SEOULTECH

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

DeepChain: Auditable and Privacy-Preserving Deep Learning with Blockchain-based Incentive

2019-03-12

Shailendra Rathore
(rathoreshailendra@seoultech.ac.kr)

Abstract

- Training data nor model is expected to be exposed.
- Federated learning.
- Propose to present DeepChain which gives mistrustful parties incentives to participate in privacy-preserving learning, share gradients and update parameters correctly, and eventually accomplish iterative learning with a win-win result.
- Give an implementation prototype by integrating deep learning module with a Blockchain development platform (Corda V3.0).
- Evaluate it in terms of encryption performance and training accuracy, which demonstrates the feasibility of DeepChain.

1. Introduction and motivation

- Privacy issue worsens in the context of distributed deep learning, as compared to conventional standalone deep learning scenario.
- Federated learning also known as collaborative learning, distributed learning, is essentially the combination of deep learning and distributed coputation.
- Intermediate gradients, Parameter Server, Parties.
- This federated learning framework, however, cannot protect the privacy of the training data.
- Two serious problems
- (a) Privacy threats from curious parameter server: may drop gradients of some parties deliberately, or wrongly update model parameters on purpose.
- (b) lack of training data will result in poor deep learning models: very concerned about possible disclosure of their valuable data. For example, in healthcare
- Propose a secure and decentralized framework based on Blockchain-based incentive mechanism
- Cryptographic primitives for privacy preserving distributed deep learning

2. Background

- 2.1 Blockchain technology
- 2.2 Deep learning and distributed deep learning
- Three layers, namely input layer, hidden layer and output layer.
- Multiple hidden layers
- $w_{i,j}$ is a weight assigned to the connection between neuron i at layer $l-1$ and neuron j at layer l.
- Each neuron i also has a bias. These weights and bias are called model parameters
- Each layer are calculated based on parameters at previous layer and current layer, respectively.
- A key component in deep neural network training is called activation

$$Act_i(l) = Act_i(\Sigma_{j=1}^{n}(w_{j,i} * Act_j(l-1)) + b_i).$$

3. Threats and security goals

- Threat 1: Disclosure of local data and model.
- Threat 2: Participants with inappropriate behaviors.
- Security Goal 1: Auditability of gradient collecting and parameter update.
- Security Goal 2: Fairness guarantee for participants.

4. The Deepchain model

4.1 System overview

- ✓ Party:
- ✓ Trading:
- ✓ Cooperative group:
- ✓ Local model training:
- ✓ Collaborative model training:
- ✓ Worker:
- ✓ Iteration:
- ✓ Round:
- ✓ DeepCoin:

