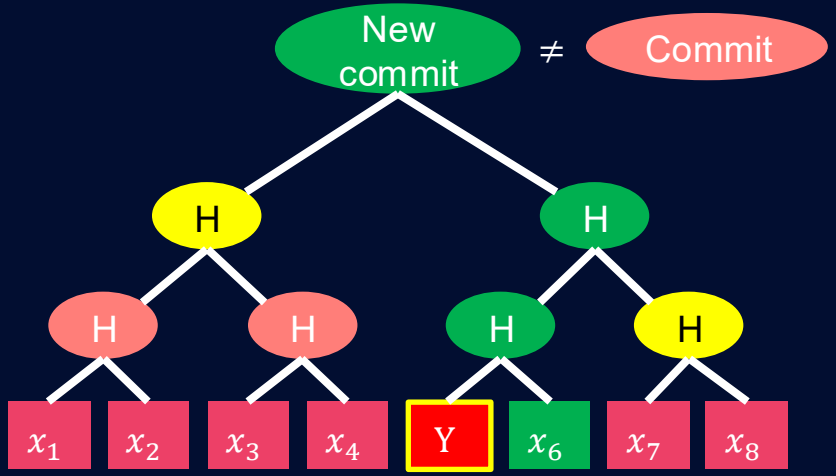
Commit

Commit

π

- Correctness
- No cheating



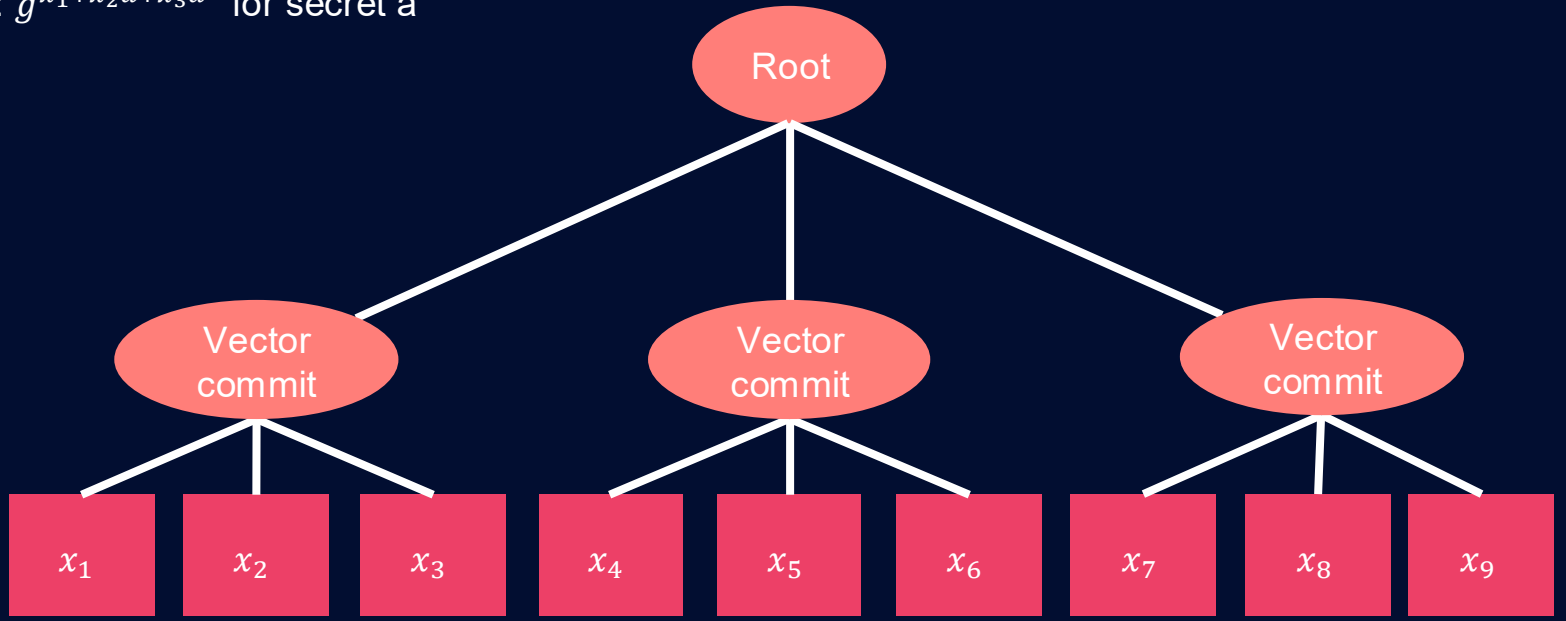


Still a lot of bandwidth

Proof is of length $\log_2(n)$

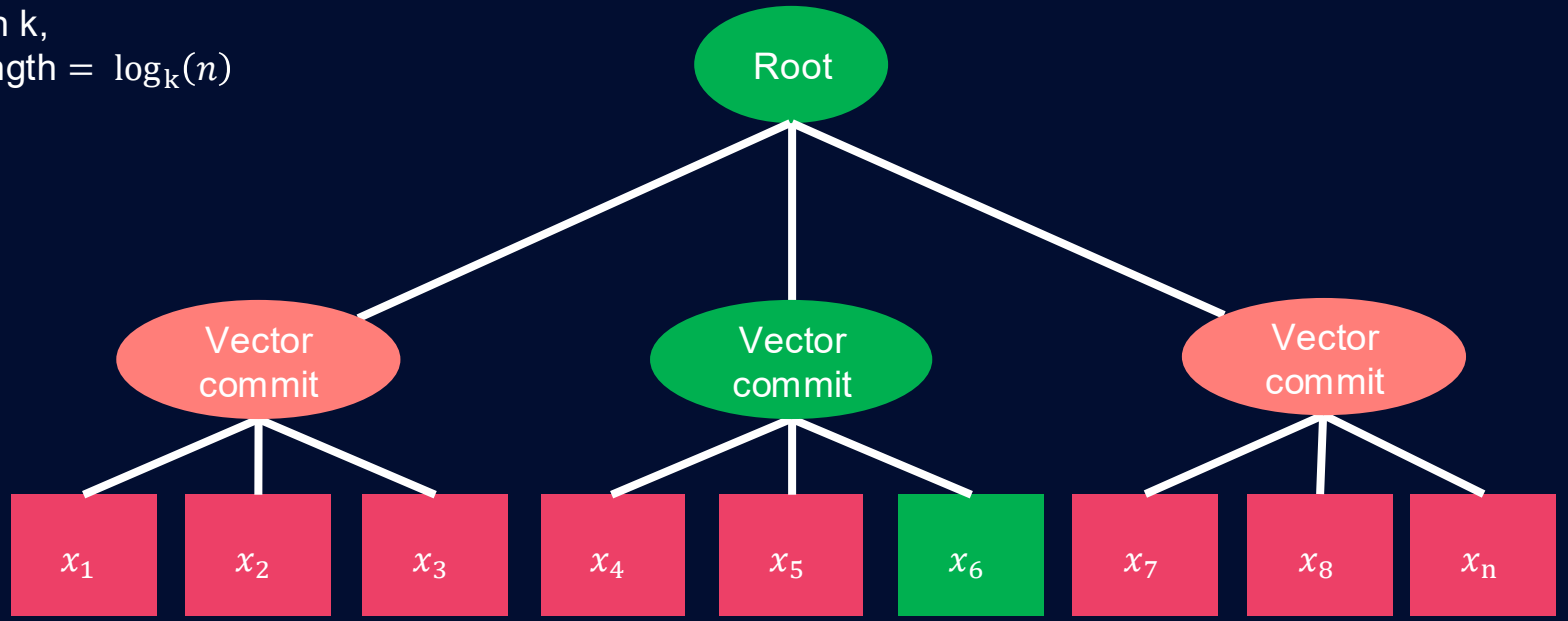
Verkle tree

Vector commitment → Pointproofs
Commitment: $g^{x_1+x_2a+x_3a^2}$ for secret a

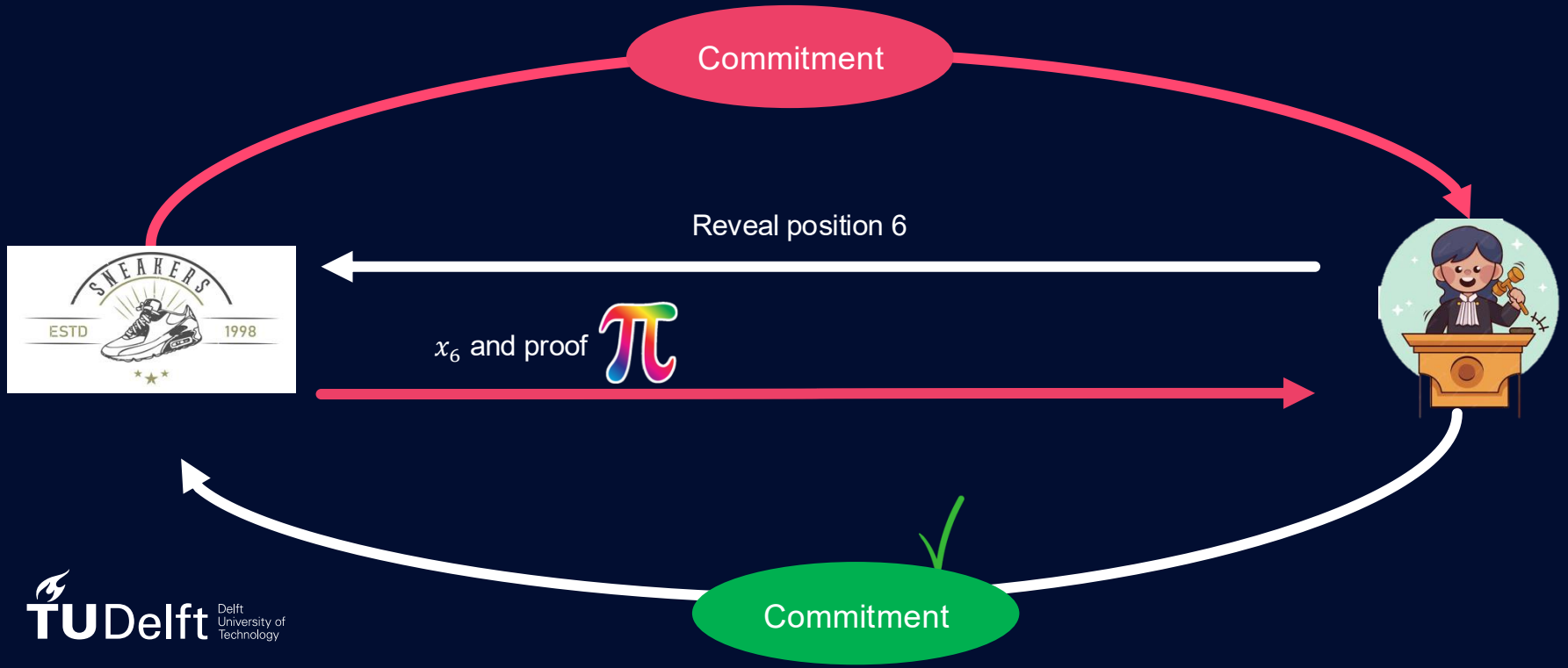


Verkle tree - proof

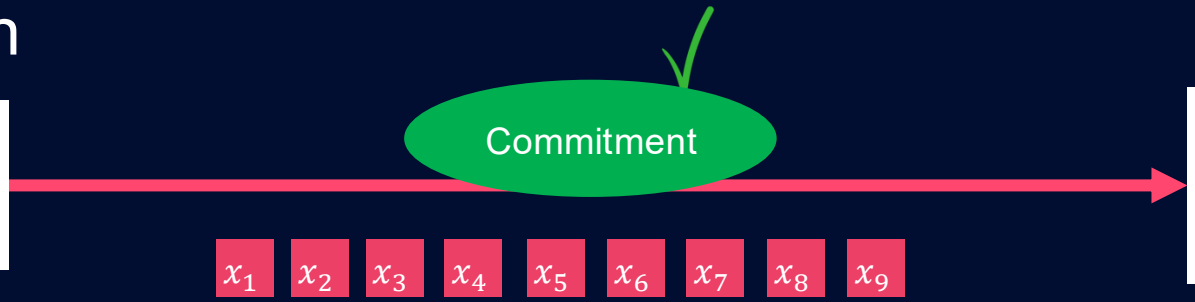
For width k,
Proof length $= \log_k(n)$



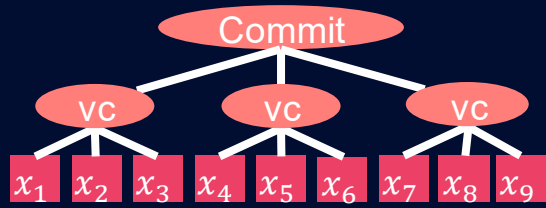
Authorization

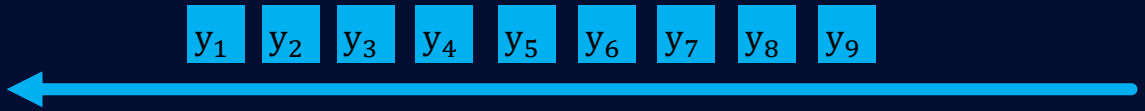


Intersection

