Optimizing Secure Decision Tree Inference Outsourcing

汇报人：张豪

汇报时间：2022年9月1日

目录
CONTENTS

PART 01

文章概览

背景介绍——决策树隐私保护算法

本文是在“Securely and efficiently outsourcing decision tree inference,”[1] 基础上进行了算法/协议的改进

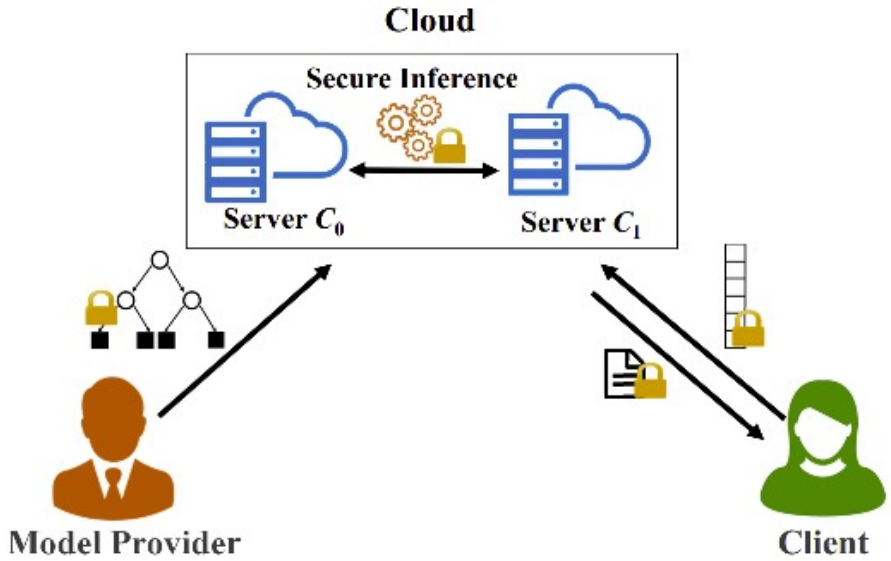
主要工作:对决策树算法inference过程进行隐私保护，在半诚实假定下，作者提出了一个安全高效的两方协议框架

作者的出发点是基于轻量的基于加性秘密共享算法来实现，没有考虑上计算复杂的加密方法（如（全）同态加密、混淆电路等）

[1]Y. Zheng, H. Duan, C. Wang, R. Wang, and S. Nepal, “Securely and efficiently outsourcing decision tree inference,” IEEE Trans. Dependable Secure Comput., pp. 1–1, 2020.

场景：

我们考虑两方的场景，Client有需要分类的私有数据，Provirder拥有训练好的决策树模型



隐私问题：

Provider 将自己训练好的模型视为一种财产，不希望自己的模型暴露在明文空间下。

Client 担心数据隐私泄漏，被Provider获取到。

概览——Our Contributions

1. 我们提出了一个安全且有效的**决策树预测隐私计算框架**，该框架可以保护模型的隐私和输入数据的隐私，而且不需要Provider 和 Client在线状态进行交互。

2. 我们开发了一个量身定制的协议，该协议仅在在线执行过程中利用轻量级的加性秘密共享，并具有正式的安全证明。

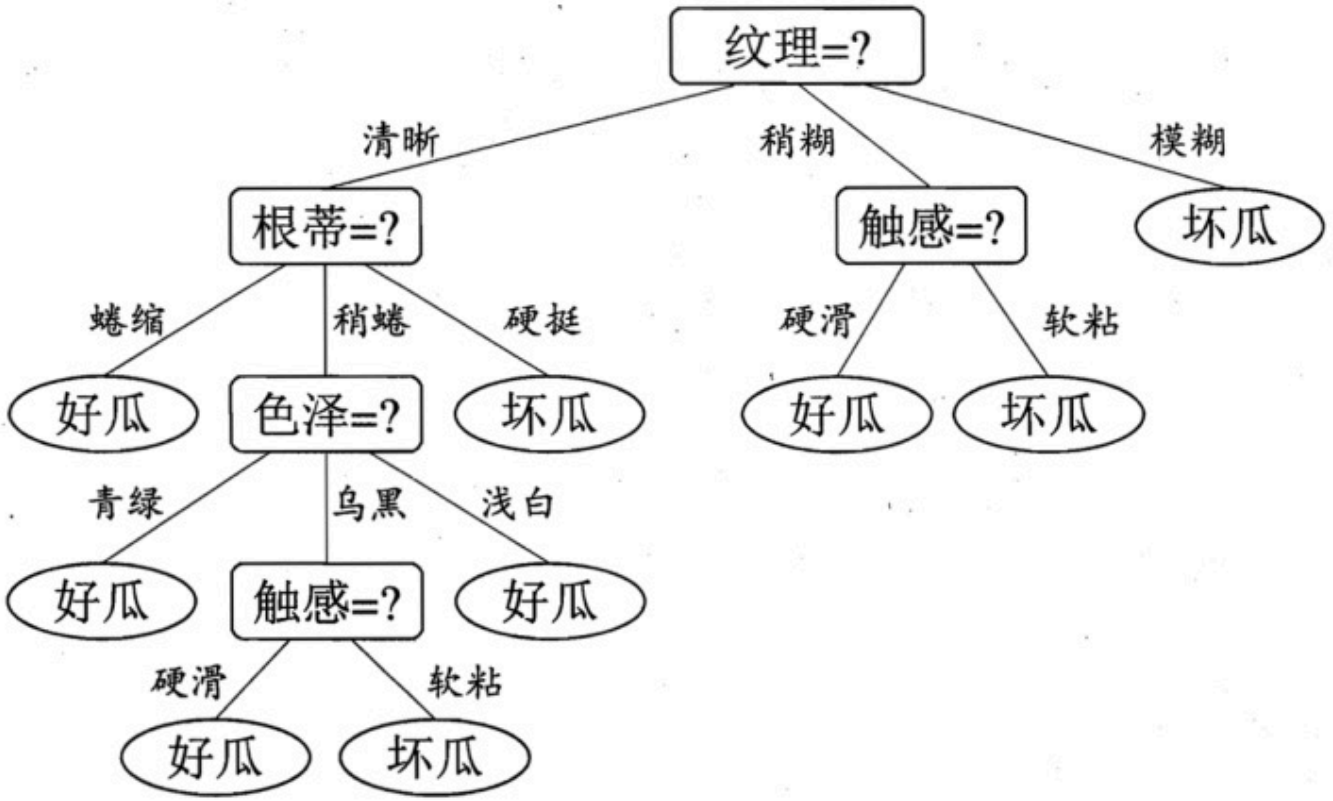
3. 我们对提出的决策树进行了实验，展示了我们安全设计的性能，以及为客户带来的性能优势（计算量优化了 4 个数量级，通信量优化了 163 个数量级），而不是之前工作中非外包设定。

PART 02

预备知识

预备知识——决策树算法

决策树算法：

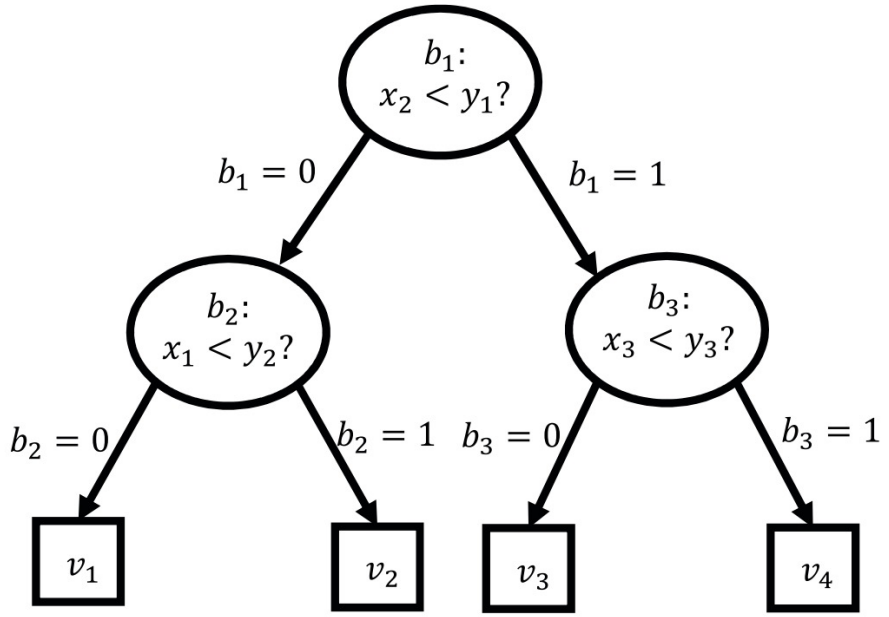


编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是

图片摘自《机器学习》周志华

预备知识——决策树算法

决策树算法：



Notation	Description
$\mathbf{x}$	Feature vector
$\mathbf{y}$	Threshold vector
x_i	The i-th feature in the feature vector
y_j	The threshold at decision node $\mathcal{D}_j$
d	Depth of a decision tree
J	Number of decision nodes
I	Dimension of feature vector

决策树中决策节点称为Decision Node $\mathcal{DN}$ 其上有常数 y_j 表示阈值

给定 m 个决策点，我们因此有阈值向量 $\mathbf{y} = \{y_1, \ldots, y_m\}$

每个Leaf node $\mathcal{LN}_k$ 都有一个预测值 v_k 表示决策树的分类结果

其中 K 表示叶子结点的数量。

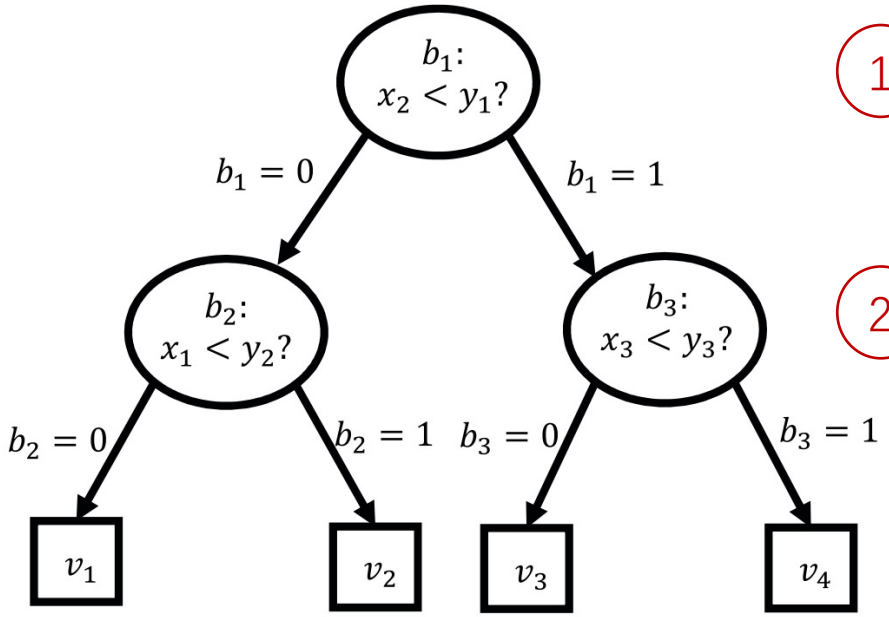
用 $\mathbf{x} = \{x_0, \cdots, x_{I-1}\}$ 表示输入的特征向量

基于映射关系 $\sigma : j \in \{0, 1, \cdots, J-1\} \rightarrow i \in \{0, 1, \cdots, I-1\}$

M. F. Delgado, E. Cernadas, S. Barro, and D. G. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" J. Mach. Learn. Res., vol. 15, no. 1, pp. 3133–3181, 2014

预备知识——决策树算法

决策树算法：



1 基于映射关系 $\sigma : j \in \{0, 1, \cdots, J-1\} \rightarrow i \in \{0, 1, \cdots, I-1\}$

Secure Feature Selection

2 $$f(x_{\sigma(j)}) = (x_{\sigma(j)} < y_j) = b_j$$

Secure Decision Node Evaluation

根据结果 b_j 决定是往左分枝(b_j =0)还是往右分枝(b_j =1)

3 持续上面步骤，直到走到叶子结点结束

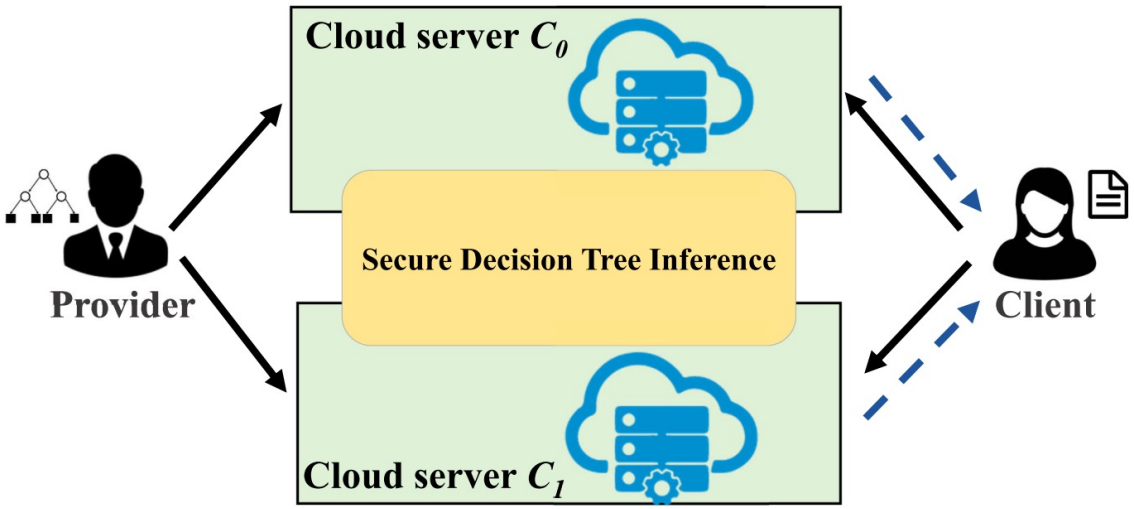
Secure Inference Generation

注：决策树的结构不在保护范围内，**隐藏决策树真实结构**，不失一般性的我们假设本文中的**决策树是完全二叉树**类型，如果不是完全二叉树类型，也可以通过添加dummy node 转化为完全二叉树。