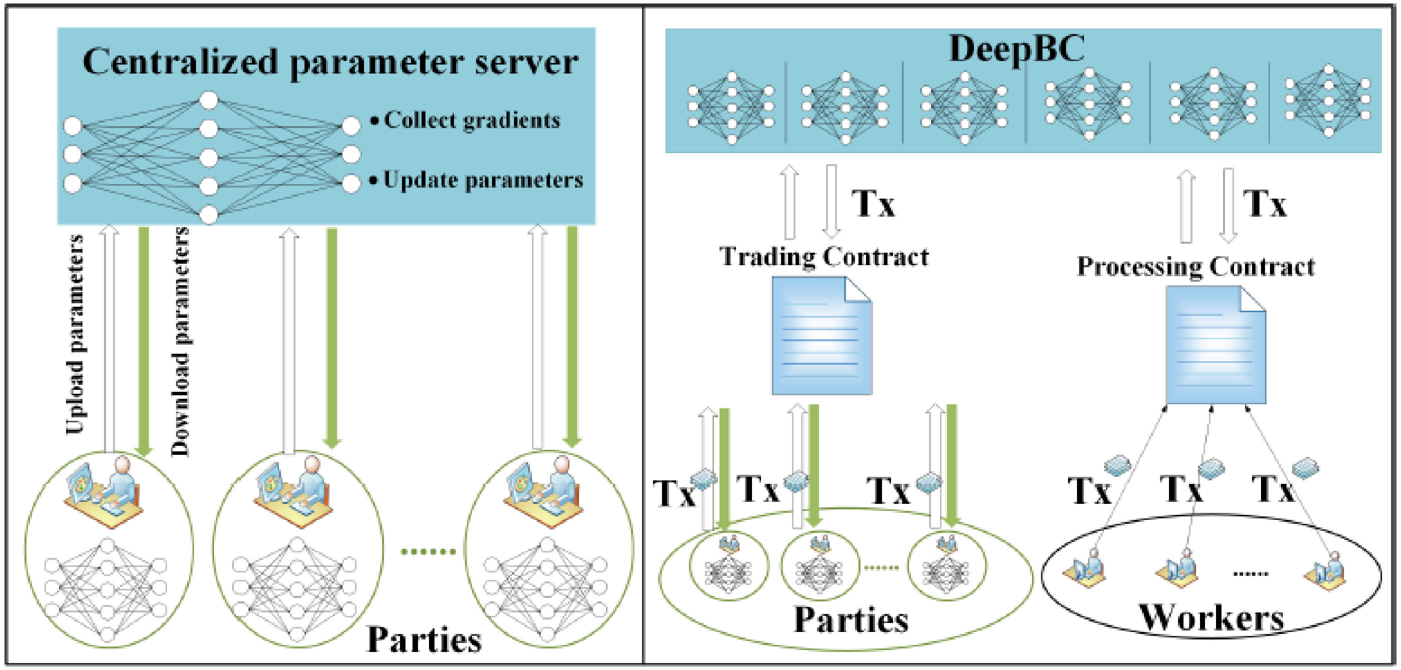


Fig. 1. The left corresponds to traditional distributed deep training framework, while the right is our DeepChain. Here, Trading Contract and Processing Contract are smart contract in DeepChain, together guiding the secure training process, while Tx refers to transaction.

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

4. The Deepchain model

4.2 Components of DeepChain

4.2.1 DeepChain bootstrapping

- Deep-Coin distribution and genesis block generation.

4.2.2 Incentive mechanism

Incentive mechanism ensures that (1) parties are honest in local model training and gradient trading, and (2) workers are honest in processing parties'

Liveness: both the party and the worker have the same common interest to obtain a trained collaborative model.transactions.
Compatibility: the more a party contributes, the more she will gain.

