SECRET INTERSECTION COUNT FOR NETWORK ALGORITHMICS

ALEXANDER BOYD

COSENERS 2025

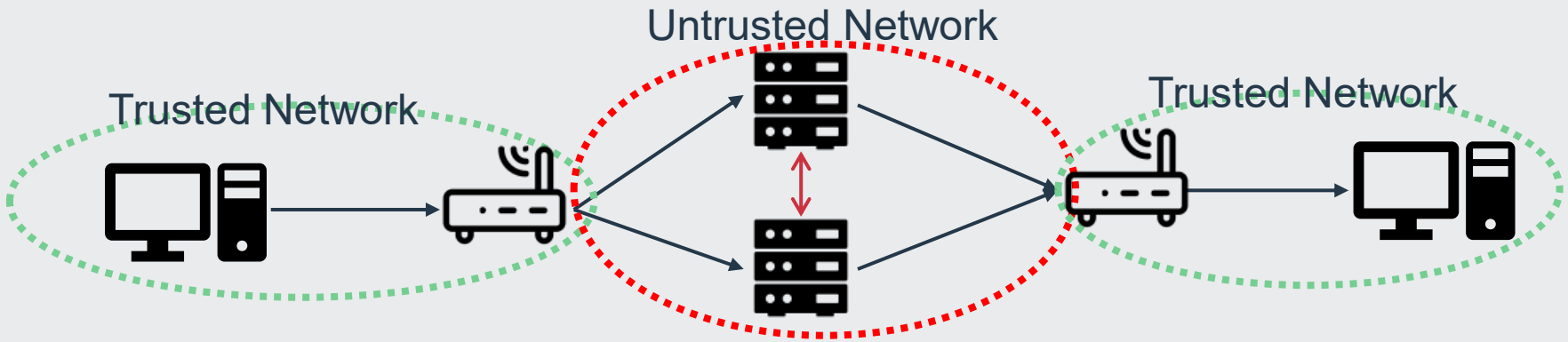
UNIVERSITY OF SURREY

SECRET NETWORK ALGORITHMS

- Real pressure for data-private algorithms
- Already much research in data-private algorithms, but very little targeted at networks
- Realistic expectation that network algorithms will become data-private
- PPM problems in networks are of primary interest

WORKED EXAMPLE

- A client wishes to send from a trusted network to another trusted network, separated by an untrusted network
- The untrusted network contains (in the simple case) one pair of potentially malicious but non-colluding servers
- The untrusted servers wish to calculate aggregate statistics over many packets
- The sender wishes for their IP to be hidden



Router and server icons by Angelo Troiano and Chunk Icons from Noun Project

WORKED EXAMPLE (CONT.)