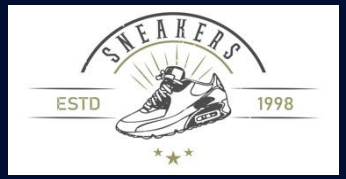


1. Build tree
2. Check commitment





Authorization – math version



$\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9\}$

$$x_i = H(m_i)^r$$

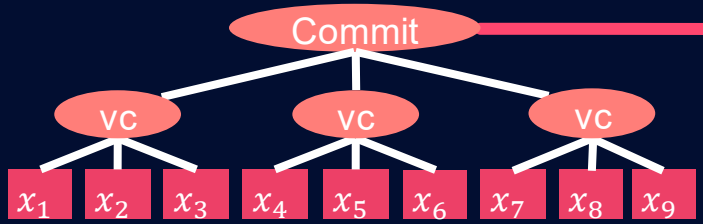
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

Make Verkle tree

Authorization – math version

$\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9\}$

$x_i = H(m_i)^r$



Commitment, r

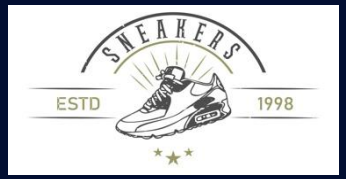
Send position 3,6,9

Proofs, π and $\{m_3, m_6, m_9\}$

1. Check elements
2. Check proofs
3. Sign

Commitment ✓

Intersection – math version



$\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9\}$

Commitment

x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

1. Build tree
2. Check commitment

$\{\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_8, \mu_7\}$

$x_i = H(m_i)^r$

$y_i^r = H(\mu_i)^{sr}$

Intersect

$\hat{x_i} = H(m_i)^{rs}$

$\{y_i\}$

$\{\hat{x_i}\}$

$y_i = H(\mu_i)^s$

$\hat{x_i} = x_i^s$

Secure

- Judge cannot learn more than the revealed elements
- Store cannot learn more than the intersection
- Google cannot learn any element

- Store: no universe attack
- Store: only intersection with authorized elements

Commitment

$y_i = H(m_i)^s$

$x_i = H(m_i)^r$

m_i for several i

$x_i = H(m_i)^{rs}$

r

Bandwidth efficient

Our protocol

Authorization

- Commitment
- Indices
- 
- Commitment

Intersection

- $x_i = H(m_i)^r$
- $y_i = H(m_i)^s$
- $x_i = H(m_i)^{rs}$

Falzon & Markatou

Authorization

- x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9
- Indices
- Proofs
- x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

Intersection

- y_i
- y_i
- y_i

It works!

- Correct
- Secure
- Bandwidth efficient

Last question

10%

How many elements should the judge check?