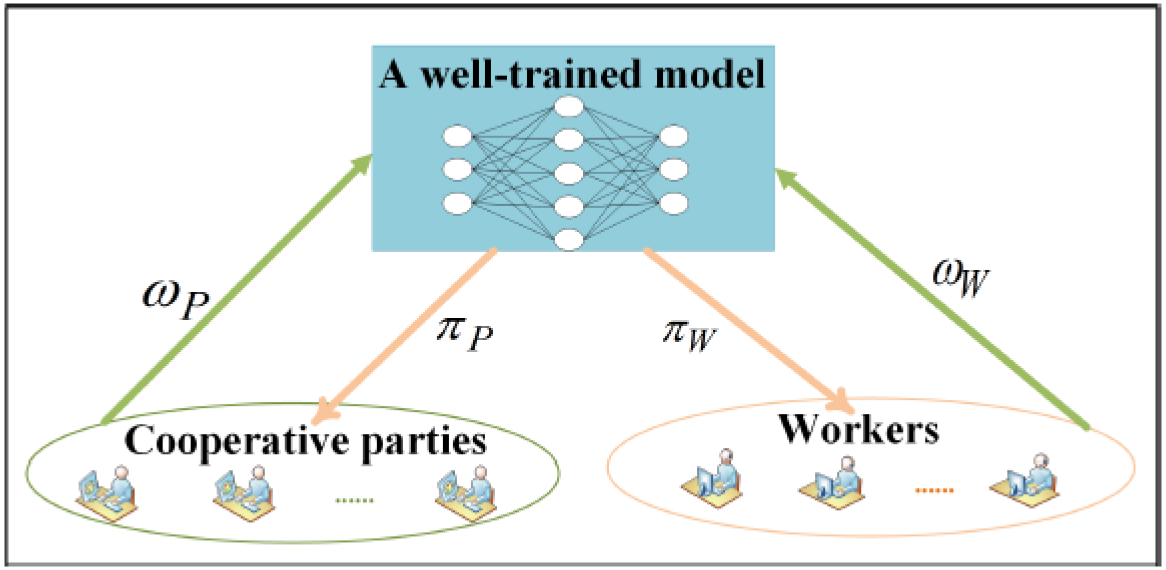


Fig.| 2. The incentive mechanism of DeepChain, where ω_P and ω_W represent the contributions of a party and a worker for maintaining v_c, respectively, and π_P and π_W represent their payoffs, respectively.

Payoff=

$$\begin{cases} Max(\pi_P) + Max(\pi_W) & \text{If} \quad Max(\omega_P) \bigwedge Max(\omega_W) \\ (\pi_P = 0) \bigwedge (\pi_W = 0) & \text{If} \quad (\omega_P = 0) \bigvee (\omega_W = 0) \end{cases}$$

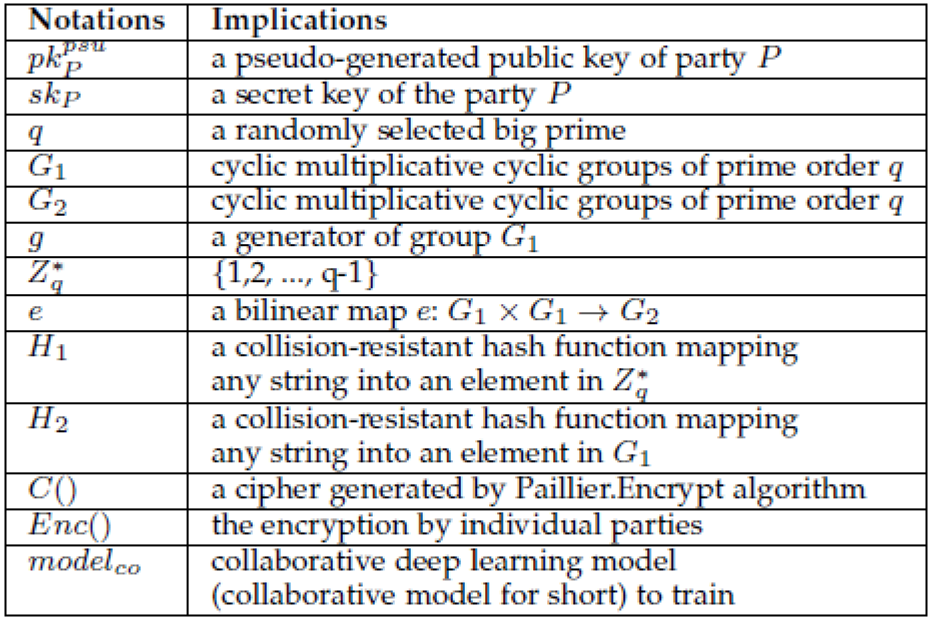
4. The Deepchain model

4.2.3 Asset statement

$$
Tran_{P_1} = pk_{P_1}^{psu} \rightarrow \Big\{ \Big(pk_{data\_P_1} = g^{H_1(data\_P_1)}, \\
\sigma_{j\_P_1} = (H_2(j) \cdot g^{H_1(data_{j}\_P_1)})^{H_1(data\_P_1)} \Big), \\
\text{"Keywords"} \Big\}.
$$

Some description of the asset, e.g., what kind of deep learning tasks the asset can be used for.

