SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Multi-Party Privacy-Preserving Decision Tree Training with a Privileged Party

Yiwen TONG[1], Qi FENG[1*], Min LUO[1, 2] & Debiao HE[1, 3*]

[1]*Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China;*
[2]*Shanghai Technology Innovation Centre of Distributed Privacy-Preserving Artificial Intelligence, Matrix Elements Technologies, Shanghai 200232, China;*
[3]*Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China*

Appendix A Proof of Theorem 1

Theorem 1 (Correctness of Π_{div}). For shares $\langle x \rangle$ and $\langle y \rangle$ held by online parties, where $\langle x \rangle, \langle y \rangle \in \mathbb{Z}_N$, Π_{div} can correctly outputs shares of division result $\left\langle \frac{x}{y} \right\rangle$ for all parties.

Proof. To prove the correctness of Π_{div}, it is necessary to demonstrate that the shares output by Π_{div} can be reconstructed using Π_{rec} to obtain the correct value of $\frac{x}{y}$. The reconstruction process will be explained for the following two scenarios:

If no assistant party drops out:

$$\begin{aligned} z &= \alpha_0 \cdot \langle z \rangle_0 + \alpha_1 \cdot \langle z \rangle_1 + \alpha_2 \cdot \langle z \rangle_2 \\ &= \frac{\alpha_0 \cdot \langle r \cdot x \rangle_0 + \alpha_1 \cdot \langle r \cdot x \rangle_1 + \alpha_2 \cdot \langle r \cdot x \rangle_2}{r \cdot y} \\ &= \frac{r \cdot x}{r \cdot y} = \frac{x}{y} \end{aligned} \tag{A1}$$

If one of the assistant parties (P_2, for example) drops out:

$$\begin{aligned} z &= \alpha'_0 \cdot \langle z \rangle_0 + \alpha'_1 \cdot \langle z \rangle_1 + \alpha'_3 \cdot \langle z \rangle_3 \\ &= \frac{\alpha'_0 \cdot \langle r \cdot x \rangle_0 + \alpha'_1 \cdot \langle r \cdot x \rangle_1 + \alpha'_3 \cdot \langle r \cdot x \rangle_3}{r \cdot y} \\ &= \frac{r \cdot x}{r \cdot y} = \frac{x}{y} \end{aligned} \tag{A2}$$

Appendix B Proof of Theorem 2

Theorem 2 (Security of Π_{div}). The divison protocol Π_{div} securely realizes the funtionality $\mathcal{F}_{div}$ under the passive adversary.

Proof. The ideal funtionality $\mathcal{F}_{div}$ for the division protocol Π_{div} is depicted in Table B1.

Table B1 Ideal Funtionality $\mathcal{F}_{div}$

Funtionality $\mathcal{F}_{div}$	
Input:	**Output:**
- P_0 inputs $\langle x \rangle_0$, $\langle x \rangle_3$ and $\langle y \rangle_0$, $\langle y \rangle_3$;	- P_0 outputs $\langle z \rangle_0$ and $\langle z \rangle_3$;
- P_1 inputs $\langle x \rangle_1$ and $\langle y \rangle_1$;	- P_1 outputs $\langle z \rangle_1$;
- P_2 inputs $\langle x \rangle_2$ and $\langle y \rangle_2$.	- P_2 outputs $\langle z \rangle_2$, where $z = \frac{x}{y}$.

For the case of corrupting P_0, the simulator $\mathcal{S}_{div}^{P_0}$ works as follows:

(1) receives $\langle x \rangle_0$, $\langle x \rangle_3$, $\langle y \rangle_0$ and $\langle y \rangle_3$ from P_0.
(2) receives $\langle r \rangle_0$ and $\langle r \rangle_3$ from P_0.
(3) obtains $\langle r \cdot y \rangle_0$ and $\langle r \cdot y \rangle_3$ by Π_{mul}, which has been formally proven to be secure in [1].
(4) selects random values $\langle r \cdot y \rangle_1$ and $\langle r \cdot y \rangle_2$.
(5) computes $r \cdot y = \alpha_0 \cdot \langle r \cdot y \rangle_0 + \alpha_1 \cdot \langle r \cdot y \rangle_1 + \alpha_2 \cdot \langle r \cdot y \rangle_2$.
(6) obtains $\langle r \cdot x \rangle_0$ and $\langle r \cdot x \rangle_3$ by Π_{mul}.
(7) computes $\langle z \rangle_0 = \frac{\langle r \cdot x \rangle_0}{r \cdot y}$ and $\langle z \rangle_3 = \frac{\langle r \cdot x \rangle_3}{r \cdot y}$.
(8) outputs $(\langle x \rangle_0, \langle x \rangle_3, \langle r \cdot y \rangle_{i'}, \langle z \rangle_0, \langle z \rangle_3, i' \in \{1, 2\})$.