PART 03

方案设计

方案设计——系统架构



Provider是决策树模型的拥有者，可以向Client提供 inference 服务。但是Provider不希望把自己的模型放在公开的明文环境下，防止模型被窃取攻击。实际生活中Provider可以是医疗结构、金融应用。

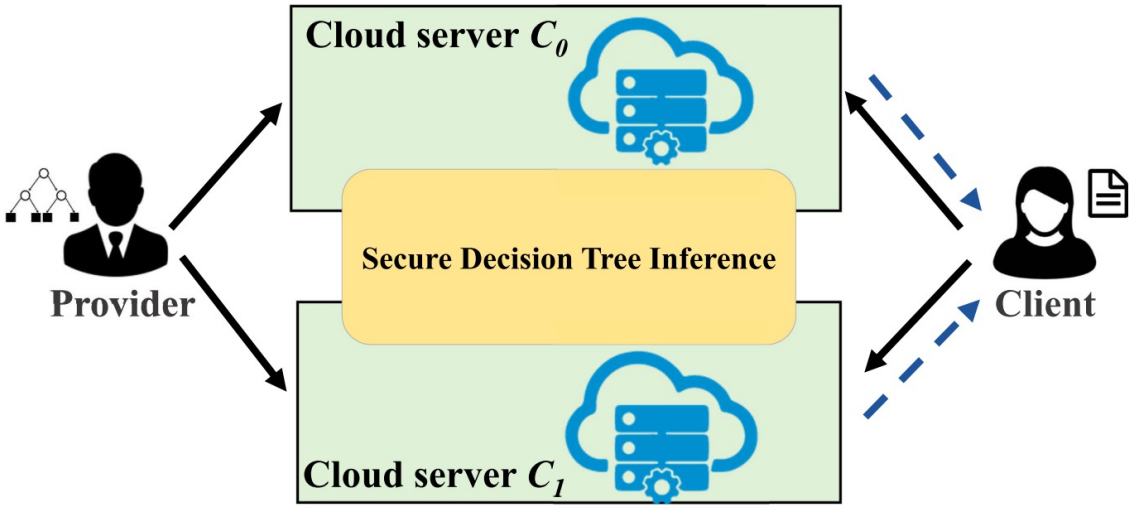
Client可以理解成用户，持有自己的隐私数据作为特征向量输入。Client可以是病人担心自己的私密数据（如体重、身高、血型）被泄漏。此外，在整个推理过程中，Client是希望处于一个**离线状态**（即把数据传到云上即可，而不是在线进行交互）。

系统架构中模型有C0和C1两个独立的服务器。

Client加密特征向量，split到两个secret share，并上传到C0和C1中；Provider也同样share他的模型，部署到在两个服务器上。

两台服务器C0和C1在接收到share后，开始运行隐私计算决策树协议，生成最终的加密预测结果并返回。

方案设计——威胁模型



在本文中，我们假定威胁模型是一个半诚实的设定。

我们考虑三个实体（cloud server、Provider、Client）可能被adversary进行Corrupted。但上面的参与方两两不会合谋。

具体来讲，Client希望自己保持自己的输入的特征向量 x 和预测的最终结果隐私的。

Provider希望保持自己的模型参数是隐私的（包括阈值向量y，映射函数 σ，叶子节点代表的预测值）。

注：我们对如下模型参数不进行保护：决策树的深度 d，特征向量维度 n，比特位长度 l（表示特征向量和阈值的元素值）。这些参数是公开的。

D. J. Wu, T. Feng, M. Naehrig, and K. E. Lauter, "Privately evaluating decision trees and random forests," PoPETs, vol. 2016, no. 4,pp. 335–355, 2016.

方案设计——安全特征选择

Secure Feature Selection

问题：如何实现**映射功能**同时又保证**隐私性**

$$\sigma : j \in \{1, 2, \ldots, m\} \rightarrow s \in \{1, 2, \ldots, n\}$$

解决方案：利用矩阵✖向量，实现映射功能

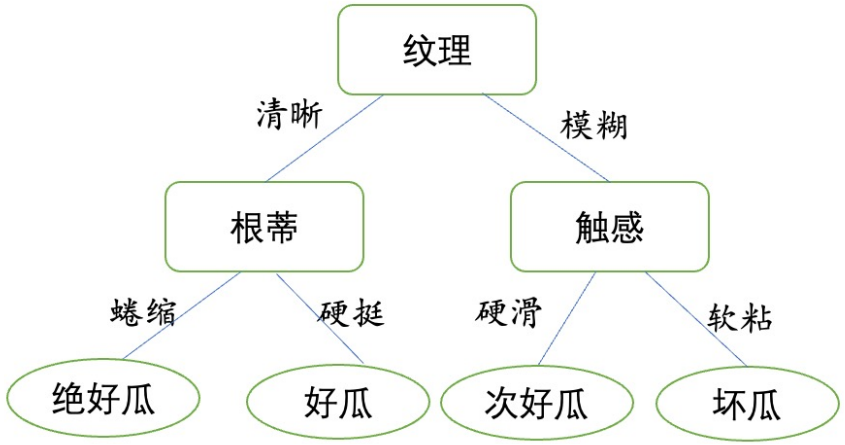
$$m \times n \text{ matrix } \mathbf{M}$$

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是

M矩阵每一行向量τ，用二进制表示

$\sigma(j)$设为1，其余位置全设为0

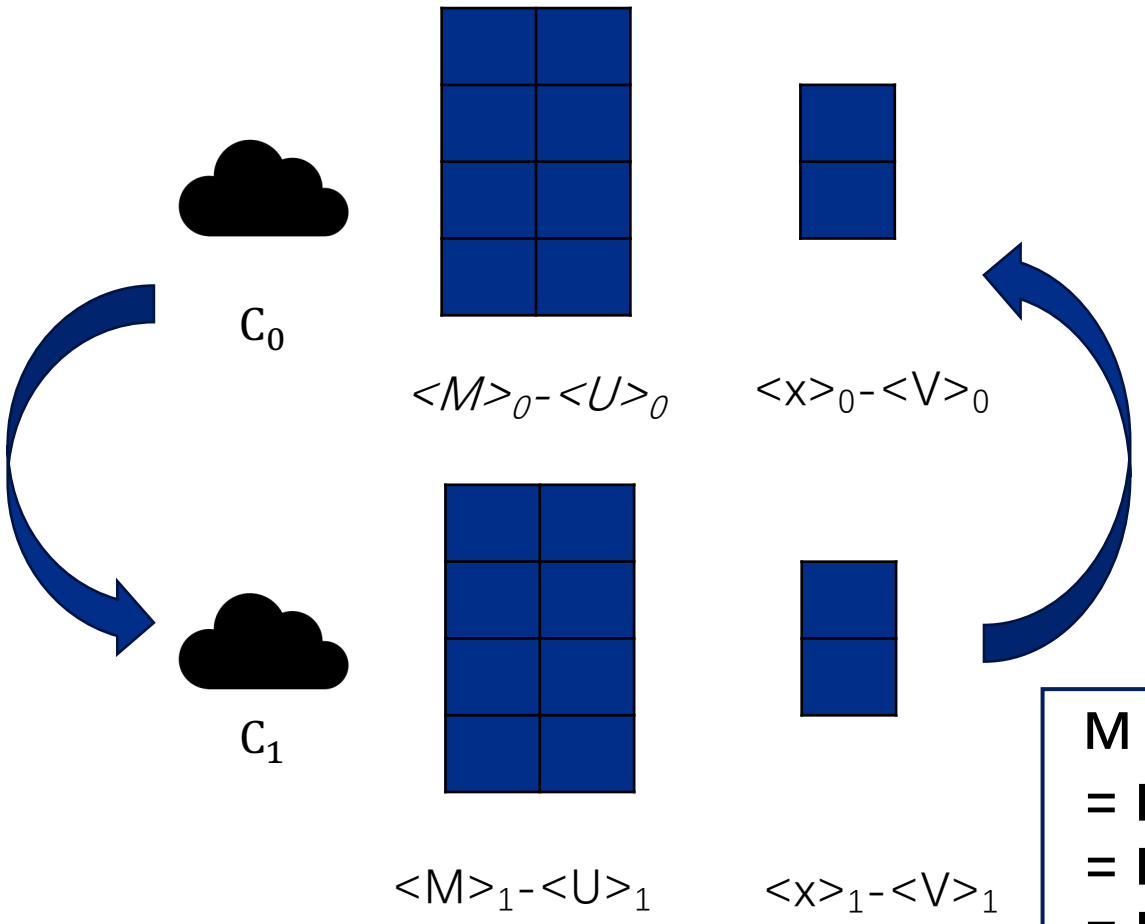
$$x_{\sigma(j)} = \boldsymbol{\tau}_j \mathbf{x} \ (j \in \{1, \ldots, m\})$$



$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 7 \\ 5 \\ 6 \\ 8 \\ 3 \\ 4 \end{pmatrix}$$

$\mathbf{M}$ $\mathbf{x}$

方案设计——安全特征选择



- Beaver Tripes：U，V，Z（Z=U*V）

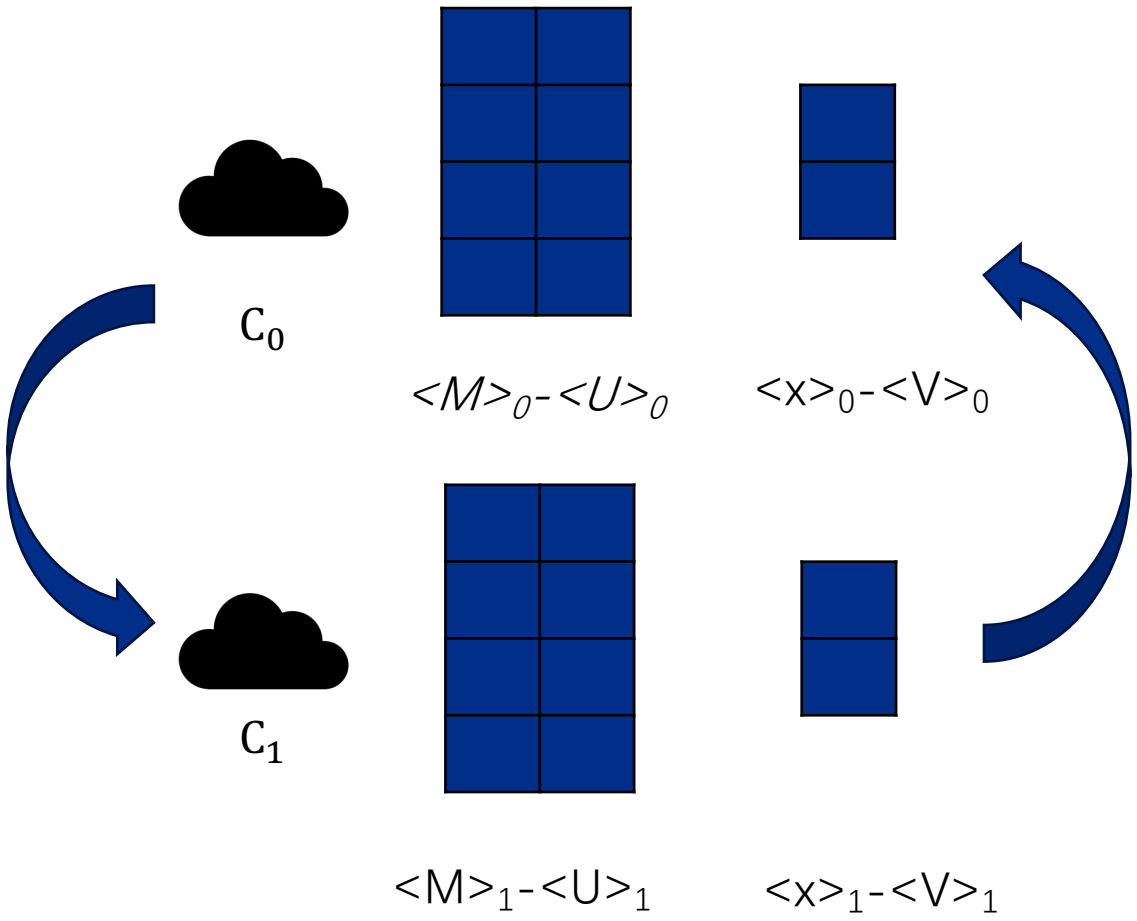
Reconstruct

E = M - U

F = x - V

$$
\begin{aligned}
M * x &= (E + U)*(F + V) \\
&= EF + EV + UF + UV \\
&= EF + E(<V>_0 + <V>_1) + (<U>_0 + <U>_1)F + Z \\
&= \underbrace{EF}_{public} + \underbrace{(E<V>_0 + <U>_0F + <Z>_0)}_{C0} + \underbrace{(E<V>_1 + <U>_1F + <Z>_1)}_{C1}
\end{aligned}
$$

方案设计——安全特征选择



- Beaver Tripes：U，V，Z（Z=U*V）

$(\mathbf{G}_1, \mathbf{g}_2, \mathbf{g}_3)$, where $\mathbf{g}_3 = \mathbf{G}_1 \cdot \mathbf{g}_2$

Input: Secret sharings $[\mathbf{M}]$ and $[\mathbf{x}]$.
Output: Secret sharing $[\mathbf{x}_\sigma] = [\mathbf{M} \cdot \mathbf{x}]$.

