Privacy Preserving Data Mining

Yehuda Lindell, Benny Pinkas

Presented by: Wenxin Li

About this paper

- *Journal of Cryptology*, 2002
- Y. Lindell
  - Cryptographic research group, IBM
- B. Pinkas
  - Trusted Systems Lab, HP
- More Cryptography than Data Mining
- An efficient Cryptographic tool for private preserving data mining

Comparison with last paper

- Last paper
  - Randomization Approach
  - One data owner, one data miner
  - Numerical attributes
  - Close to the non-privacy-preserving computation solution
- This paper
  - Crytographic Approach
  - Two data (owner+miner)
  - Categorical Attributes
  - Same solution as the non-privacy-preserving computation

Problem

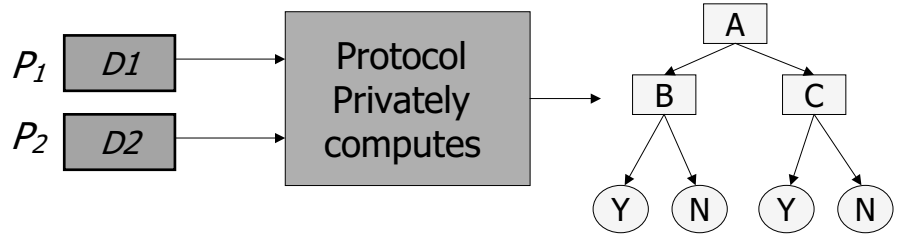
- Two parties P_1 and P_2
- P_1 own database D_1, P_2 own D_2
- Wish to build a decision-tree classifier on joint databases ($D_1 \cup D_2$)
- Without revealing any unnecessary information about their individual databases

Yao's two-party protocol

- Party 1 with input *x*
- Party 2 with input *y*
- Wish to compute *f(x,y)* without revealing *x,y*.
- Most related works use similar methodology as Yao's

Related works of secure two party computation

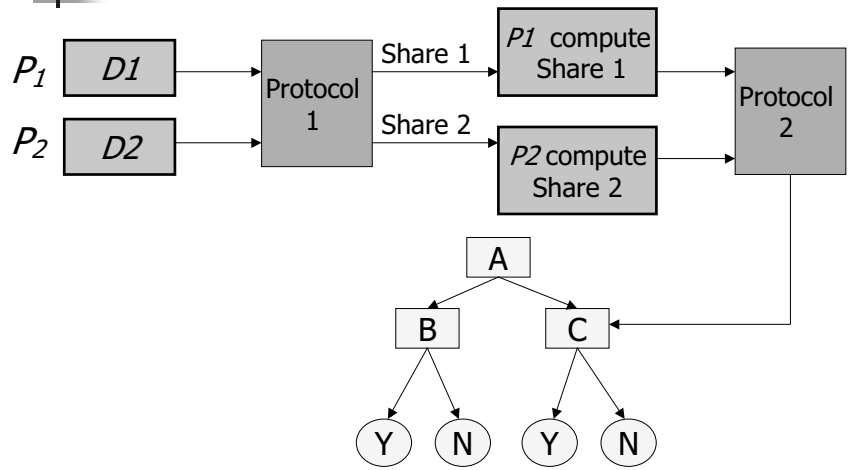
- Protocol depend on size(input)
- Dataset for DM is too large for it
- Need protocol handle large DB & compute efficiently



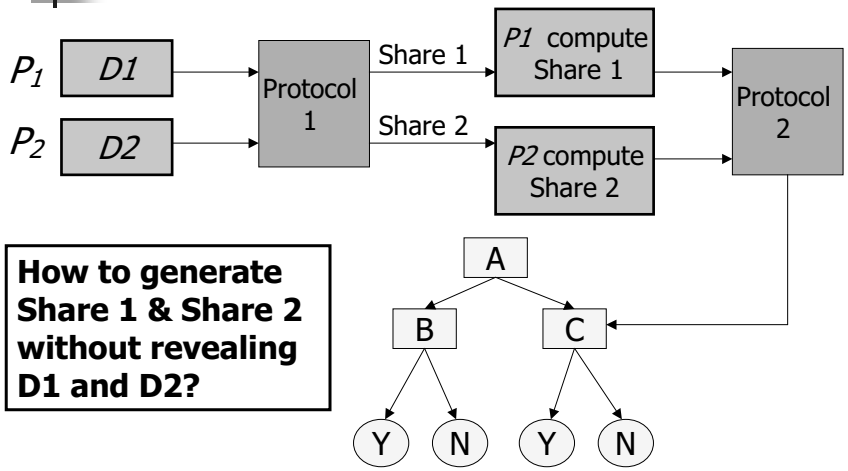
Idea

- Have P_1 and P_2 do the majority of the computation independently
- Without leaking any information
- Contribution of this paper

Protocols of this paper



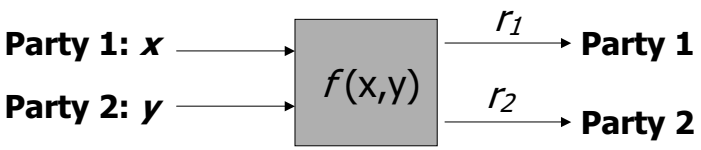
Protocols of this paper



How to generate Share 1 & Share 2 without revealing D1 and D2?