TABLE 1
Summary of notations



Notations	Implications
pk_P^{psu}	a pseudo-generated public key of party P
sk_P	a secret key of the party P
q	a randomly selected big prime
G_1	cyclic multiplicative cyclic groups of prime order q
G_2	cyclic multiplicative cyclic groups of prime order q
g	a generator of group G_1
Z_q^*	{1,2, ..., q-1}
e	a bilinear map e: $G_1 \times G_1 \rightarrow G_2$
H_1	a collision-resistant hash function mapping any string into an element in Z_q^*
H_2	a collision-resistant hash function mapping any string into an element in G_1
$C()$	a cipher generated by Paillier.Encrypt algorithm
$Enc()$	the encryption by individual parties
$model_{co}$	collaborative deep learning model (collaborative model for short) to train

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

4. The Deepchain model

4.2.4 Collaborative training

- Collaborative group establishment.
- Collaborative information commitment.
- Parties agree on the following information.

(1) Number of cooperative parties, N.

(2) Index of the current round, r.

(3) Parameters of Threshold Paillier algorithm.

Then, have the following equation

$$PK_{model} = (n_{model}, g_{model}, \theta = as, V = (v, \{v_i\}_{i \in [1,...,N]}))$$

$$(g^1_{model}, ..., g^l_{model}) = (g^{\alpha_1}_{model}, ..., g^{\alpha_l}_{model}).$$

(4) A collaborative model $model_{co}$ to be trained.

(5) $$commit_{SK_{model}} = (Enc(s_1||r||Sign(s_1||r)), ..., Enc(s_N||r||Sign(s_N||r)))$$

(6) The initial weights $W_{0,j}$ of local model of party j.

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

4. The Deepchain model

(7) A amount of deposits d($Coin).

$$Tran_{co} = pk_{co}^{psu} \rightarrow \Big\{N, r, PK_{model}, d, \vec{\alpha}, model_{co},$$
$$commit_{SK_{model}}, C(\mathbf{W}_{0,j}), d(\$Coin)\Big\}.$$

4. The Deepchain model

4.2.4 Collaborative training

- Gradient collecting via Trading Contract.

```
Algorithm 1: Trading($Tran^i_{P_1}$,...,$Tran^i_{P_N}$)
1  receiveGradientTX()
2  checkTimeout($T_{t1}$)
3  updateTime()      // $T'_{t_1} = T_{t_1} + |T_{i+1} - T_i|$
4  verifyGradientTX()
5  checkTimeout($T_{t2}$)
6  updateTime()      // $T'_{t2} = T_{t2} + |T_{i+1} - T_i|$
7  uploadGradientTX()#attaching to the address $pk^{psu}_{co}$
8  checkTimeout($T_{t3}$)
9  updateTime()      // $T'_{t3} = T_{t3} + |T_{i+1} - T_i|$
10 downloadUpdatedParam()#from the address $pk^{psu}_{co}$
11 checkTimeout($T_{t4}$)
12 updateTime()      // $T'_{t4} = T_{t4} + |T_{i+1} - T_i|$
13 decryptUpdatedParam()
14 checkTimeout($T_{t5}$)
15 updateTime()      // $T'_{t5} = T_{t5} + |T_{i+1} - T_i|$
16 return()
17 checkTimeout($T_{t6}$)
18 updateTime()      // $T'_{t6} = T_{t6} + |T_{i+1} - T_i|$
```

Parameter updating via Processing Contract.

```
Algorithm 2: Processing()
1 updateTX()
2 checkTimeout($T_{t7}$)
3 updateTime()      // $T'_{t7} = T_{t7} + T_r$
4 verifyTX()
5 checkTimeout($T_{t8}$)
6 updateTime()      // $T'_{t8} = T_{t8} + T_r$
7 appendTX()
8 checkTimeout($T_{t9}$)
9 updateTime()      // $T'_{t9} = T_{t9} + T_r$
```

$$C(\mathbf{W}_i) = C(\mathbf{W}_{i-1}) \cdot \frac{1}{N} \cdot (C(-\triangle\mathbf{W}_{i,1}) \cdot C(-\triangle\mathbf{W}_{i,2}) \cdot \\ \dots \cdot C(-\triangle\mathbf{W}_{i,N})).$$