1: $\mathcal{C}_i$ computes $[\mathbf{E}]_i = [\mathbf{M}]_i - [\mathbf{G}_1]_i$ and $[\mathbf{f}]_i = [\mathbf{x}]_i - [\mathbf{g}_2]_i$.
2: $\mathcal{C}_0$ and $\mathcal{C}_1$ jointly reconstruct $\mathbf{E}$ and $\mathbf{f}$.
3: $\mathcal{C}_i$ computes $[\mathbf{M} \cdot \mathbf{x}]_i = i \cdot \mathbf{E} \cdot \mathbf{f} + \mathbf{E} \cdot [\mathbf{g}_2]_i + [\mathbf{G}_1]_i \cdot \mathbf{f} + [\mathbf{g}_3]_i$.

Fig. 3. Secure input selection.

方案设计——安全特征选择（改进）

北京郵電大學 Beijing University of Posts and Telecommunications

Secure Feature Selection

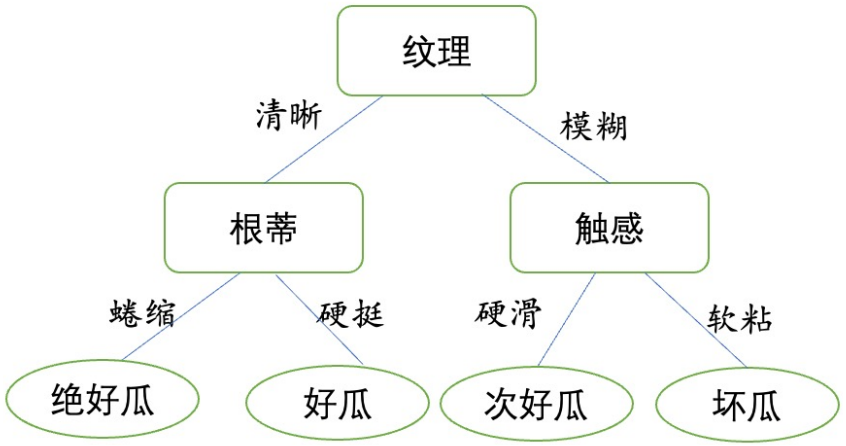
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是

问题：矩阵乘法所造成，协议有O(m*n)的时间复杂度

改进思路：压缩空间提高信息密度，避免使用矩阵，转化为安全数组访问

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 7 \\ 5 \\ 6 \\ 8 \\ 3 \\ 4 \end{pmatrix} \rightarrow [3 \quad 1 \quad 5] \quad \begin{pmatrix} 7 \\ 5 \\ 6 \\ 8 \\ 3 \\ 4 \end{pmatrix}$$

M　　x　　Index　　x

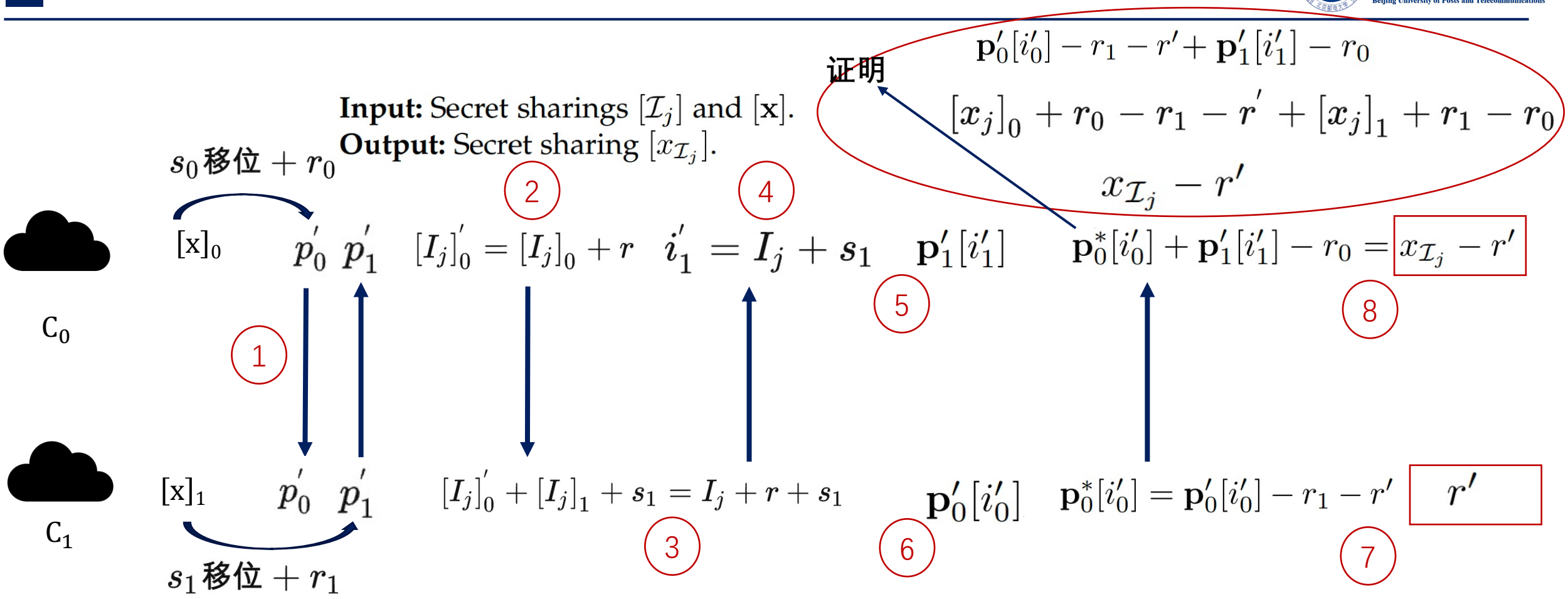


问题归结为茫然数组访问（oblivious array-entry read）/安全数组访问问题：

在加密数组中根据加密索引值茫然选择数组项

M. Curran, X. Liang, H. Gupta, O. Pandey, and S. R. Das, "Procsa: Protecting privacy in crowdsourced spectrum allocation," in Proc. of ESORICS, 2019.

方案设计——安全特征选择（改进）



$$\mathbf{p}'_m[i] = \mathbf{p}_m[i^*_m] + r_m$$

$$i^*_m = ((i + s_m) \bmod 2^l) \bmod I$$

方案设计——安全特征选择（改进）

安全输入选择改进:

Input: Secret sharings $[\mathcal{I}_j]$ and $[\mathbf{x}]$.
Output: Secret sharing $[x_{\mathcal{I}_j}]$.

1: Each $\mathcal{C}_m$ creates an array $\mathbf{p}'_m$ where the i-th element is $\mathbf{p}'_m[i] = \mathbf{p}_m[i^*_m] + r_m$. Here, $s_m \leftarrow \mathbb{Z}_{2^l}$ and $r_m \leftarrow \mathbb{Z}_{2^l}$ are random values chosen by $\mathcal{C}_m$; and $i^*_m = ((i + s_m) \bmod 2^l) \bmod I$.
2: $\mathcal{C}_0$ chooses a random value $r \leftarrow \mathbb{Z}_{2^l}$ and sends $[\mathcal{I}_j]'_0 = [\mathcal{I}_j]_0 + r$ to $\mathcal{C}_1$.
3: $\mathcal{C}_1$ computes $[\mathcal{I}_j]'_0 + [\mathcal{I}_j]_1 + s_1 = \mathcal{I}_j + r + s_1$ and sends it to $\mathcal{C}_0$.
4: $\mathcal{C}_0$ removes r and produces $i'_1 = ((\mathcal{I}_j + s_1) \bmod 2^l) \bmod I$.
5: $\mathcal{C}_0$, with i'_1 as input, acts as the receiver to run an OT protocol with $\mathcal{C}_1$ to obtain $\mathbf{p}'_1[i'_1]$.
6: $\mathcal{C}_1$, in a symmetric manner following Steps 2-5, obtains $\mathbf{p}'_0[i'_0]$, where $i'_0 = ((\mathcal{I}_j + s_0) \bmod 2^l) \bmod I$.
7: $\mathcal{C}_1$ chooses a random value $r' \leftarrow \mathbb{Z}_{2^l}$ and sends $\mathbf{p}^*_0[i'_0] = \mathbf{p}'_0[i'_0] - r_1 - r'$ to $\mathcal{C}_0$. Also, $\mathcal{C}_1$ sets r' as its share $[x_{\mathcal{I}_j}]_1$ for the expected feature $x_{\mathcal{I}_j}$.
8: $\mathcal{C}_0$ computes $\mathbf{p}^*_0[i'_0] + \mathbf{p}'_1[i'_1] - r_0 = x_{\mathcal{I}_j} - r'$ and sets the result as its share $[x_{\mathcal{I}_j}]_0$ for $x_{\mathcal{I}_j}$.

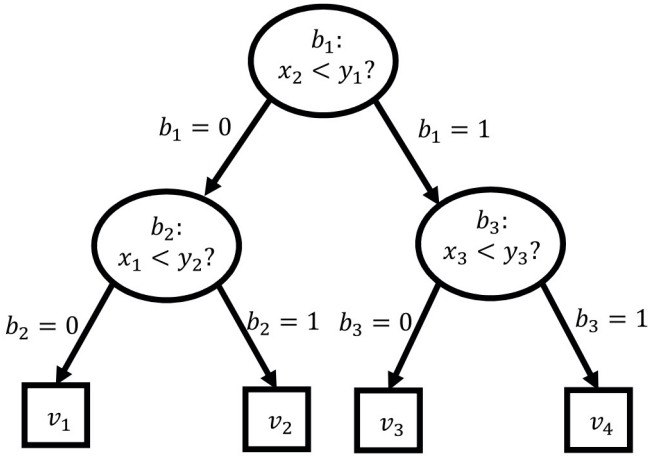
方案设计——安全决策点判定

北京郵電大學 Beijing University of Posts and Telecommunications

Secure Decision Node Evaluation　　求：MSB(x-y)

Input: Secret sharings $[y_j]$ and $[x_{\sigma(j)}]$.
Output: Secret sharing $[b_j]$.

1: $\mathcal{C}_i$ computes $[a]_i = [y_j]_i - [x_{\sigma(j)}]_i$.
// Bit extraction ($\langle\cdot\rangle$ denotes sharing over $\mathbb{Z}_2$)
2: Let p denote $\mathcal{C}_0$'s share $[a]_0$, with the bit string being $p_l, \cdots, p_1$. Let q denote $\mathcal{C}_1$'s share $[a]_1$, with the bit string being $q_l, \cdots, q_1$. Define the secret sharing $\langle w_k \rangle$ over $\mathbb{Z}_2$ as $\{\langle w_k \rangle_0 = p_k, \langle w_k \rangle_1 = q_k\}$. Also, define $\langle p_k \rangle$ as $(\langle p_k \rangle_0 = p_k, \langle p_k \rangle_1 = 0)$ and $\langle q_k \rangle$ as $\{\langle q_k \rangle_0 = 0, \langle q_k \rangle_1 = q_k\}$.
3: $\mathcal{C}_0$ and $\mathcal{C}_1$ compute $\langle c_1 \rangle = \langle p_1 \rangle \cdot \langle q_1 \rangle$.
4: For $k \in [2, \cdots, l-1]$,
a) $\mathcal{C}_0$ and $\mathcal{C}_1$ compute $\langle d_k \rangle = \langle p_k \rangle \cdot \langle q_k \rangle + 1$.
b) $\mathcal{C}_0$ and $\mathcal{C}_1$ compute $\langle e_k \rangle = \langle w_k \rangle \cdot \langle c_{k-1} \rangle + 1$.
c) $\mathcal{C}_0$ and $\mathcal{C}_1$ compute $\langle c_k \rangle = \langle e_k \rangle \cdot \langle d_k \rangle + 1$.
5: $\mathcal{C}_0$ and $\mathcal{C}_1$ compute $\langle a_l \rangle = \langle w_l \rangle + \langle c_{l-1} \rangle$, with $\langle a_l \rangle$ defined as $\{\langle a_l \rangle_0 = t_1, \langle a_l \rangle_1 = t_2\}$.
// Conversion from $\mathbb{Z}_2$ to $\mathbb{Z}_{2^l}$
6: Let $[t_1]$ over $\mathbb{Z}_{2^l}$ be defined as $[t_1]_0 = t_1, [t_1]_1 = 0\}$ and $[t_2]$ as $\{[t_2]_0 = 0, [t_2]_1 = t_2\}$.
7: $\mathcal{C}_0$ and $\mathcal{C}_1$ compute $[a_l] = [t_1] + [t_2] - 2 \cdot [t_1] \cdot [t_2]$.
8: Output $[b_j] = [a_l]$.