Yao's circuit evaluation computing random shares.

- Party 1 with input *x*
- Party 2 with input *y*
- Compute f(x,y)
  - *Party 1* gets random share r_1,
  - *Party 2* gets random share r_2
  - $\boldsymbol{r_1 + r_2 = f(x,y)}$



Assumption

- Semi-honest parties: Correctly follows the protocol specification, yet attempts to learn additional information by analyzing the messages
- D_1 and D_2 Have same structure and attributes, names are public
- Each attribute is categorical(ID3)

Notations

- R : set of attributes
- C : class attribute
- T : set of transactions
- $T(c_i)$:set of transactions in class i
- $T(c_i, a_j)$: set of transactions in class i with attribute value(A) = a_j

ID3(*R,C,T*) Recursive process:

- If *R* is empty, return a leaf-node with class value of the majority of the transactions in *T*
- If transactions in T with same class *c*, return a leaf-node labled with *c*
- Otherwise, *select attribute* that best classifies the transcitions in *T* (highest information gain)

Information Gain

- $Gain(A) = Entropy(T) - Entropy(T|A)$
- $Entropy(T) = \sum_{i=1}^{l} -\frac{T(ci)}{|T|}\log\frac{|T(c_i)|}{|T|}$
- $Entropy(T|A) = \sum_{j=1}^{m}\frac{|T(aj)|}{|T|}Entropy(T(a_j)) =$

$$\frac{1}{|T|}\left(-\sum_{j=1}^{m}\sum_{i=1}^{l}|T(a_j,c_i)|\log(|T(a_j,c_j)|+\sum_{j=1}^{m}|T(a_j)|\log(|T(a_j)|)\right)$$

Select split attribute(1)

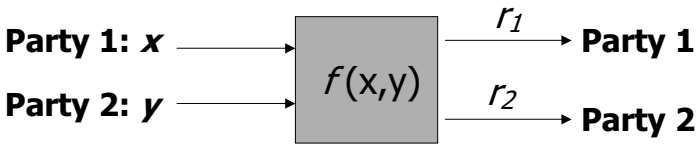
- Need to compute
  - $\Sigma_j \Sigma_i |T(a_j, c_i)|\ \log |T(a_j, c_i)|$
  - $\Sigma_j\ |T(a_j)|\ \log |T(a_j)|$
- $|T(a_j)| = |T_1(a_j)| + |T_2(a_j)|$
- $|T(a_j, c_i)| = |T_1(a_j, c_i)| + |T_2(a_j, c_i)|$
- Expression to be compared written as

$$(v_1 + v_2)\ln(v_1 + v_2)$$

  - v_1 is known to P_1
  - v_2 is known to P_2

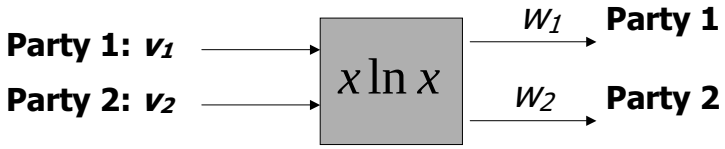
Select split attribute(2)

- problem is reduced to privately computing function $x \ln x$
- Use Yao's circuit evaluation
- Function *f* is $x \ln x$



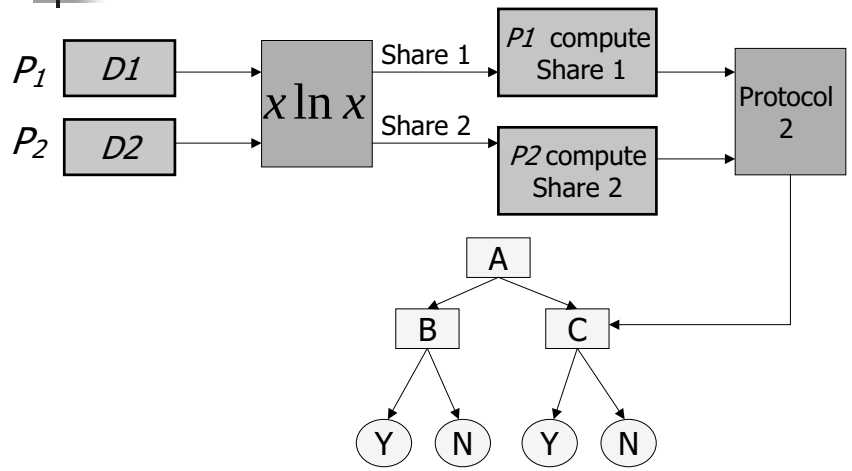
Simplified Protocol $x \ln x$

- Input:
  - P_1 input a value v_1, P_2 input a value v_2
- Output:
  - P_1 obtains share w_1, P_2 obtains share w_2
  - $w_1 + w_2 = (w_1 + w_2) \ln (w_1 + w_2)$