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

4. The Deepchain model

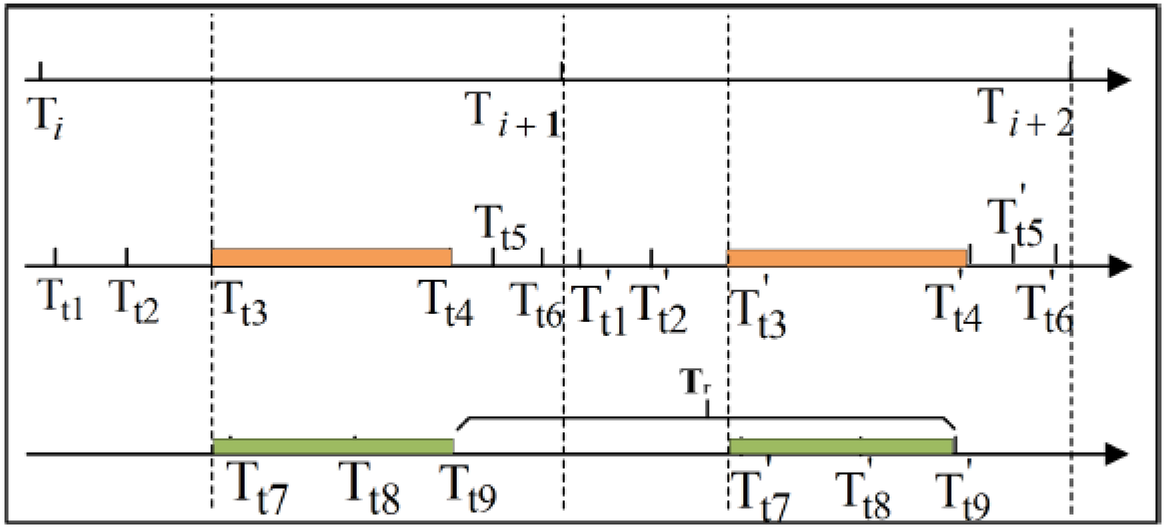
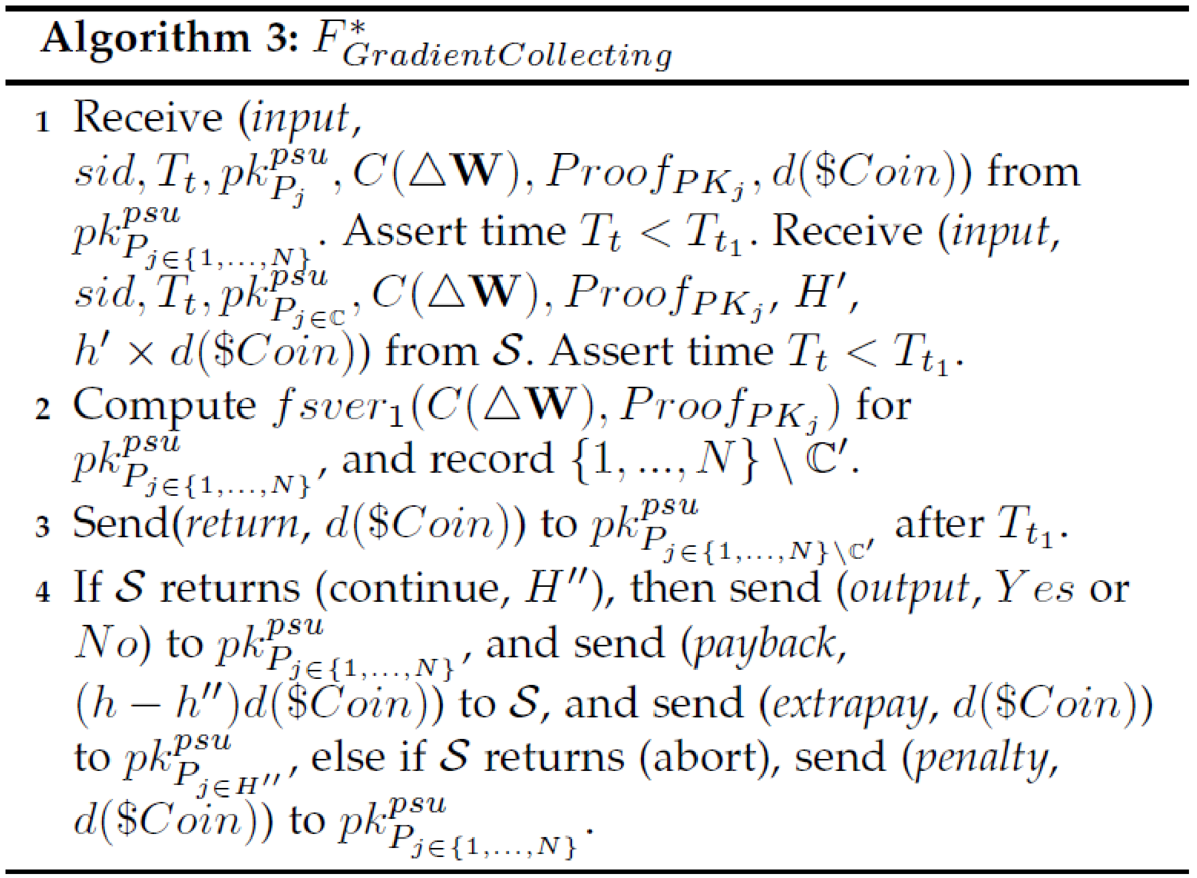


