Privacy-Preserving AIS for Network Security

Karthik S
Chandan Yeshwanth

March 8, 2016

Recap

- 3 step protocol:
  - Generation of random secret-shared detector set
  - Tolerization of generated detector set on safe connections
  - Usage of trained detector set on incoming connections

Bitwise additive secret sharing

- Assume an n party fully connected system
- Assume each bitstring x is an xor of n random shares i.e, $x = x_1 \oplus x_2 \oplus ... \oplus x_n$
  - Each x_i is an l bit string
- Generalized bitwise additive share generation[1]:
  - Initially generate $n-1$ random bit strings $x_1, x_2, \ldots, x_{n-1}$
  - The last share is determined by : $x_n = x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1} \oplus x$
  - The shares $x_1, x_2, \ldots, x_n$ are sent to the n parties

Detector construction

- Let q, l, n be the initial number of detectors, encoding bit length and number of parties respectively
- P_i generates a random set of detector shares (indexed l bit random strings)
  $D_i = \{d_{ab}, x \in \{0,1\}^l \wedge (1 \leq a \leq q) \wedge (1 \leq b \leq n)\}$
- The actual detector is given by $d_a = \oplus d_{ab}, 1 \leq b \leq n$
- At the end of the random detector construction phase each party has shares of q bit-wise additive secret shared detectors, i.e., $\{[\![d_1]\!], [\![d_2]\!], \ldots, [\![d_q]\!]\}$
- Note: No communication among parties during construction

Proof of randomness

- The detector set generated above is random assuming each party generates a random share
- Proof
  - Assume a trusted third party T that generates a random detector d and generates n shares
  - From the generation protocol, T generates random $d_1, d_2, \ldots, d_{n-1}$
  - Then n^{th}share, $d_n = d_1 \oplus d_2 \oplus \ldots \oplus d_{n-1} \oplus d$
  - This implies d_n is also random (since it is an xor of random bit strings)
  - This is equivalent to generating a random d_n

Affinity function

- Affinity function modelled as a black box
- Given a secret shared detector $[\![d]\!]$ and a secret shared connection $[\![c]\!]$, the affinity function black box works as follows:

$$A_f([\![d]\!], [\![c]\!]) = [\![z]\!]$$

where z is the affinity metric between the detector and the connection.

Tolerization

- The tolerization protocol is as follows.
- For a given party P_i and threshold affinity th:
  - On receiving a connection c, generate shares $c_1, c_2, \ldots, c_n$ using BASS
  - For each detector d_i
    - Calculate $[\![z]\!] = A_f([\![d_i]\!], [\![c]\!])$
    - If $[\![z]\!] \stackrel{?}{>} th$, discard $[\![d_i]\!]$

Michael Mortensen.
Secret sharing and secure multi-party computation.
University of Bergen, 2007.