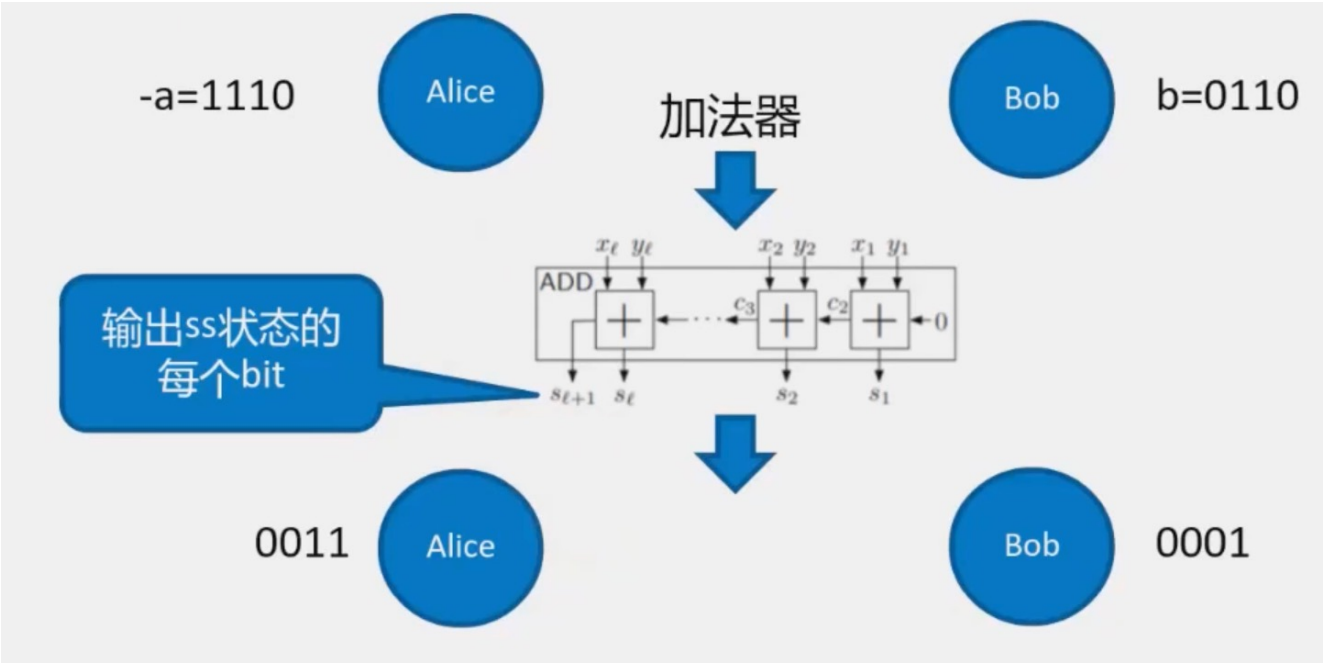


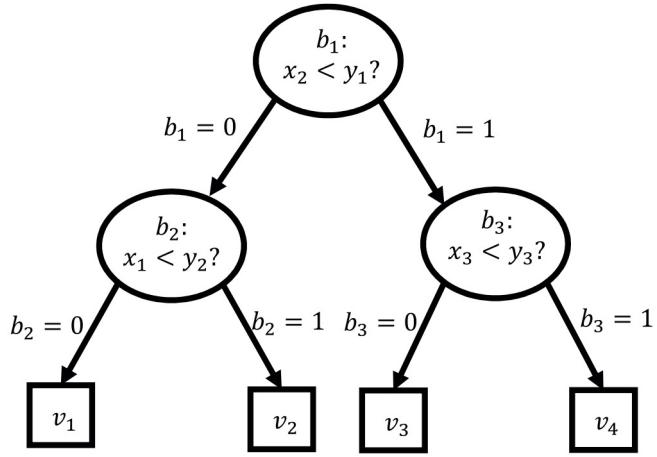
M. D. Cock et al., "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation," IEEE Trans. Depend. Secure Comput., vol. 16, no. 2, pp. 217–230, Mar./Apr. 2017.

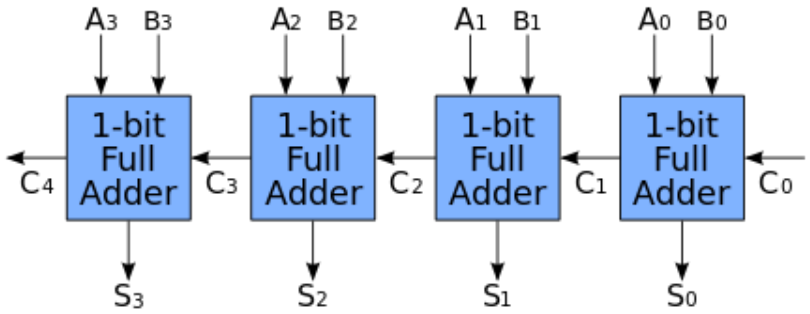
方案设计——安全决策点判定

Secure Decision Node Evaluation

计算Compare(x,y)的简单方案：执行布尔电路加法器以得到MSB(x-y)







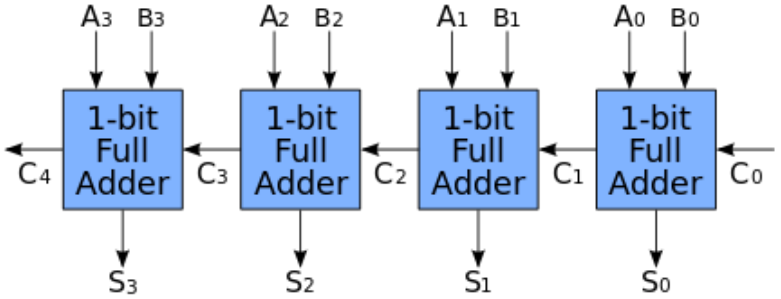
M. D. Cock et al., "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation," IEEE Trans. Depend. Secure Comput., vol. 16,no. 2, pp. 217–230, Mar./Apr. 2017.

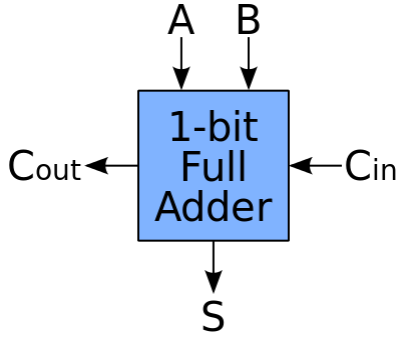
方案设计——安全决策点判定（改进）

Secure Decision Node Evaluation

A	B	Cin	Cout	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

波纹进位加法器：





$$C_{out} = (A \cdot B) + (A \oplus B) \cdot C_{in}$$

Generate carry *Propagate carry*

$$C_{out} = G + P \cdot C_{in}$$

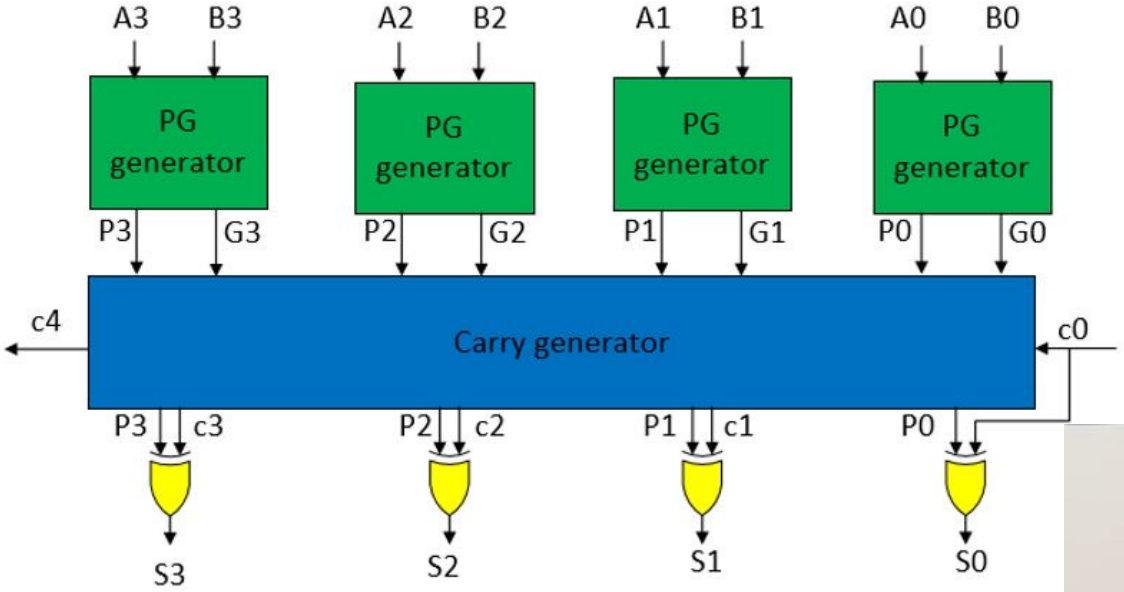
$$c_{i+1} = G_i + P_i \cdot c_i$$

1) $c_1 = G_0 + P_0 \cdot c_0 = G_0;$
2) $c_2 = G_1 + P_1 \cdot c_1 = G_1 + P_1 \cdot G_0;$
3) $c_3 = G_2 + P_2 \cdot c_2 = G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0);$
4) $c_4 = G_3 + P_3 \cdot c_3 = G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0)).$

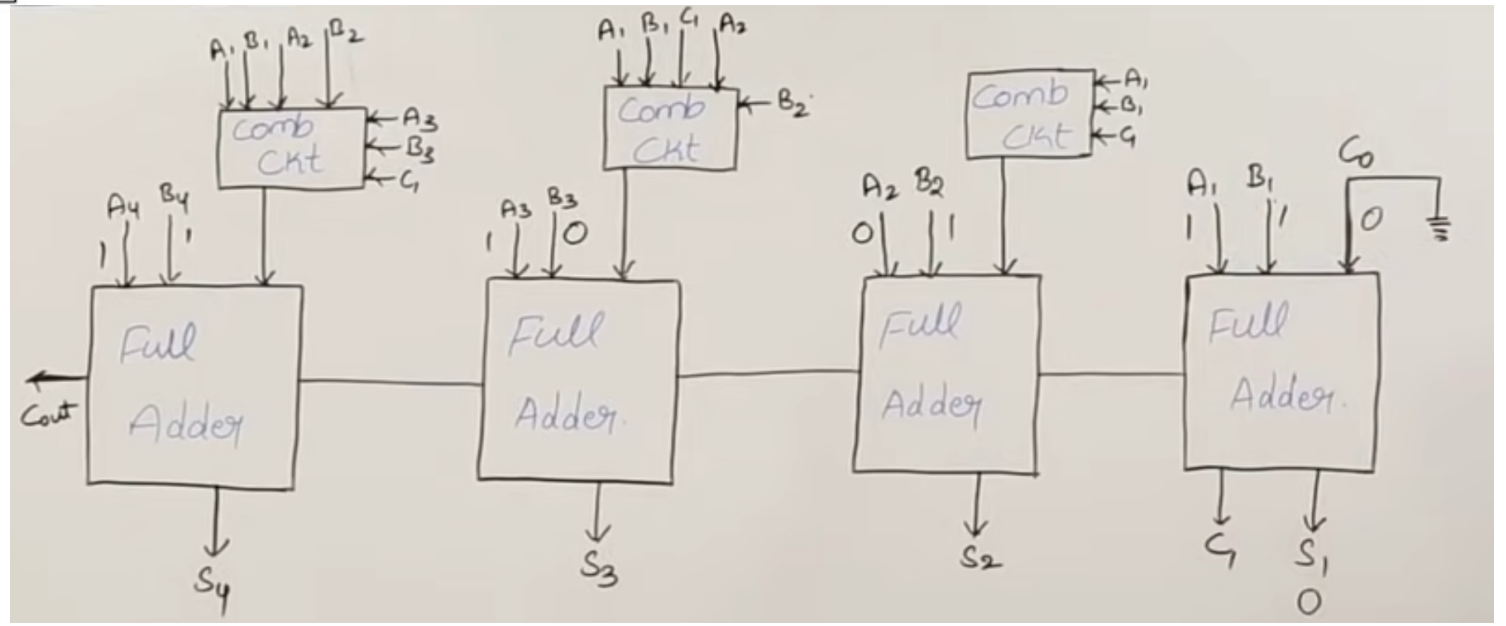
方案设计——安全决策点判定（改进）

Secure Decision Node Evaluation

超前进位加法器：



1) $c_1 = G_0 + P_0 \cdot c_0 = G_0;$
2) $c_2 = G_1 + P_1 \cdot c_1 = G_1 + P_1 \cdot G_0;$
3) $c_3 = G_2 + P_2 \cdot c_2 = G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0);$
4) $c_4 = G_3 + P_3 \cdot c_3 = G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0)).$