For the case of corrupting P_1 and P_2, the simulator $\mathcal{S}_{div}^{P_1, P_2}$ works as follows:

(1) receives $\langle x \rangle_1$, $\langle x \rangle_2$, $\langle y \rangle_1$ and $\langle y \rangle_2$ from P_1 and P_2.

* Corresponding author (email: fengqi.whu@whu.edu.cn, hedebiao@163.com)

(2) receives $\langle r\rangle_1$ and $\langle r\rangle_2$ from P_1 and P_2.
(3) obtains $\langle r\cdot y\rangle_1$ and $\langle r\cdot y\rangle_2$ by Π_{mul}.
(4) selects random values $\langle r\cdot y\rangle_0$.
(5) computes $r\cdot y=\alpha_0\cdot\langle r\cdot y\rangle_0+\alpha_1\cdot\langle r\cdot y\rangle_1+\alpha_2\cdot\langle r\cdot y\rangle_2$.
(6) obtains $\langle r\cdot x\rangle_1$ and $\langle r\cdot x\rangle_2$ by Π_{mul}.
(7) computes $\langle z\rangle_0=\frac{\langle r\cdot x\rangle_0}{r\cdot y}$ and $\langle z\rangle_1=\frac{\langle r\cdot x\rangle_2}{r\cdot y}$.
(8) outputs $(\langle x\rangle_{i'},\langle r\cdot y\rangle_0,\langle z\rangle_{i'},i'\in\{1,2\})$.

We represent the view of P_0 and $P1$, $P2$ as $\boldsymbol{view}^{div}_{P_0}$ and $\boldsymbol{view}^{div}_{P_1,P_2}$ respectively. It is clear that the distribution of P_0's simulated view in the ideal world is identical to the distribution of its view in the actual execution, that is,

$$\boldsymbol{view}^{div}_{P_0}(\langle x\rangle_j,\langle y\rangle_j,\langle z\rangle_j,j\in\{0,1,2,3\})\cong\mathcal{S}^{P_0}_{div}(\langle x\rangle_0,\langle y\rangle_0,\langle x\rangle_3,\langle y\rangle_3,\langle z\rangle_0,\langle z\rangle_3) \tag{B1}$$

Besides, the view of P_1 and P_2 in the real world is the same as $\mathcal{S}^{P_0,P_1}_{div}$'s output. Mathematically,

$$\boldsymbol{view}^{div}_{P_1,P_2}(\langle x\rangle_j,\langle y\rangle_j,\langle z\rangle_j,j\in\{0,1,2,3\})\cong\mathcal{S}^{P_1,P_2}_{div}(\langle x\rangle_1,\langle y\rangle_1,\langle x\rangle_2,\langle y\rangle_2,\langle z\rangle_1,\langle z\rangle_2) \tag{B2}$$

Appendix C Proof of Theorem 4

Theorem 4 (Security of Π_{comp}). The comparison protocol Π_{comp} securely realizes the functionality $\mathcal{F}_{comp}$ under the passive adversary.

Proof. The ideal funtionality $\mathcal{F}_{comp}$ for the comparison protocol Π_{comp} is depicted in Table C1.

Table C1 Ideal Funtionality $\mathcal{F}_{comp}$

Funtionality $\mathcal{F}_{comp}$	
Input:	**Output:**
- P_0 inputs $\langle x\rangle_0$, $\langle x\rangle_3$, $\langle y\rangle_0$ and $\langle y\rangle_3$;	- P_0 outputs $\langle z\rangle_0$ and $\langle z\rangle_3$;
- P_1 inputs $\langle x\rangle_1$ and $\langle y\rangle_1$;	- P_1 outputs $\langle z\rangle_1$;
- P_2 inputs $\langle x\rangle_2$ and $\langle y\rangle_2$.	- P_2 outputs $\langle z\rangle_2$, where $z=\mathbf{1}\{x>y\}$.