$$w_1 + w_2 = (v_1 + v_2) \ln ((v_1 + v_2)$$

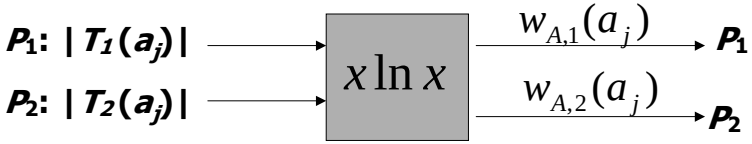
Protocols of this paper



Select split attribute Stage 1: computing shares

- $\forall A, \forall a_j \in A, \forall c_i \in C,$ and use the xlnx protocol to obtain random shares :$w_{A,1}(a_j), w_{A,2}(a_j), w_{A,1}(a_j, c_i), w_{A,2}(a_j, c_i)$

$$w_{A,1}(a_j) + w_{A,2}(a_j) = |T(a_j)| \ln(|T(a_j)|)$$
$$w_{A,1}(a_j, c_i) + w_{A,2}(a_j, c_i) = |T(a_j, c_i)| \ln(|T(a_j, c_i)|)$$

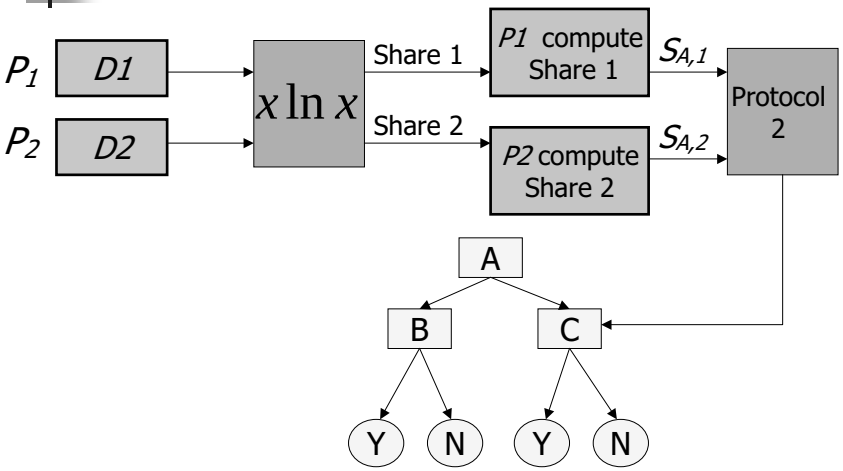


Select split attribute Stage 1: computing shares

- $\forall A, \forall a_j \in A, \forall c_i \in C,$ *given shares* :$w_{A,1}(a_j), w_{A,2}(a_j), w_{A,1}(a_j, c_i), w_{A,2}(a_j, c_i)$
- P_1 and P_2 and compute their shares in Entropy($T|A$)

$$S_{A,1} = -\sum_{j=1}^{m} \sum_{i=1}^{l} w_{A,1}(a_j, c_i) + \sum_{j=1}^{m} w_{A,1}(a_j)$$
$$S_{A,2} = -\sum_{j=1}^{m} \sum_{i=1}^{l} w_{A,2}(a_j, c_i) + \sum_{j=1}^{m} w_{A,2}(a_j)$$
$$S_{A,1} + S_{A,2} = Entropy(T \mid A)$$

Protocols of this paper



Select split attribute Stage 2: finding the attribute

- Given
  - entropy share $S_{A,1}$ calculated by P_1
  - entropy share $S_{A,1}$ calculated by P_2
- Use Yao circuit evaluation
- Get Entropy($T|A$)
- Find the attribute with Maximum Gain

Privacy

- Stage 1
  - Involves many invocations of private protocol *xlnx* that outputs random shares
  - Local computation
- Stage 2
  - Involves only a single invocation of Yao's circuit evaluation

Protocol Efficiency

- A concrete example
  - $|T| = 2^{20}$, $|R| = 15$, $m=10$
  - Generic solution requires 60 million exponentiations
  - This solution only requires 12000

Conclusion

- Preserve Privacy in Data Mining
- Improve Efficiency