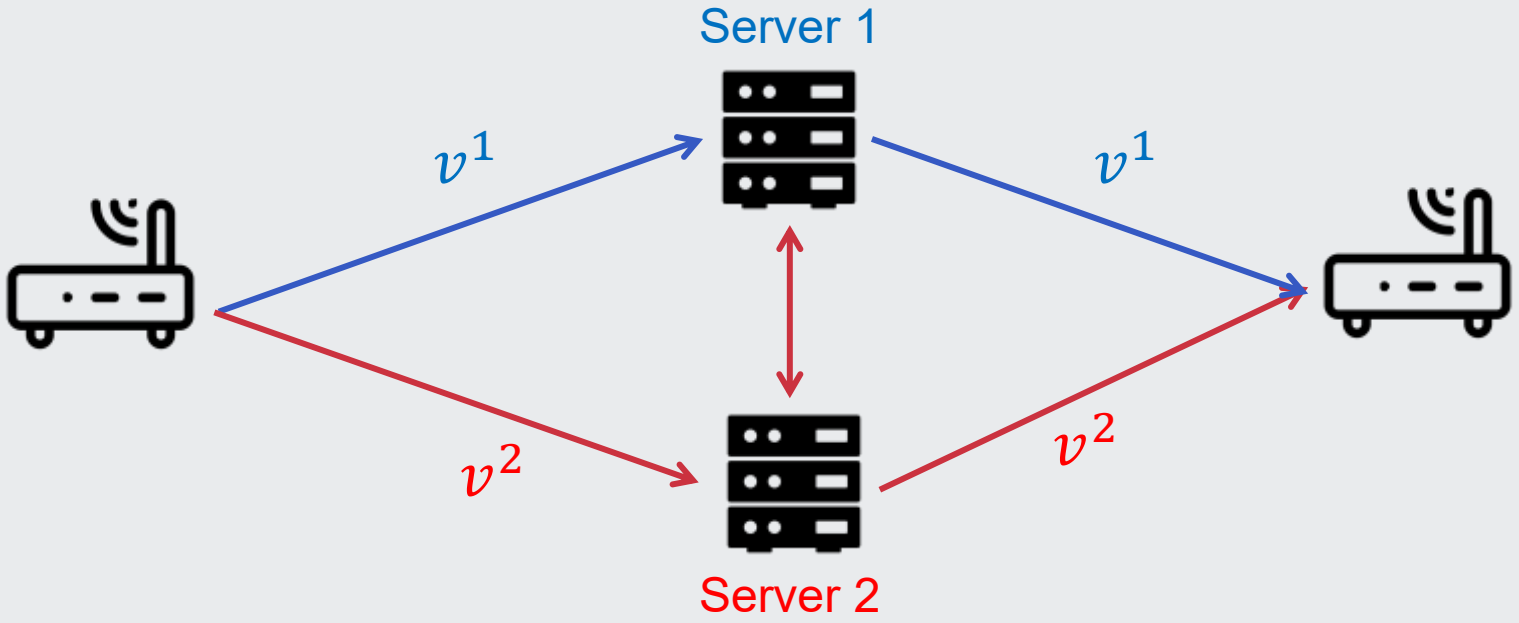
- Local gateway splits packets
  - Payload is split in 2, evenly
  - Header is duplicated except source field
  - Source field becomes a random Boolean secret share

- e.g. $\underbrace{10010101}_{v} \rightarrow \underbrace{01001010}_{v^1} \oplus \underbrace{11011111}_{v^2}$

Real data
Server 1 share
Server 2 share

- For our purposes, the servers wish to calculate how many packets have a source address equal to t

WORKED EXAMPLE DIAGRAM



- Server 1 gets v^1 in its header and server 2 gets v^2
  - No single server learns anything about the source
  - We can still calculate aggregates, as we will now see…