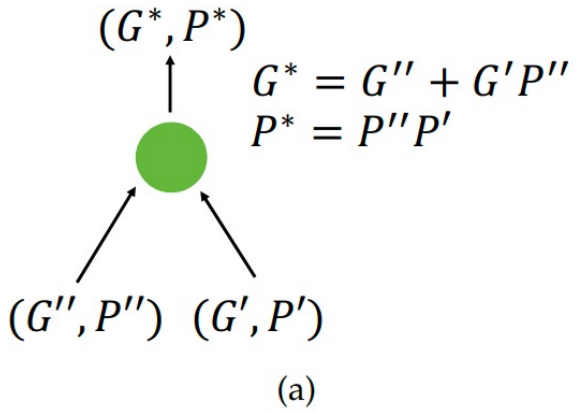
方案设计——安全决策点判定（改进）

北京郵電大學 Beijing University of Posts and Telecommunications

Secure Decision Node Evaluation

定义运算 binary operator $\diamond$

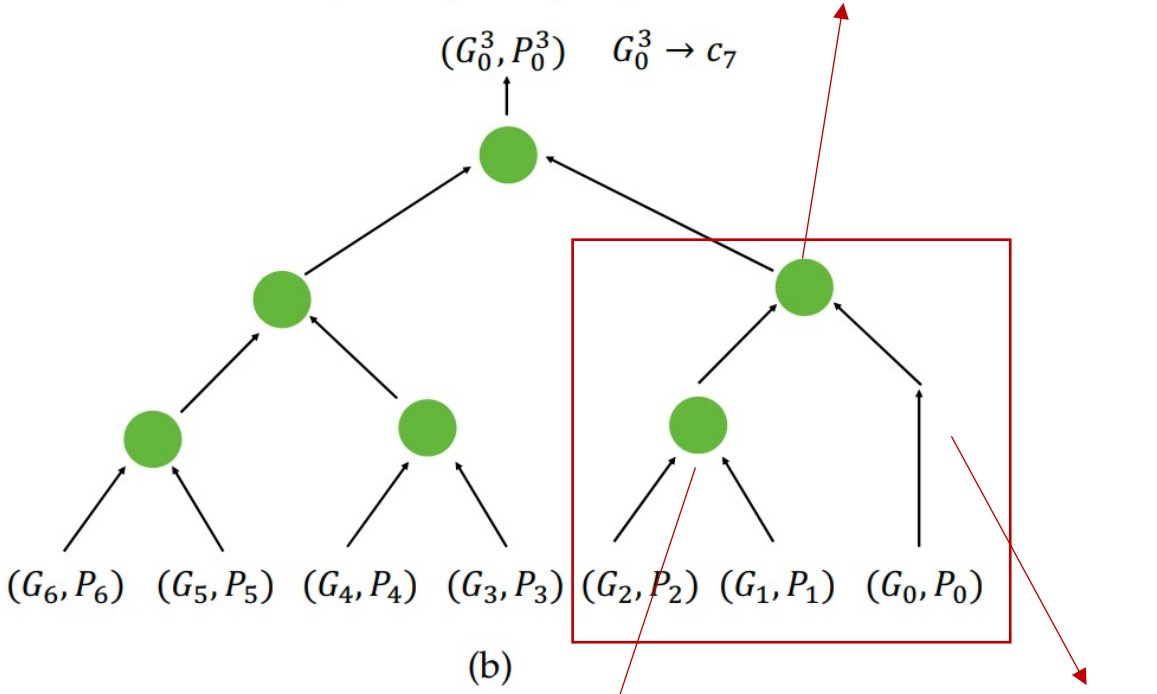
$$(G^*, P^*) = (G'', P'') \diamond (G', P')$$



(a)

1) $c_1 = G_0 + P_0 \cdot c_0 = G_0;$
2) $c_2 = G_1 + P_1 \cdot c_1 = G_1 + P_1 \cdot G_0;$
3) $c_3 = G_2 + P_2 \cdot c_2 = G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0);$
4) $c_4 = G_3 + P_3 \cdot c_3 = G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0)).$

$$G_0^2 = \bar{G}_1^1 + G_0^1 \cdot P_1^1 = G_2 + \bar{G}_1 \cdot P_2 + G_0 \cdot P_2 P_1$$



(b)

$(G_1^1, P_1^1) = (G_2, P_2) \diamond (G_1, P_1)$ $(G_0^1, P_0^1) = (G_0, P_0)$

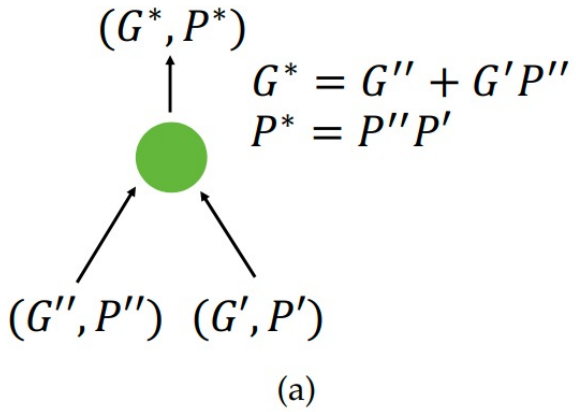
方案设计——安全决策点判定（改进）

北京郵電大學 Beijing University of Posts and Telecommunications

Secure Decision Node Evaluation

定义运算 binary operator $\diamond$

$$(G^*, P^*) = (G'', P'') \diamond (G', P')$$



(a)

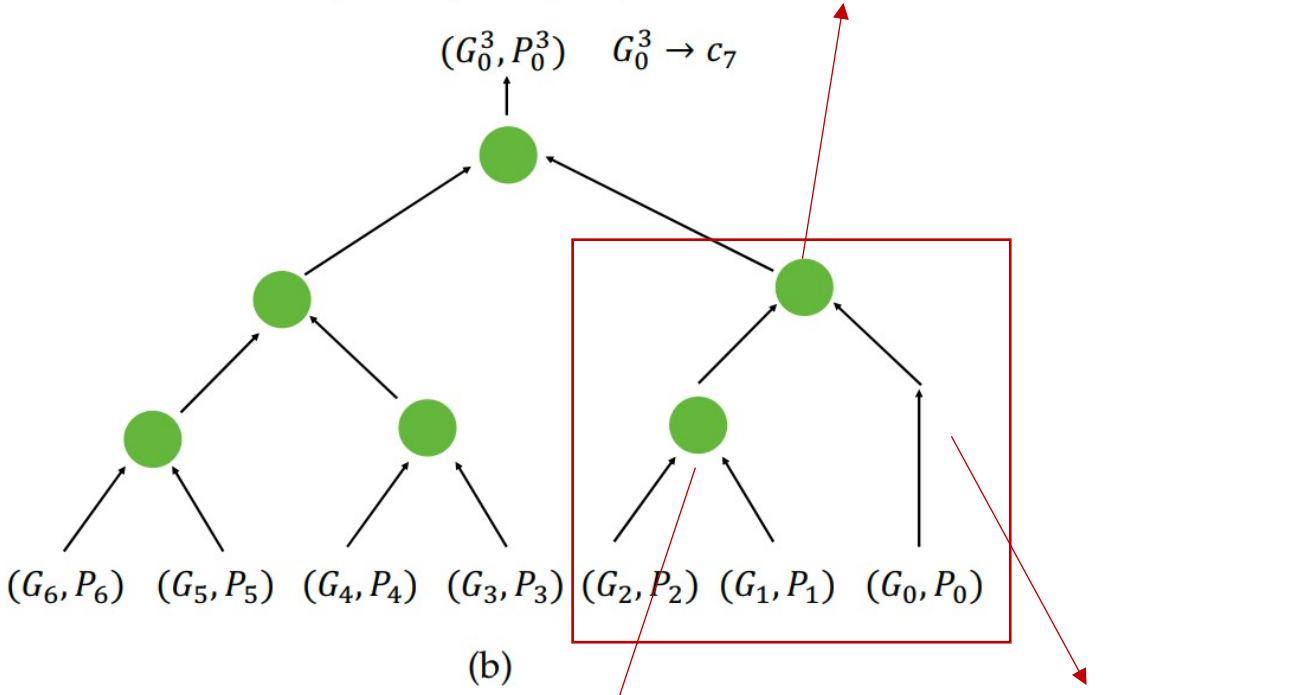
密文状态：

$$(\langle G^* \rangle, \langle P^* \rangle) = (\langle G'' \rangle, \langle P'' \rangle) \tilde{\diamond} (\langle G' \rangle, \langle P' \rangle)$$

1) $c_1 = G_0 + P_0 \cdot c_0 = G_0;$
2) $c_2 = G_1 + P_1 \cdot c_1 = G_1 + P_1 \cdot G_0;$
3) $c_3 = G_2 + P_2 \cdot c_2 = G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0);$
4) $c_4 = G_3 + P_3 \cdot c_3 = G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0)).$

例：

$$G_0^2 = \bar{G}_1^1 + G_0^1 \cdot P_1^1 = G_2 + \bar{G}_1 \cdot P_2 + G_0 \cdot P_2 P_1$$



(b)

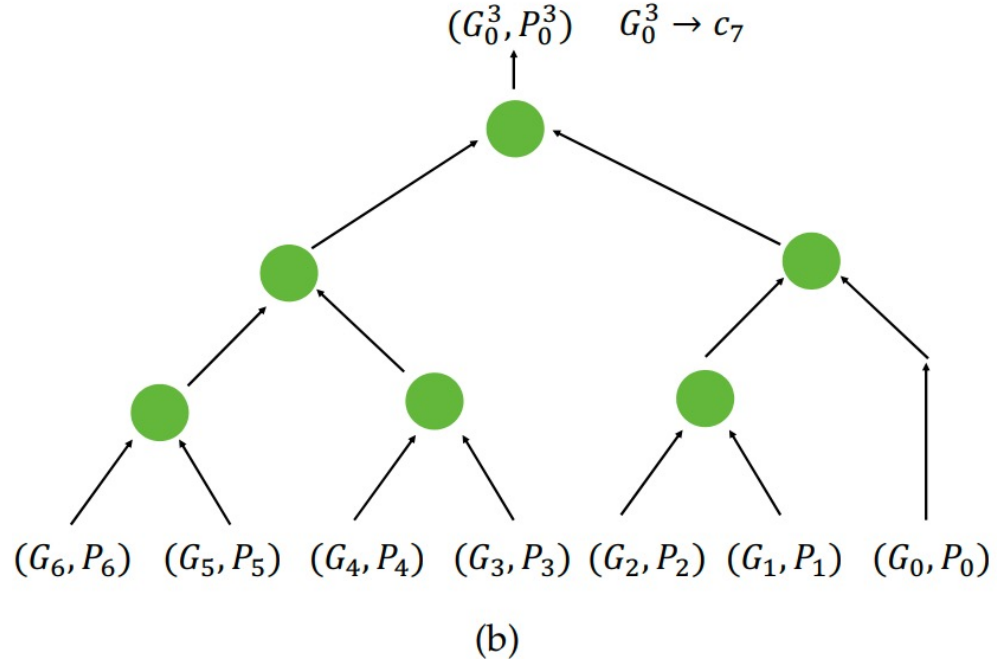
$(G_1^1, P_1^1) = (G_2, P_2) \diamond (G_1, P_1)$ $(G_0^1, P_0^1) = (G_0, P_0)$

方案设计——安全决策点判定（改进）

Secure Decision Node Evaluation

Input: Secret sharings $[y_j]$ and $[x_{\mathcal{I}_j}]$.
Output: Secret sharing $\langle v_j \rangle$.