Fig. 3. Configuration of time points. From top to bottom: (1) the timeline of collaborative training, (2) the timeline of trading (in Trading Contract), (3) the timeline of block creation (in Processing Contract).

4. The Deepchain model

Employ secure monetary penalty mechanism to guarantee fairness in gradient collecting and collaborative decryption.



Algorithm 3: $F^*_{GradientCollecting}$

```
1 Receive (input, $sid, T_t, pk^{psu}_{P_j}, C(\triangle \mathbf{W}), Proof_{PK_j}, d(\$Coin)$) from $pk^{psu}_{P_{j\in\{1,...,N\}}}$. Assert time $T_t < T_{t_1}$. Receive (input, $sid, T_t, pk^{psu}_{P_{j\in\mathbb{C}}}, C(\triangle \mathbf{W}), Proof_{PK_j}, H'$, $h' \times d(\$Coin)$) from $\mathcal{S}$. Assert time $T_t < T_{t_1}$.
2 Compute $fsver_1(C(\triangle \mathbf{W}), Proof_{PK_j})$ for $pk^{psu}_{P_{j\in\{1,...,N\}}}$, and record $\{1, ..., N\} \setminus \mathbb{C}'$.
3 Send(return, $d(\$Coin)$) to $pk^{psu}_{P_{j\in\{1,...,N\}\setminus\mathbb{C}'}}$ after $T_{t_1}$.
4 If $\mathcal{S}$ returns (continue, $H''$), then send (output, $Yes$ or $No$) to $pk^{psu}_{P_{j\in\{1,...,N\}}}$, and send (payback, $(h-h'')d(\$Coin)$) to $\mathcal{S}$, and send (extrapay, $d(\$Coin)$) to $pk^{psu}_{P_{j\in H''}}$, else if $\mathcal{S}$ returns (abort), send (penalty, $d(\$Coin)$) to $pk^{psu}_{P_{j\in\{1,...,N\}}}$.
```