FREQUENCY COUNT USING PRIVATE EQUALITY TESTING

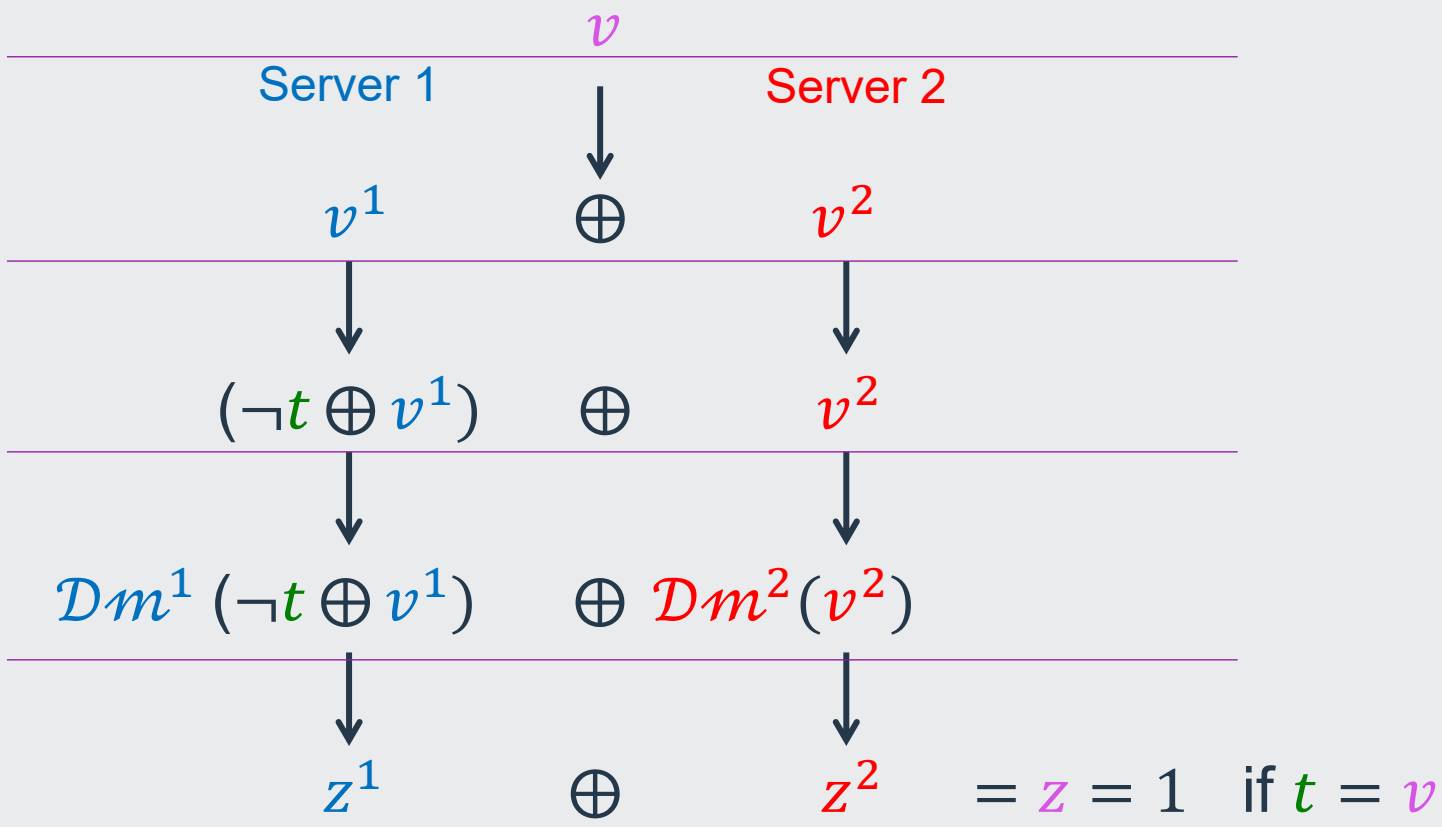
$$v = v^1 \oplus v^2 = t?$$

$$t \oplus v^1 \oplus v^2 = 0, \quad \text{if } t = v$$

$$\therefore \neg t \oplus v^1 \oplus v^2 = 1, \quad \text{if } t = v$$

- Multiply all bits together and test if it equals 1!
- Boolean multiplication is appealing because the efficiency of secret multiplication scales with the size of finite field
- Ideally the result will be shared between both servers, e.g. $0 \oplus 1$ instead of 1
  - This allows us to aggregate the results without revealing an individual's equality test result