1: $\mathcal{C}_m$ computes $[\Delta]_m = [y_j]_m - [x_{\mathcal{I}_j}]_m$.
// Secure MSB extraction (with $l = 64$ assumed; $\langle \cdot \rangle$ denotes sharing over $\mathbb{Z}_2$)
2: Let a (resp. b) represent the share $[\Delta]_0$ (resp. $[\Delta]_1$), with the bit string being $a_{l-1}, \cdots, a_0$ (resp. $b_{l-1}, \cdots, b_0$). Let $\langle a_q \rangle$ be defined as $(\langle a_q \rangle_0 = a_q, \langle a_q \rangle_1 = 0)$ and $\langle b_q \rangle$ as $\{\langle b_q \rangle_0 = 0, \langle b_q \rangle_1 = b_q\}$, where $q \in [0, l-1]$. Also, let $\langle w_q \rangle$ be defined as $\{\langle w_q \rangle_0 = a_q, \langle w_q \rangle_1 = b_q\}$.
// Setup round for secure carry computation (*SCC*):
3: Compute $\langle G_q \rangle = \langle a_q \rangle \cdot \langle b_q \rangle$, for $q \in [0, l-1]$
4: Compute $\langle P_q \rangle = \langle a_q \rangle + \langle b_q \rangle$, for $q \in [0, l-1]$
// *SCC* round 1 (with $l = 64$ as example):
5: Compute $(\langle G_0^1 \rangle, \langle P_0^1 \rangle) = (\langle G_0 \rangle, \langle P_0 \rangle)$
6: For $k \in \{1, \cdots, 31\}$
a) Compute $(\langle G_k^1 \rangle, \langle P_k^1 \rangle) = (\langle G_{2 \cdot k} \rangle, \langle P_{2 \cdot k} \rangle) \tilde{\diamond} (\langle G_{2 \cdot k-1} \rangle, \langle P_{2 \cdot k-1} \rangle)$
// *SCC* round 2:
7: For $k \in \{0, \cdots, 15\}$
a) Compute $(\langle G_k^2 \rangle, \langle P_k^2 \rangle) = (\langle G_{2 \cdot k+1}^1 \rangle, \langle P_{2 \cdot k+1}^1 \rangle) \tilde{\diamond} (\langle G_{2 \cdot k}^1 \rangle, \langle P_{2 \cdot k}^1 \rangle)$



(b)

Secure Decision Node Evaluation

// SCC round 3:

8: For $k \in \{0, \cdots, 7\}$

a) Compute $(\langle G_k^3 \rangle, \langle P_k^3 \rangle) = (\langle G_{2\cdot k+1}^2 \rangle, \langle P_{2\cdot k+1}^2 \rangle) \tilde{\diamond} (\langle G_{2\cdot k}^2 \rangle, \langle P_{2\cdot k}^2 \rangle)$

// SCC round 4:

9: For $k \in \{0, \cdots, 3\}$

a) Compute $(\langle G_k^4 \rangle, \langle P_k^4 \rangle) = (\langle G_{2\cdot k+1}^3 \rangle, \langle P_{2\cdot k+1}^3 \rangle) \tilde{\diamond} (\langle G_{2\cdot k}^3 \rangle, \langle P_{2\cdot k}^3 \rangle)$

// SCC round 5:

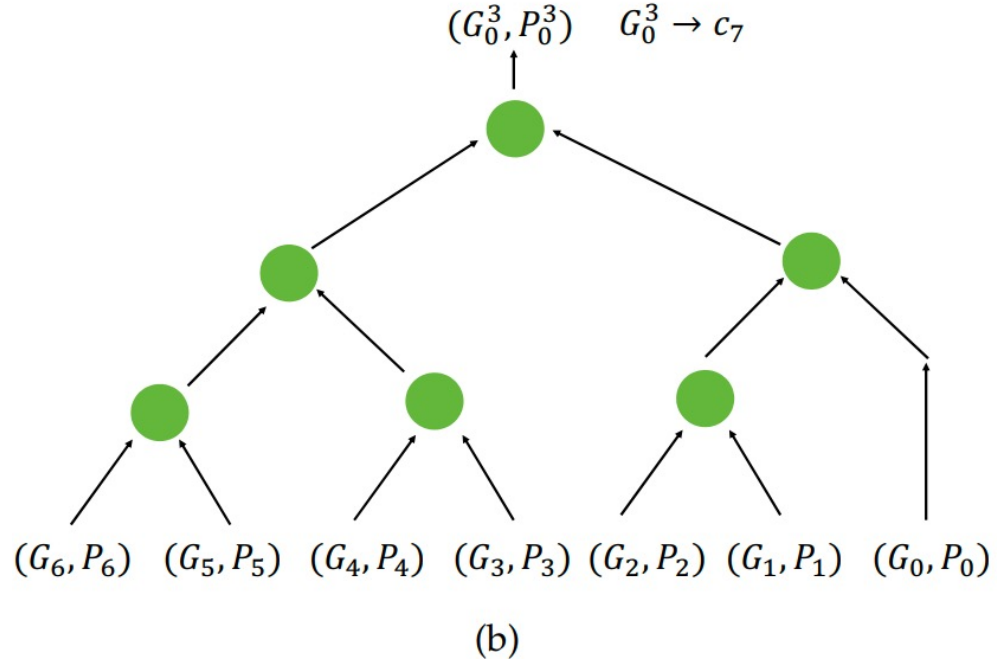
10: For $k \in \{0, 1\}$

a) Compute $(\langle G_k^5 \rangle, \langle P_k^5 \rangle) = (\langle G_{2\cdot k+1}^4 \rangle, \langle P_{2\cdot k+1}^4 \rangle) \tilde{\diamond} (\langle G_{2\cdot k}^4 \rangle, \langle P_{2\cdot k}^4 \rangle)$

// SCC round 6:

11: Compute $\langle G_0^6 \rangle = \langle G_1^5 \rangle + \langle G_0^5 \rangle \cdot \langle P_1^5 \rangle = \langle c_{l-1} \rangle$

12: Compute $\langle v_j \rangle = \langle w_{l-1} \rangle + \langle c_{l-1} \rangle$.



(b)

方案设计——安全决策点判定（改进）

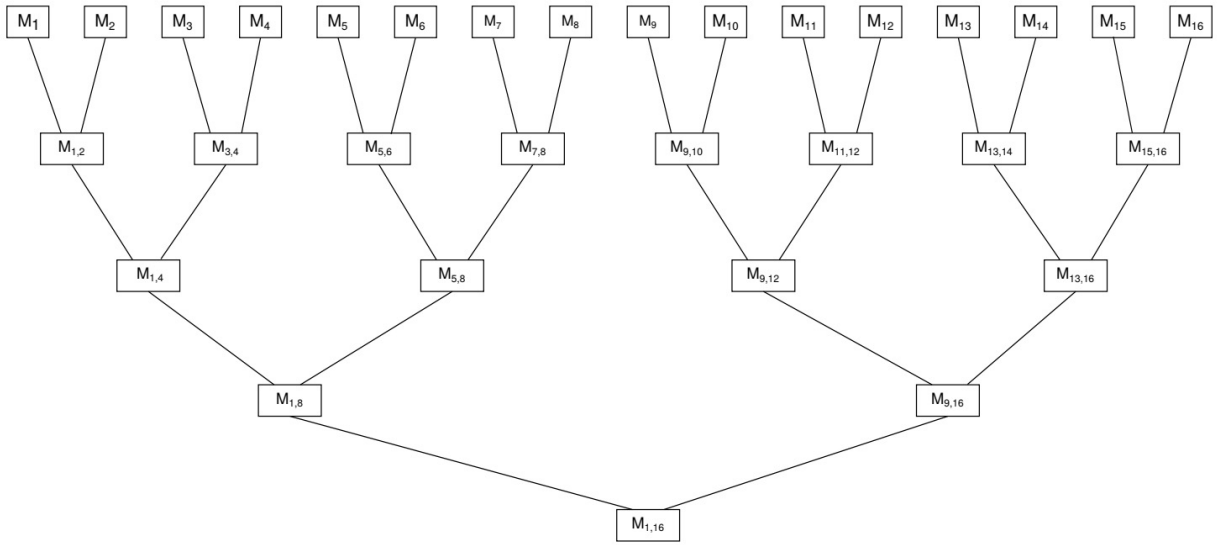
π_{BTX} bit extraction protocol

Protocol 7: Secure Protocol $\pi_{\mathsf{decompOPT}}$ for computing $\mathcal{F}_{\mathsf{decomp}}$ more efficiently.

Input : $[\![x]\!]_{2^\lambda}$
Output: $[\![x_1]\!]_2 ... [\![x_\lambda]\!]_2$

1 Party i regards its share x_i as $p_{i,1}, \ldots, p_{i,\lambda}$ s.t.
2 $[\![p_j]\!]_2, = p_{1,j} \oplus p_{2,j}$ for $j = 1, \ldots, \lambda$
3 Party 1 creates the sharing $[\![g_{1,j}]\!]_2 = (p_{1,j}, 0)$.
4 Party 2 creates the sharing $[\![g_{2,j}]\!]_2 = (0, p_{2,j})$.
5 $[\![g_j]\!]_2 \leftarrow [\![g_{1,j}]\!]_2 [\![g_{2,j}]\!]_2$
6 $[\![M_j]\!]_2 \leftarrow \begin{bmatrix} [\![p_j]\!]_2 & [\![g_j]\!]_2 \\ 0 & 1 \end{bmatrix}$ for all j
7 $\{[\![M_{1.j}]\!]_2 | 1 \leq j < \lambda\} \leftarrow \mathsf{ComposeNet}_\lambda([\![M]\!]_2)$
8 $[\![c_j]\!]_2 \leftarrow$ the upper right entry of $[\![M_{1.j}]\!]_2$
9 $[\![s_1]\!]_2 \leftarrow [\![p_1]\!]_2$
10 $[\![s_j]\!]_2 \leftarrow [\![p_j]\!]_2 \oplus [\![c_{j-1}]\!]_2$ for all $j > 1$
11 **return** $[\![s_1]\!]_2 ... [\![s_\lambda]\!]_2$

The protocol π_{BTX} requires $\lceil \log \alpha - 1) \rceil$ rounds and $2(\alpha - 1) + 4 \log(\alpha - 1) - 4$ bits of communication. Figure 1 shows an example circuit to compute the matrix composition phase for $\alpha = 17$. For batched inputs $\{x_1, \ldots, x_k\}$, the number of communication rounds remains the same and the total data transfer is scaled by k.



De Cock M, Dowsley R, Nascimento A C A, et al. High performance logistic regression for privacy-preserving genome analysis[J]. BMC Medical Genomics, 2021, 14(1): 1-18.

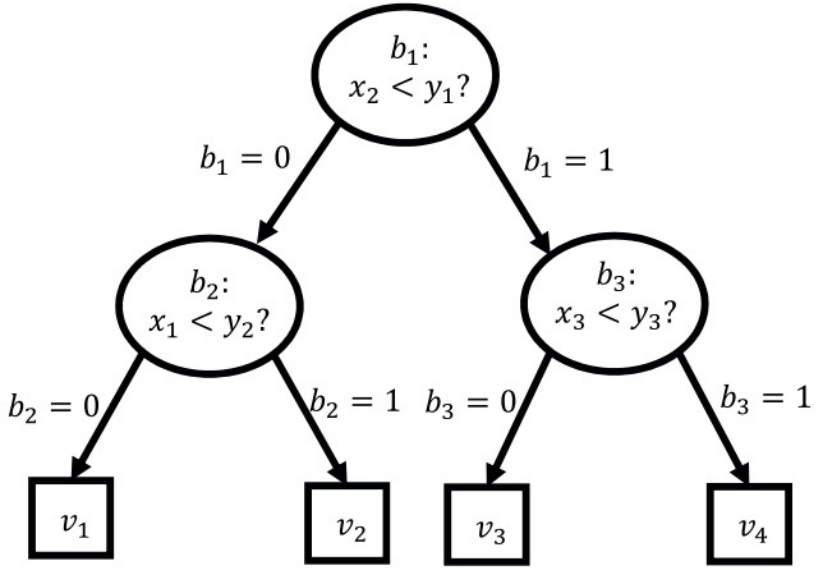
方案设计——安全预测生成

Secure Inference Result Generation

方案一 **Polynomial-Based Method** *[1]*

$O(1)$ communication complexity

$O(d \cdot 2^d)$ secret-shared multiplications



方案二 **Path Cost-Based Method** *[2]*

$O(2^d)$ communication complexity

两个方法，各有利弊；单就基于加性秘密共享方案（本文讨论）来看，方案一更优

[1] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in Proc. Conf. Netw. Distrib. Syst. Secur. Symp., 2015, pp. 1–14.

[2] R. K. H. Tai, J. P. K. Ma, Y. Zhao, and S. S. M. Chow, "Privacypreserving decision trees evaluation via linear functions," in Proc. Eur. Symp. Res. Comput. Secur., 2017, pp 494–512.

方案设计——安全预测生成

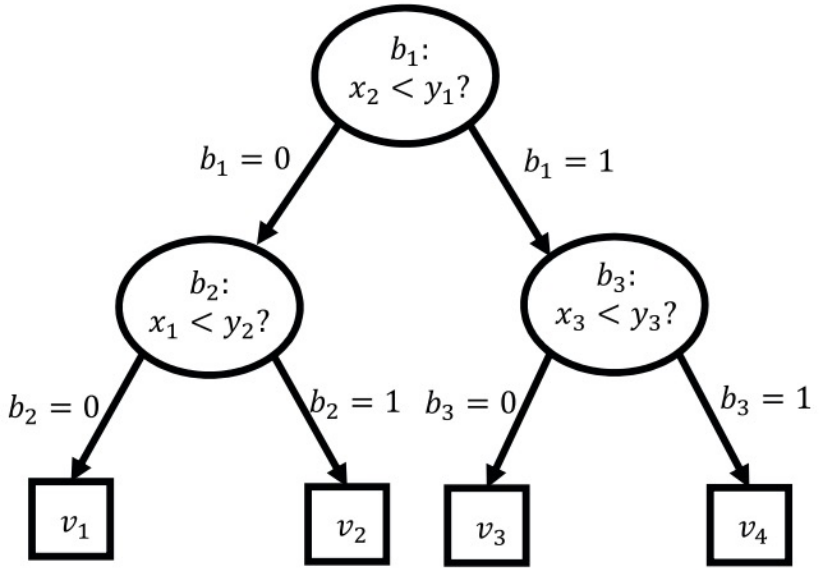
Secure Inference Result Generation

方案一 **Polynomial-Based Method** [1]

$$\text{sum of } \{z_k\}_{k=1}^{K}$$

例：找到合适的编码方式

$$z_1 = v_1 \cdot (1 - b_1) \cdot (1 - b_2)$$
$$z_2 = v_2 \cdot (1 - b_1) \cdot b_2$$
$$z_3 = v_3 \cdot b_1 \cdot (1 - b_3)$$
$$z_4 = v_4 \cdot b_1 \cdot b_3$$



假设最终预测结果是 $\mathcal{LN}_2$

$$b_1 = 0$$
$$b_2 = 1$$

$$z_1 = v_1 \cdot 1 \cdot 0 = 0$$
$$z_2 = v_2 \cdot 1 \cdot 1 = v_2$$
$$z_3 = v_3 \cdot 0 \cdot (1 - b_3) = 0$$
$$z_4 = v_4 \cdot 0 \cdot b_3 = 0$$

[1] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in Proc. Conf. Netw. Distrib. Syst. Secur. Symp., 2015, pp. 1–14.