4. The Deepchain model

Algorithm 4: $F^{*}_{CollaborativeDecryption}$

```
1 Receive ($input$,
  $sid, T_t, pk^{psu}_{P_j}, C, C_j, Proof_{CD_j}, d(\$Coin)$) from
  $pk^{psu}_{P_{j\in\{1,...,N\}}}$. Assert time $T_t < T_{t_5}$. Receive (input,
  $sid, T_t, pk^{psu}_{P_j} \in \mathbb{C}, C, C_j, Proof_{CD_j}, H'$,
  $h' * d(\$Coin)$) from $\mathcal{S}$. Assert time $T_t < T_{t_5}$.
2 Compute $fsver_2(C, C_j, Proof_{CD_j})$ for $pk^{psu}_{P_{j\in\{1,...,N\}}}$
  and record $\{1, ..., N\} \setminus \mathbb{C}'$.
3 Send($return$, $d(\$Coin)$) to $pk^{psu}_{P_{j\in\{1,...,N\}\setminus\mathbb{C}'}}$ after $T_{t_5}$;
4 If $\mathcal{S}$ returns (continue, $H''$), then send ($output$, $Yes$ or
  $No$) to $pk^{psu}_{P_{j\in\{1,...,N\}}}$, and send ($payback$,
  $(h - h'')d(\$Coin)$) to $S$, and send ($extrapay$, $d(\$Coin)$)
  to $pk^{psu}_{P_{j\in H''}}$, else if $S$ returns (abort), send ($penalty$,
  $d(\$Coin)$) to $pk^{psu}_{P_{j\in\{1,...,N\}}}$.
```

4. The Deepchain model

4.2.5 Consensus protocol

- Enables all participants to make a consensus upon some event in a decentralized environment
- Blockwise-BA protocol
- (a) Leader selection. A leader is randomly chosen from workers who collect transactions into block block.
- (b) Committee agreement: After leader verification, the selected block is sent to the committee. Each participant in the committee verifies the transactions processed by leader
- (c) Neighbor gossip.: Suppose blocki has been agreed on by the committee, then participants in the committee are responsible for telling their neighbors block, by using the popular gossip protocol.

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

5. Security analysis

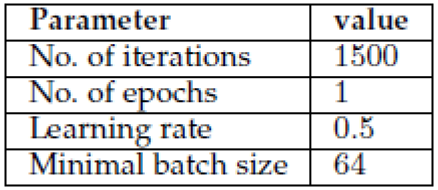
Confidentiality guarantee for gradients: DeepChain employs Threshold Paillier algorithm that provide additive homomorphic property.

- Auditability of gradient collecting and parameter update: Ensures that any party can audit the correctness of encrypted gradients and decryption shares in gradient collecting and parameter updating.
- Fairness guarantee for collaborative training: Employ two security mechanisms in Blockchain to enhance fairness during collaborative training, namely the trusted time clock mechanism and secure monetary penalty mechanism.

6. Implementation and evaluation

- Use Corda V3.0 [55] to simulate DeepChain for its adaptability and simplification.
- A decentralized ledger that has some features of Bitcoin and Ethereum [56], such as data sharing based on need-to-know basis and deconflicting transactions with pluggable notaries.
- Popular MINIST dataset
- Threshold Paillier algorithm is implemented in JAVA within 160-line codes.
- CordaDeepChain, TrainAlgorithm, and CryptoSystem.
- DeepChain on a desktop computer with 3.3GHz Intel(R) Xeon(R) CPU 16 GB memory.