FREQUENCY COUNT PROTOCOL DIAGRAM



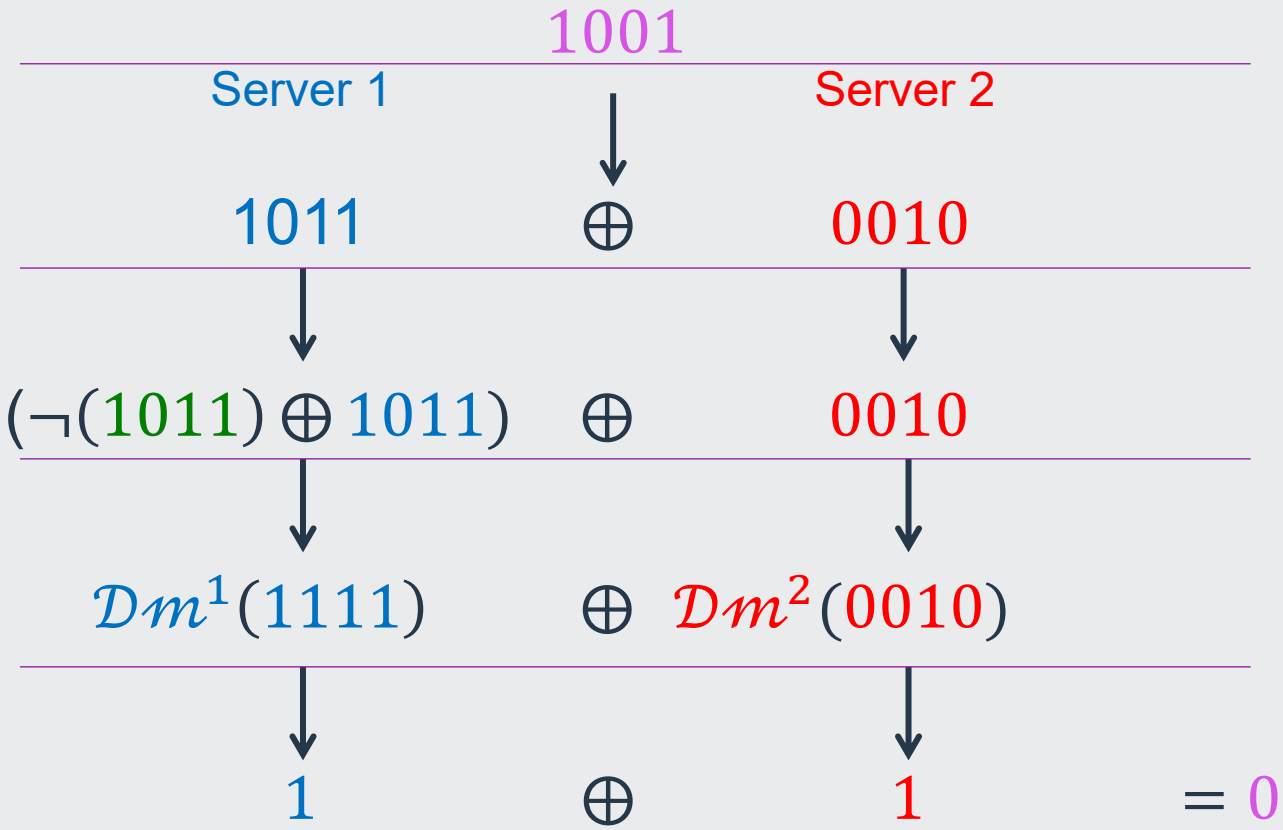
DISTRIBUTED MULTIPLICATION

- $m(v) = z$
- $m(1010) = 0$
- $m(1111) = 1$

- $\mathcal{D}m(v) = \mathcal{D}m^1(v^1) \oplus \mathcal{D}m^2(v^2) = z^1 \oplus z^2 = z$
- $\mathcal{D}m(1010) = \mathcal{D}m^1(0111) \oplus \mathcal{D}m^2(1001) = 1 \oplus 1 = 0$
- $\mathcal{D}m(1111) = \mathcal{D}m^1(0101) \oplus \mathcal{D}m^2(1010) = 0 \oplus 1 = 1$

FREQUENCY COUNT EXAMPLE

$t = 1011$



$= 0$

because $t \neq v$

SECRET MULTIPLICATION

- $x = x^1 + x^2, \quad y = y^1 + y^2$; everything in $\mathbb{F}$
- Server 1 has (x^1, y^1), server 2 has (x^2, y^2)
- Want to calculate $x \cdot y = z = z^1 + z^2$
- $x \cdot y = (x^1 + x^2) \cdot (y^1 + y^2) = x^1y^1 + x^1y^2 + x^2y^1 + x^2y^2$
- Need suitable values for z^1 and z^2
- Server 1 can provide the x^1y^1 part, server 2 the x^2y^2 part
- The hard bit is the cross-products