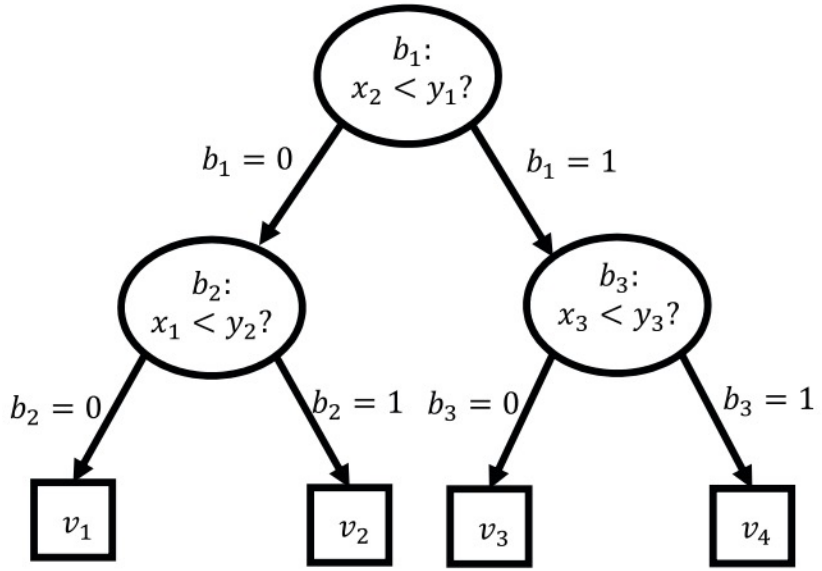
方案设计——安全预测生成

Secure Inference Result Generation

方案一 Polynomial-Based Method *[1]*

Input: Secret sharing $[b_j]$ for each decision node $\mathcal{DN}_j$.
Output: Inference result at the client.

1: For each decision node j,
   a) $\mathcal{C}_0$ sets $[ev_{j,0}]_0 = 1 - [b_j]_0$ and $\mathcal{C}_1$ sets $[ec_{j,0}]_1 = [b_j]_1$. This produces the secret-shared value of the left outgoing edge.
   b) $\mathcal{C}_i$ sets $[ev_{j,1}]_i = [b_j]_i$ as the secret-shared value of the right outgoing edge.
2: For each leaf node $\mathcal{LN}_k$, $\mathcal{C}_0$ and $\mathcal{C}_1$ compute a secret-shared polynomial term $[z_k]$ by multiplying the secret-shared values of all edges on its path and its prediction value.
3: The secret sharing $[v^*]$ of the inference result v^* is produced by summation of terms: $[v^*] = \sum_k [z_k]$.
4: Client receives $[v^*]$ and reconstructs v^* as the inference result for the feature vector $\mathbf{x}$.



开销：

$O(1)$ communication complexity

$O(d \cdot 2^d)$ secret-shared multiplications

[1] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in Proc. Conf. Netw. Distrib. Syst. Secur. Symp., 2015, pp. 1–14.

方案设计——安全预测生成

北京郵電大學 Beijing University of Posts and Telecommunications

Secure Inference Result Generation

方案二　**Path Cost-Based Method** *[2]*

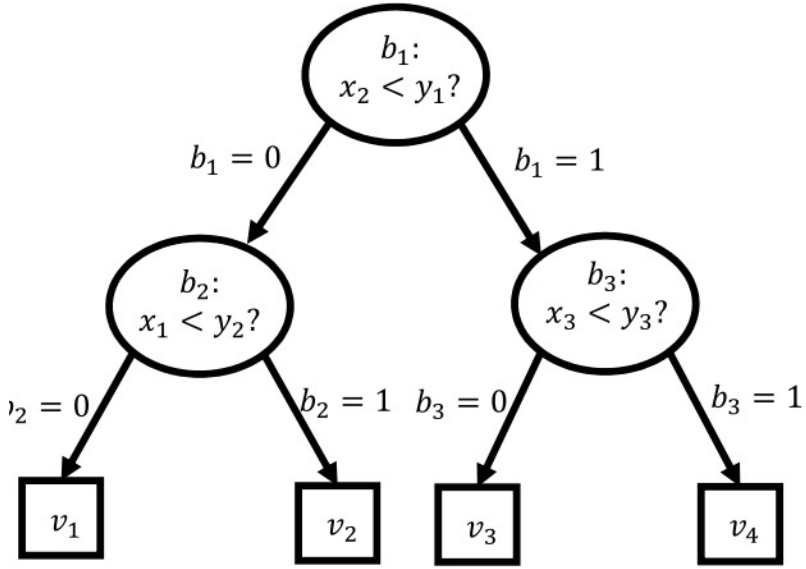
与方案一的设定相反

$$\text{left outgoing edge}: ec_{j,0} = b_j$$

$$\text{right outgoing edge}: ec_{j,1} = 1 - b_j$$

当前仅当path cost 为0时，找对对应的叶子结点

$$z_1 = b_1 + b_2$$
$$z_2 = b_1 + (1 - b_2)$$
$$z_3 = (1 - b_1) + b_3$$
$$z_4 = (1 - b_1) + (1 - b_3)$$



$$\mathrm{P}\,[1, 0, b_3, 1 + (1 - b_3)]$$

$$[v_1, v_2, v_3, v_4]$$

问题：[2]中的方案是基于同态加密，如果直接转化成MPC会使Client知道节点信息。

解决：random multiplicative masking

使非0 path cost 是一个随机值
Path cost为0 的叶子结点还是0

[2] R. K. H. Tai, J. P. K. Ma, Y. Zhao, and S. S. M. Chow, "Privacypreserving decision trees evaluation via linear functions," in Proc. Eur. Symp. Res. Comput. Secur., 2017, pp 494–512.

方案设计——安全预测生成

Secure Inference Result Generation

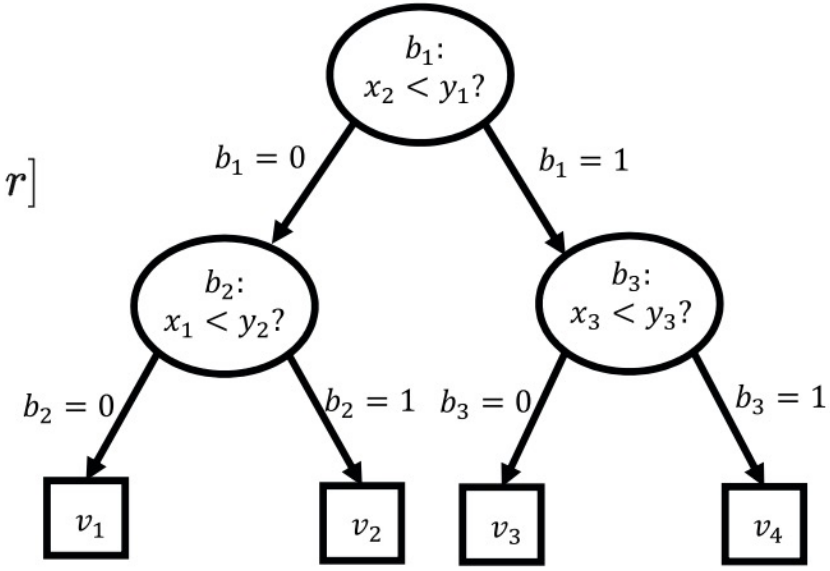
方案二　Path Cost-Based Method *[2]*

Input: Secret sharing $[b_j]^p$ of each decision node $\mathcal{DN}_j$.
Output: Inference result at the client.

1: For each decision node $\mathcal{DN}_j$,
   a) $\mathcal{C}_i$ sets $[ec_{j,0}]_i^p = [b_j]_i^p$ as the secret-shared left edge cost.
   b) $\mathcal{C}_0$ sets $[ec_{j,1}]_0^p = 1 - [b_j]_0^p$ and $\mathcal{C}_1$ sets $[ec_{j,1}]_1^p = [b_j]_1^p$, where $[ec_{j,1}]^p$ is the secret-shared right edge cost.
2: For each leaf node $\mathcal{LN}_k$, the secret sharing of path cost $[pc_k]^p$ is computed by summing up all edge costs along that path.
3: $\mathcal{C}_0$ computes $[pc_k^*]_0^p = [r_k \cdot pc_k]_0^p = r_k \cdot [pc_k]_0^p$ and $[v_k^*]_0^p = [r_k' \cdot pc_k + v_k]_0^p = r_k' \cdot [pc_k]_0^p + [v_k]_0^p$.
4: $\mathcal{C}_0$ applies a random permutation π to the above shares, and obtains $\{[pc_{\pi(k)}^*]_0^p\}_{k=1}^K$ and $\{[v_{\pi(k)}^*]_0^p\}_{k=1}^K$.

注：不是安全数组访问问题

$[1 \cdot r, 0, b_3 \cdot r, (1 - b_3) \cdot r]$
$[v_1', v_2, v_3', v_4']$



5: $\mathcal{C}_0$ shares with $\mathcal{C}_1$ the random masks $\{r_k\}_{k=1}^K$, $\{r_k'\}_{k=1}^K$, and the random permutation π.
6: $\mathcal{C}_1$ computes $[pc_k^*]_1^p = [r_k \cdot pc_k]_1^p = r_k \cdot [pc_k]_1^p$ and $[v_k^*]_1^p = [r_k' \cdot pc_k + v_k]_1^p = r_k' \cdot [pc_k]_1^p + [v_k]_1^p$.
7: $\mathcal{C}_1$ applies the same random permutation π and produces $\{[pc_{\pi(k)}^*]_1^p\}_{k=1}^K$ and $\{[v_{\pi(k)}^*]_1^p\}_{k=1}^K$.
8: Client receives the secret sharings $\{[pc_{\pi(k)}^*]^p\}_{k=1}^K$ and $\{[v_{\pi(k)}^*]^p\}_{k=1}^K$, and reconstructs $\{pc_{\pi(k)}^*\}_{k=1}^K$ and $\{v_{\pi(k)}^*\}_{k=1}^K$.
9: Client outputs v^* as the inference result when finding that its pc^* is 0.

PART 04

安全性证明

安全性证明

假设参与方是半诚实

通用可组合框架：若某些子协议是安全的，则组合这些子协议构成的新协议或框架也是安全的。

协议安全定义：现实执行中的敌手，对应理想执行中的模拟器，若已知任何输入，不存在任何一个环境可以让敌手在计算上区分这两者执行情况，则认为现实执行中的协议可以模拟理想执行中的既定函数功能，即认为该协议是计算安全的。

在半可信模型下，本文框架可以保证计算正确性和安全性；在恶意模型下（至多一个服务器被腐蚀），本文框架仅能保证计算安全性，并不能保证恶意敌手具备的行为能力不破坏计算正确性。

Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In FOCS, pages 136–145. IEEE Computer Society, 2001.

PART 05

实验部分

实验部分

实验环境： two AWS t3.xlarge instances (Intel Xeon Platinum 8175M CPU 2.50GHz and 16GB RAM)

平均延迟：75.422ms 带宽： 161 Mbits/s.

编程语言：C++
库函数： libOTe *[1][2]*（茫然传输）

树的深度：3-7
特征维度：9-57

对比文献： **ZDWWN** protocol *[3]*

[1] P. Rindal, "libOTe: an efficient, portable, and easy to use Oblivious Transfer Library," https://github.com/osu-crypto/libOTe.

[2] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu, "Efficient batched oblivious PRF with applications to private set intersection," in Proc. of ACM CCS, 2016.

[3] Y. Zheng, H. Duan, C. Wang, R. Wang, and S. Nepal, "Securely and efficiently outsourcing decision tree inference," IEEE Trans. Dependable Secure Comput., pp. 1–1, 2020.

实验部分

Local-side Performance Evaluation

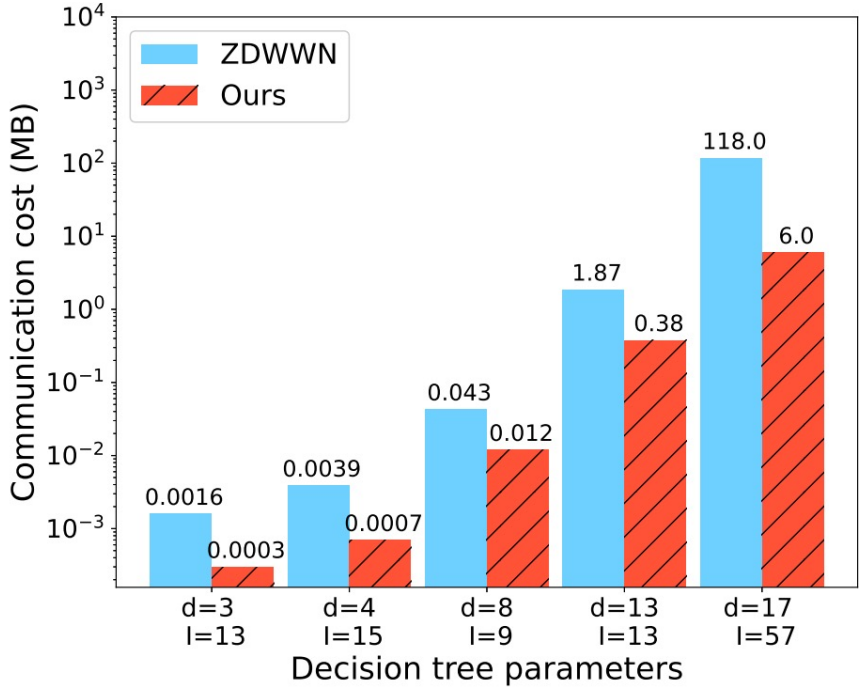


Fig. 8. Communication performance of the provider.