For the case of corrupting P_0, the simulator $\mathcal{S}^{P_0}_{comp}$ works as follows:
(1) receives $\langle x\rangle_0$, $\langle x\rangle_3$, $\langle y\rangle_0$ and $\langle y\rangle_3$ from P_0.
(2) receives k_1 and k_2 from P_0.
(3) receives r from P_0.
(4) computes $\langle x-y\rangle_i=\langle x\rangle_i-\langle y\rangle_i$, where $i\in\{0,3\}$.
(5) obtains $\langle r\cdot(x-y)\rangle_0$ and $\langle r\cdot(x-y)\rangle_3$ by Π_{extmul}, which has been formally proven to be secure in Appendix E.
(6) obtains $\langle f\rangle_0$ and $\langle f\rangle_3$ by Π_{shr}, which has been formally proven to be secure in [1].
(7) select random values $\langle f\rangle_1$ and $\langle f\rangle_2$.
(8) computes $\langle z\rangle_0=\langle f\rangle_0\oplus\mathsf{MSB}(r)$ and $\langle z\rangle_3=\langle f\rangle_3$.
(9) outputs $(\langle x\rangle_0,\langle x\rangle_3,\langle f\rangle_{i'},\langle z\rangle_0,\langle z\rangle_3,i'\in\{0,3\})$.

For the case of corrupting P_1 and P_2, the simulator $\mathcal{S}^{P_1,P_2}_{comp}$ works as follows:
(1) receives $\langle x\rangle_1$, $\langle x\rangle_2$, $\langle y\rangle_1$ and $\langle y\rangle_2$ from P_1 and P_2.
(2) receives k_1 and k_2 from P_1 and P_2.
(3) computes $\langle x-y\rangle_i=\langle x\rangle_i-\langle y\rangle_i$, where $i\in\{1,2\}$.
(4) obtains $\langle r\cdot(x-y)\rangle_1$ and $\langle r\cdot(x-y)\rangle_2$ by Π_{extmul}, which has been formally proven to be secure in Appendix E.
(5) obtains $r\cdot(x-y)$ by computing $r\cdot(x-y)=\alpha_0\cdot\langle r\cdot(x-y)\rangle_0+\alpha_1\cdot\langle r\cdot(x-y)\rangle_1+\alpha_2\cdot\langle r\cdot(x-y)\rangle_2$.
(6) obtains $[f]_1$ and $[f]_2$ by $\mathsf{Eval}^{<}_{\frac{N}{2},1}$, and FSS has a well-documented proof of security.
(7) computes $f=[f]_1+[f]_2$, and obtains $\langle f\rangle_1$ and $\langle f\rangle_2$ by Π_{shr}, which has been formally proven to be secure in [1].
(8) computes $\langle z\rangle_1=\langle f\rangle_1$ and $\langle z\rangle_2=\langle f\rangle_2$.
(9) outputs $(\langle x\rangle_1,\langle x\rangle_2,\langle f\rangle_{i'},\langle z\rangle_1,\langle z\rangle_2,i'\in\{1,2\})$.

We represent the view of P_0 and $P1$, $P2$ as $\boldsymbol{view}^{comp}_{P_0}$ and $\boldsymbol{view}^{comp}_{P_1,P_2}$ respectively. It is clear that the distribution of P_0's simulated view in the ideal world is identical to the distribution of its view in the actual execution, that is,

$$\boldsymbol{view}^{comp}_{P_0}(\langle x\rangle_j,\langle y\rangle_j,\langle z\rangle_j,j\in\{0,1,2,3\})\cong\mathcal{S}^{P_0}_{comp}(\langle x\rangle_0,\langle y\rangle_0,\langle x\rangle_3,\langle y\rangle_3,\langle z\rangle_0,\langle z\rangle_3) \tag{C1}$$