???

30%

Where can we cheat? And not get caught…

Commitment of Elements store

Request of some elements

Proof of commitment of those elements

Signature on commitment

Signature, Elements store

Elements Google + Elements store

Where can we cheat? And not get caught…

Commitment of Elements store

Request of some elements

Proof of commitment of those elements

Signature on commitment

Signature, Elements store

Elements Google + Elements store

Game theoretic analysis; where can we cheat?

- Input $\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_{666}, m_{\mathrm{n}}\}$
- Judge samples p percentage of elements
- Let μ be percentage malicious elements
- Probability Judge sees m_{666}?
- $A = 1 - \frac{\binom{\mathrm{n}-\mathrm{n}\mu}{\mathrm{n}p}}{\binom{\mathrm{n}}{\mathrm{n}p}}$
- Here, for $\mathrm{n}p = 1$, chance of seeing m_{666}=10%

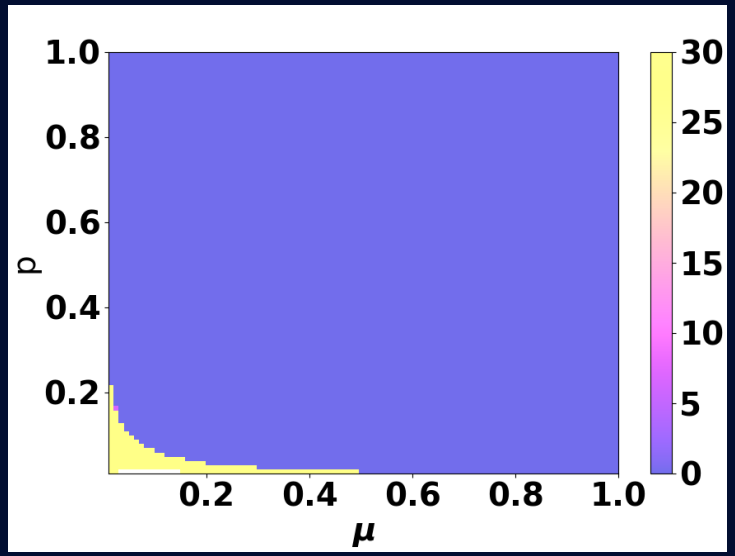
Is it worth cheating?

- Loss of revealing elements to judge
- Gain of computing intersection
- Gain of finding malicious elements
- Loss of getting detected

- \$20 $p \cdot n$
- \$353
- \$20 $\mu \cdot n$
- ????? fine

Cost of fine to ensure Store behaves well

Percentage the judge checks



Fine in dollars of detected cheating

Percentage malicious elements

Runtime protocol – green is where we do better

Set sizes		Scheme	Communcation (KB)		Runtime (ms)			Total Runtime			
Google	Store		Auth	Inter	Judge	Store	Google	LAN	1Gbps	200Mbps	50Mbps
2^{20}	2^{10}	FM	147	202473	1	750	8332	9083	9643	9643	9643
		Verkle	9	4203	21	58	124	202	203	203	203
	2^{16}	FM	9520	208666	78	1301	8361	9741	10301	10301	10301
		Verkle	557	4719	37	239	162	438	438	438	439
	2^{20}	FM	152256	303038	1196	10558	8390	20145	20705	20705	20705
		Verkle	8907	12583	168	2263	979	3410	3411	3412	3414
	2^{24}	FM	2436770	1812987	19454	171482	8357	199295	199855	199855	199855
		Verkle	142499	138412	1382	35356	11076	47814	47817	47826	47859
	2^{26}	FM	9738788	6644826	75754	727536	8399	811690	812250	812250	812252
		Verkle	569989	541065	4832	166082	40176	211090	211099	211135	211268

What did we do today?

- How to prevent malicious input in PSI?
  - Solution Falzon Markatou – Partial Authorized PSI
  - Our solution – faster
  - Determine percentage p – game theory
- Future research
  - Post-Quantum secure
  - Eliminate the judge

Contact me

Check out our eprint