索引向量 O(J) 对比ZDWWN协议里的辅助矩阵 O(J·I) 。

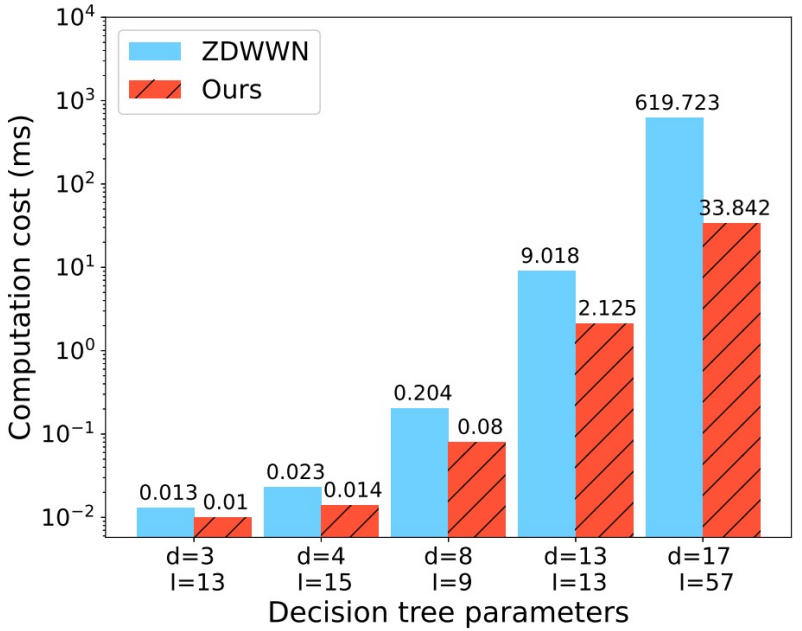


Fig. 9. Computation performance of the provider.

测试了不同深度的决策树：
通信开销：0.0003MB—6MB,ZWDDN协议则是0.0016MB—118MB
运行时间：0.01ms—33.842ms,ZWDDN协议则是0.013ms—619.723ms

综合来看
通信：改进后的协议比ZDWWN协议最高提升19 ×（平均7×）
计算：改进后的协议比ZDWWN协议最高提升18 ×（平均5×）

实验部分

Local-side Performance Evaluation

TABLE 2
Performace of the Client

d	I	Computation (ms)	Communication (KB)
3	13	0.0057	0.22
4	15	0.0060	0.25
8	9	0.0054	0.16
13	13	0.0056	0.22
17	57	0.0098	0.91

输入的维度越小，通信和计算开销就越小。

两个方案中，Client执行表现是一样的。

实验部分

Cloud-side Performance Evaluation

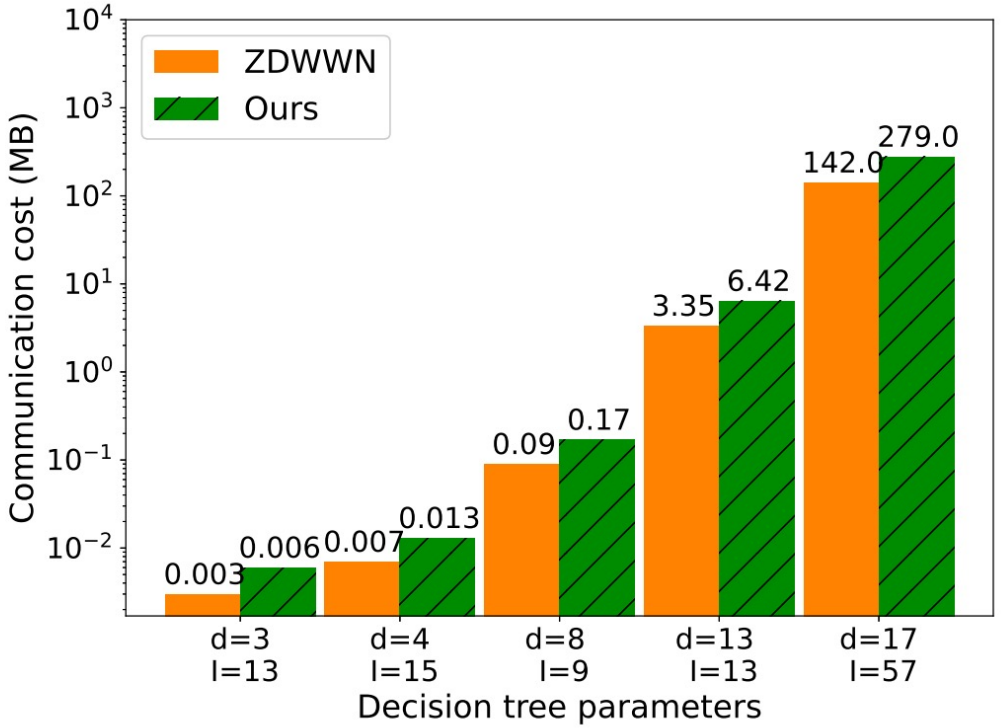


Fig. 10. Communication performance at the cloud.

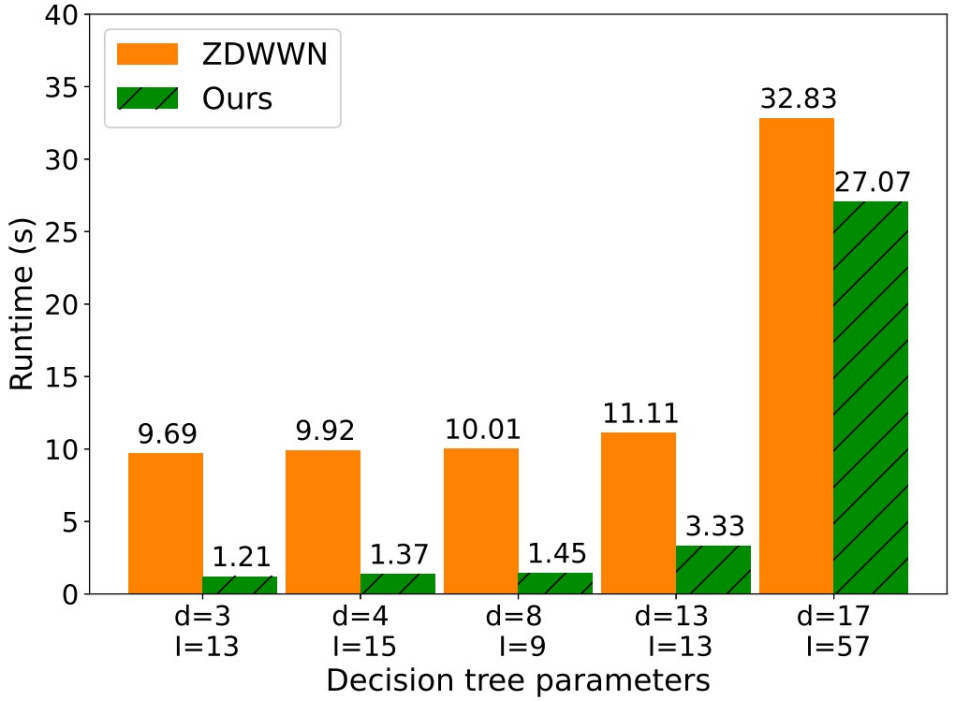


Fig. 11. Overall *end-to-end* online runtime performance at the cloud over realistic WAN.

通信开销增大，因为有特征选择阶段的OT

决策点评估算法的优化，计算复杂度从O(L)降低到O(log(L))

极大的减少了延迟

实验部分

Cloud-side Performance Evaluation

TABLE 3
Breakdown of Runtimes in Different Phases (in seconds) at the Cloud Servers, over WAN

Parameters		Secure Feature Selection		Secure Decision Node Evaluation		Secure Inference Generation	
d	I	ZDWWN	Ours	ZDWWN	Ours	ZDWWN	Ours
3	13	0.151	**0.527**	9.38	**0.529**	0.154	**0.154**
4	15	0.156	**0.529**	9.454	**0.53**	0.306	**0.306**
8	9	0.167	**0.533**	9.456	**0.532**	0.383	**0.383**
13	13	0.393	**0.91**	10.03	**1.735**	0.69	**0.69**
17	57	19.376	**21.006**	11.669	**4.281**	1.785	**1.785**

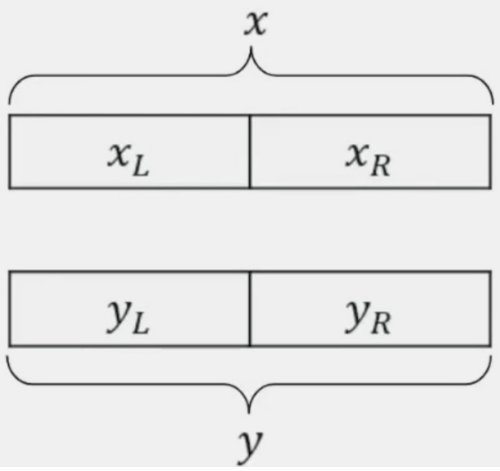
PART 06

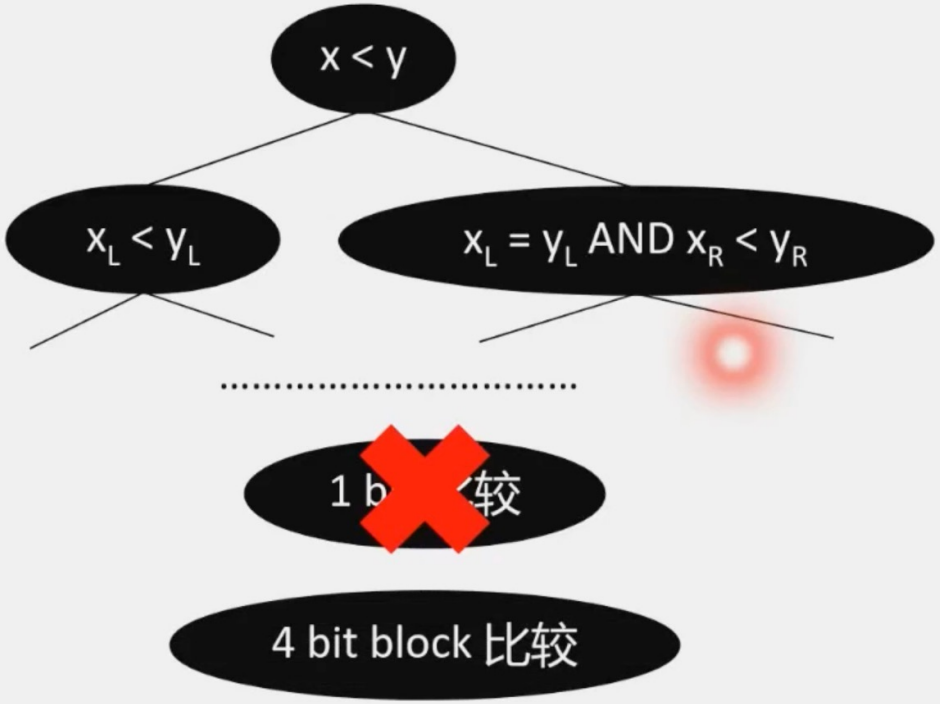
讨论部分

比较操作

Compare 怎么算

- CryptFlow2: 比较树
  - 分而治之
  - 各个分治部分并行运算





讨论部分

比较操作

Compare 怎么算

- CryptFlow2: 比较树
  - 分而治之
  - 各个分治部分并行运算
- 减少了通信轮数和AND门数目

4 bit block 比较

假设 x=a

x < 0
x < 1
...
x < a
} 0

x < a+1
...
x < 15
} 1

1-of-16 OT

Alice 输入: $r \oplus \{x < i\}$, 0 <= i <= 15
Bob 输入: y
➔
Alice 得到: r
Bob 得到: $r \oplus \{x < y\}$

讨论部分

比较操作

Compare 需要的原语

- 1-of-2^m OT
- AND Gate
  - 需要 beaver triple
  - 1-of-2 Random OT
- CryptFlow2 使用了传统的 IKNP-OT
- 近两年涌现了一批 Silent OT 方案
  - [CCS19], [Crypto21], **[Ferret]**
  - 可以在接近 0 通信开销的情况下生成大量 Random Correlated OT：

$$c_i = b_i + a_i \cdot x$$

其中，$(b_i, b_i + x)$ 是 Sender 的correlated消息，$a_i \in \{0,1\}$ 是 Receiver 的 choice bit

  - Random Correlated OT → 任意其他类型 OT

讨论部分

编码

1、算术编码和二进制编码不兼容，需要额外地转换；mod 2^k,mod 2,mod p

A B Y三个协议的转换

2、算术运算和转换电路

谢谢观看