Besides, the view of P_1 and P_2 in the real world is the same as $\mathcal{S}^{P_0,P_1}_{comp}$'s output. Mathematically,

$$\boldsymbol{view}^{comp}_{P_1,P_2}(\langle x\rangle_j,\langle y\rangle_j,\langle z\rangle_j,j\in\{0,1,2,3\})\cong\mathcal{S}^{P_1,P_2}_{comp}(\langle x\rangle_1,\langle y\rangle_1,\langle x\rangle_2,\langle y\rangle_2,\langle z\rangle_1,\langle z\rangle_2) \tag{C2}$$

Appendix D Proof of Theorem 5

Theorem 5 (Correctness of Π_{extmul}). For the plaintext x held by P_0 and the shares $\langle y\rangle$ held by online parties, where $x,\langle y\rangle\in\mathbb{Z}_N$, Π_{extmul} can correctly outputs shares of multiplication result $\langle x\cdot y\rangle$ for all parties.

Proof. To prove the correctness of Π_{extmul}, it is necessary to demonstrate that the shares output by Π_{extmul} can be reconstructed using Π_{rec} to obtain the correct value of $x\cdot y$. The reconstruction process will be explained for the following two scenarios:

If no assistant party drops out:

$$\begin{aligned}
z&=\alpha_0\cdot\langle z\rangle_0+\alpha_1\cdot\langle z\rangle_1+\alpha_2\cdot\langle z\rangle_2\\
&=\alpha_0\cdot\frac{x\cdot f}{\alpha_0}+(\alpha_0\cdot\langle h\rangle_0+\alpha_1\cdot\langle h\rangle_1+\alpha_2\cdot\langle h\rangle_2)-e\cdot(\alpha_0\cdot\langle v\rangle_0+\alpha_1\cdot\langle v\rangle_1+\alpha_2\cdot\langle v\rangle_2)\\
&=x\cdot(y+v)+u\cdot v-v\cdot(x+u)\\
&=x\cdot y
\end{aligned} \tag{D1}$$

If one of the assistant parties (P_2, for example) drops out:

$$\begin{aligned} z &= \alpha_0' \cdot \langle z \rangle_0 + \alpha_1' \cdot \langle z \rangle_1 + \alpha_3' \cdot \langle z \rangle_3 \\ &= \alpha_0' \cdot \frac{x \cdot f}{\alpha_0} + (\alpha_0' \cdot \langle h \rangle_0 + \alpha_1' \cdot \langle h \rangle_1 + \alpha_3' \cdot \langle h \rangle_3) - e \cdot (\alpha_0' \cdot \langle v \rangle_0 + \alpha_1' \cdot \langle v \rangle_1 + \alpha_3' \cdot \langle v \rangle_3) \\ &= x \cdot (y + v) + u \cdot v - v \cdot (x + u) \\ &= x \cdot y \end{aligned} \tag{D2}$$

Since there exist public constants that satisfy the requirement and are equal, i.e., $\alpha_0 = \alpha_0' = \alpha_0'' = 1$, we can compute $\alpha_0' \cdot \frac{x \cdot f}{\alpha_0} = x \cdot f$.

Appendix E Proof of Theorem 6

Theorem 6 (Security of Π_{extmul}). The extension protocol of secure multiplication Π_{extmul} securely realizes the funtionality $\mathcal{F}_{extmul}$ under the passive adversary.

Proof. The ideal funtionality $\mathcal{F}_{extmul}$ for the extention protocol of the multiplication protocol Π_{extmul} is depicted in Table E1.

Table E1 Ideal Funtionality $\mathcal{F}_{extmul}$

Funtionality $\mathcal{F}_{extmul}$	
Input:	**Output:**
- P_0 inputs x, $\langle y \rangle_0$ and $\langle y \rangle_3$;	- P_0 outputs $\langle z \rangle_0$ and $\langle z \rangle_3$;
- P_1 inputs $\langle y \rangle_1$;	- P_1 outputs $\langle z \rangle_1$;
- P_2 inputs $\langle y \rangle_2$.	- P_2 outputs $\langle z \rangle_2$, where $z = x \cdot y$.