BEAVER TRIPLES

- Randomly generated triple of multiplication (a, b, c), where $c = ab$
- $(a, b, c) = (a^1, b^1, c^1) + (a^2, b^2, c^2)$
- Calculated collaboratively, no single server knows real values
  - (details of triple generation omitted)
- Allows the servers to combine their secrets and safely publish
- The servers can then multiply the published values and ‘fix’ according to their secrets
- $d^i = x^i - a^i,\ e^i = y^i - b^i$
- $d = d^1 + d^2,\ e = e^1 + e^2$
- z^i is calculated as: $\frac{1}{2}de + db^i + ea^i + c^i$

BEAVER TRIPLES (CONT.)

$$d = x - a, \qquad e = y - b$$

- $\sum_i \frac{1}{2} de + db^i + ea^i + c^i$

  $= de + db + ea + c$

  $= (x - a) \cdot (y - b) + (x - a) \cdot b + (y - b) \cdot a + c$

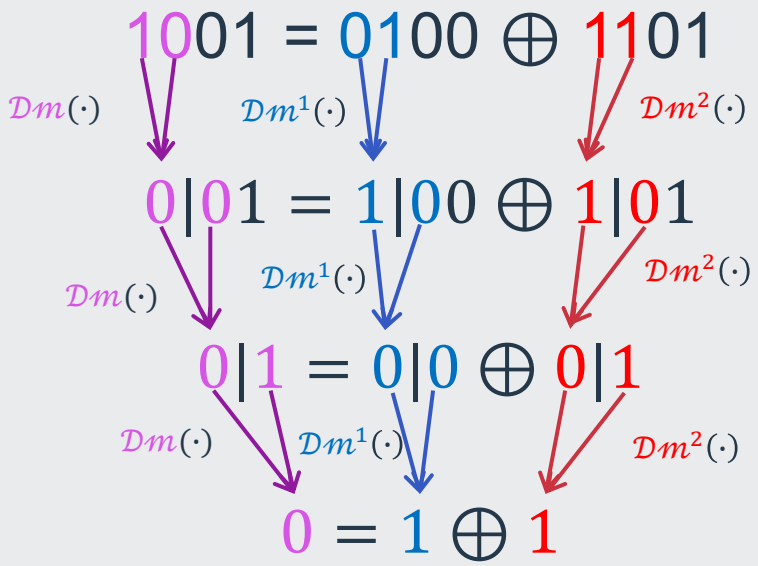
  $\vdots$

  $= xy$

- Beaver triples can be precomputed
  - ‘Live’ communication of 2 bits both ways

GOING BACK…

- Recall: $\mathcal{D}m(v) = \mathcal{D}m^1(v^1) \oplus \mathcal{D}m^2(v^2) = z^1 \oplus z^2 = z$
- Since the output is secret shared, we can repeat protocol until all bits have been multiplied



PRIO+ PROTOCOL

- Cannot aggregate in $\mathbb{F}_2$
- $1 = 0 \oplus 1 \rightarrow 5 + 6$ in $\mathbb{F}_{10}$
- $0 = 1 \oplus 1 \rightarrow 3 + 7$
- $1 = 0 \oplus 1 \rightarrow 1 + 0$
- $0 = 0 \oplus 0 \rightarrow 9 + 1$

$$8 + 4 = 2$$

- We can do this conversion in a similar way to Beaver triples

ALL IN ALL…

- We now have a complete system for privately querying frequency count of a specific source
  - One could query prefixes too
- Efficiency: highly parallelisable and requires only a small amount of communication per packet
  - e.g. $((160 - 1) \cdot 4 + 2 = 638)$ / 8 = ~80 bytes per packet for 20-byte source field
  - ~80kB for 1000 packets
- Future work: there are surely many other network algorithms that can be privatised


UNIVERSITY OF
SURREY