For the case of corrupting P_0, the simulator $\mathcal{S}_{extmul}^{P_0}$ works as follows:
(1) receives x, $\langle y \rangle_0$ and $\langle y \rangle_3$ from P_0.
(2) receives $\langle u \rangle_0$, $\langle v \rangle_0$, $\langle h \rangle_0$, $\langle u \rangle_3$, $\langle v \rangle_3$ and $\langle h \rangle_3$ from P_0.
(3) selects random values $\langle u \rangle_1$, $\langle u \rangle_2$, $\langle d \rangle_1$ and $\langle d \rangle_2$.
(4) computes $\langle d \rangle_i = \langle y \rangle_i + \langle v \rangle_i$, where $i \in \{0, 3\}$.
(5) computes $u = \alpha_0 \cdot \langle u \rangle_0 + \alpha_1 \cdot \langle u \rangle_1 + \alpha_2 \cdot \langle u \rangle_2$ and $d = \alpha_0 \cdot \langle d \rangle_0 + \alpha_1 \cdot \langle d \rangle_1 + \alpha_2 \cdot \langle d \rangle_2$.
(6) computes $e = u + x$.
(7) computes $\langle z \rangle_0 = \frac{x \cdot d}{\alpha_0} + \langle h \rangle_0 - \langle v \rangle_0 \cdot e$.
(8) outputs $(x, \langle y \rangle_0, \langle y \rangle_3, \langle u \rangle_{i'}, \langle d \rangle_{i'}, \langle z \rangle_0, \langle z \rangle_3$, where $i' \in \{1, 2\})$.

For the case of corrupting P_1 and P_2, the simulator $\mathcal{S}_{extmul}^{P_1,P_2}$ works as follows:
(1) receives $\langle y \rangle_1$ and $\langle y \rangle_2$ from P_1 and P_2.
(2) receives $\langle u \rangle_1$, $\langle v \rangle_1$, $\langle h \rangle_1$, $\langle u \rangle_2$, $\langle v \rangle_2$ and $\langle h \rangle_2$ from P_1 and P_2.
(3) selects random values $\langle d \rangle_0$.
(4) computes $\langle d \rangle_i = \langle y \rangle_i + \langle v \rangle_i$, where $i \in \{1, 2\}$.
(5) computes $d = \alpha_0 \cdot \langle d \rangle_0 + \alpha_1 \cdot \langle d \rangle_1 + \alpha_2 \cdot \langle d \rangle_2$.
(6) computes $\langle z \rangle_1 = \langle h \rangle_1 - e \cdot \langle v \rangle_1$ and $\langle z \rangle_2 = \langle h \rangle_2 - e \cdot \langle v \rangle_2$.
(7) outputs $(\langle y \rangle_j, \langle d \rangle_0, \langle z \rangle_{i'}, i' \in \{1, 2\})$.

We present the view of P_0 and P_1,P_2 as $\boldsymbol{view}_{P_0}^{extmul}$ and $\boldsymbol{view}_{P_1,P_2}^{extmul}$ respectively. It is clear that the distribution of P_0's simulated view in the ideal world is identical to the distribution of its view in the real execution, i.e.,

$$\boldsymbol{view}_{P_0}^{extmul}(x, \langle y \rangle_j, \langle z \rangle_j, j \in \{0, 1, 2, 3\}) \cong \mathcal{S}_{extmul}^{P_0}(x, \langle y \rangle_0, \langle y \rangle_3, \langle z \rangle_0, \langle z \rangle_3) \tag{E1}$$

In addition, the view of $P1$, P_2 in the real world is the same as $\mathcal{S}_{extmul}^{P_1,P_2}$'s output. Mathematically,

$$\boldsymbol{view}_{P_1,P_2}^{extmul}(\langle y \rangle_j, \langle z \rangle_j, j \in \{0, 1, 2, 3\}) \cong \mathcal{S}_{extmul}^{P_1,P_2}(\langle y \rangle_1, \langle z \rangle_1, \langle y \rangle_2, \langle z \rangle_2) \tag{E2}$$

References

1 L. Song, J. Wang, Z. Wang, X. Tu, G. Lin, W. Ruan, H. Wu, and W. Han, "pmpl: A robust multi-party learning framework with a privileged party," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2689–2703, 2022.