TABLE 2
Training configuration



Parameter	value
No. of iterations	1500
No. of epochs	1
Learning rate	0.5
Minimal batch size	64

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

6. Implementation and evaluation

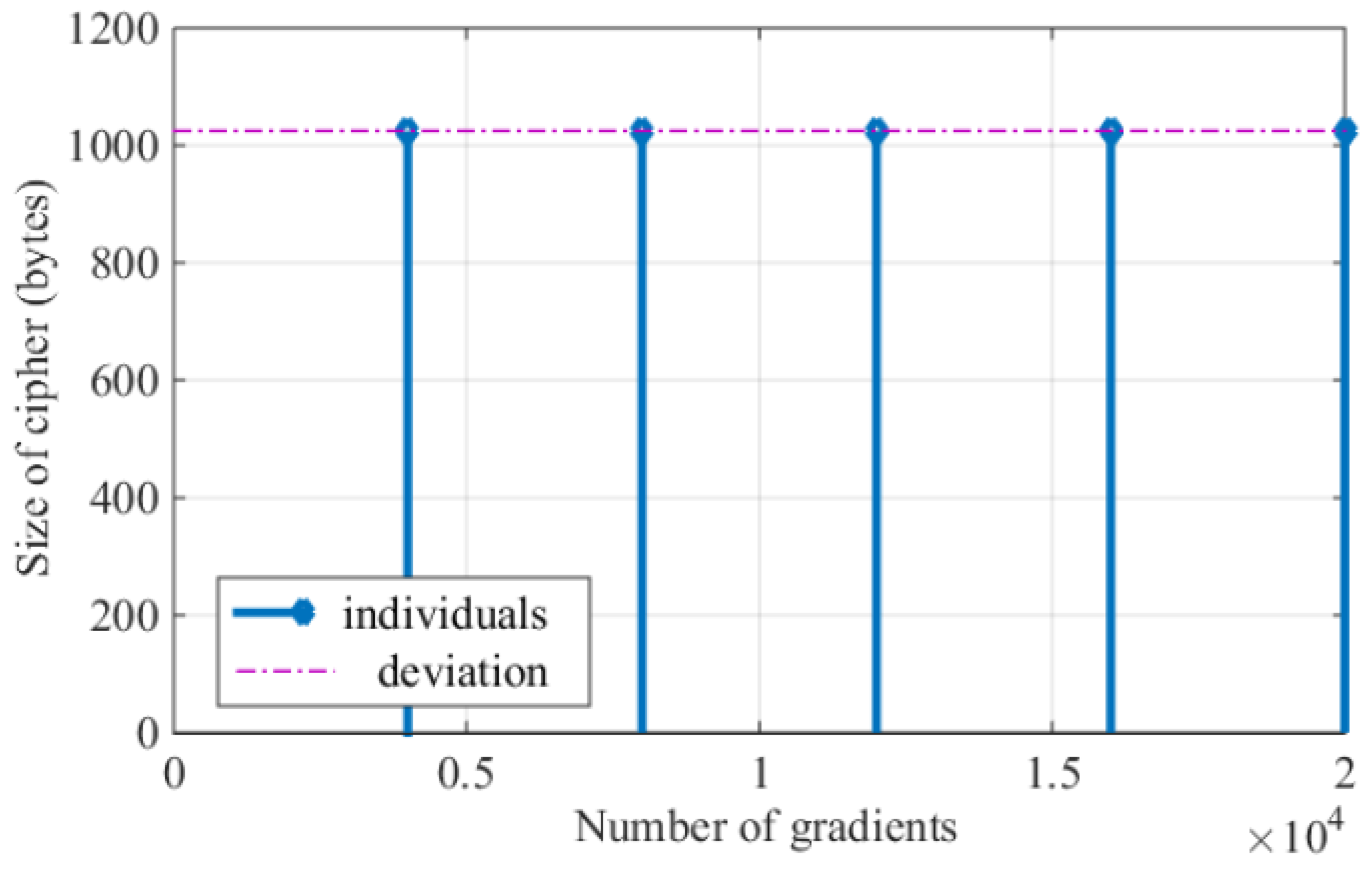


Fig. 4. Impact of No. of gradients on cipher size.

6. Implementation and evaluation

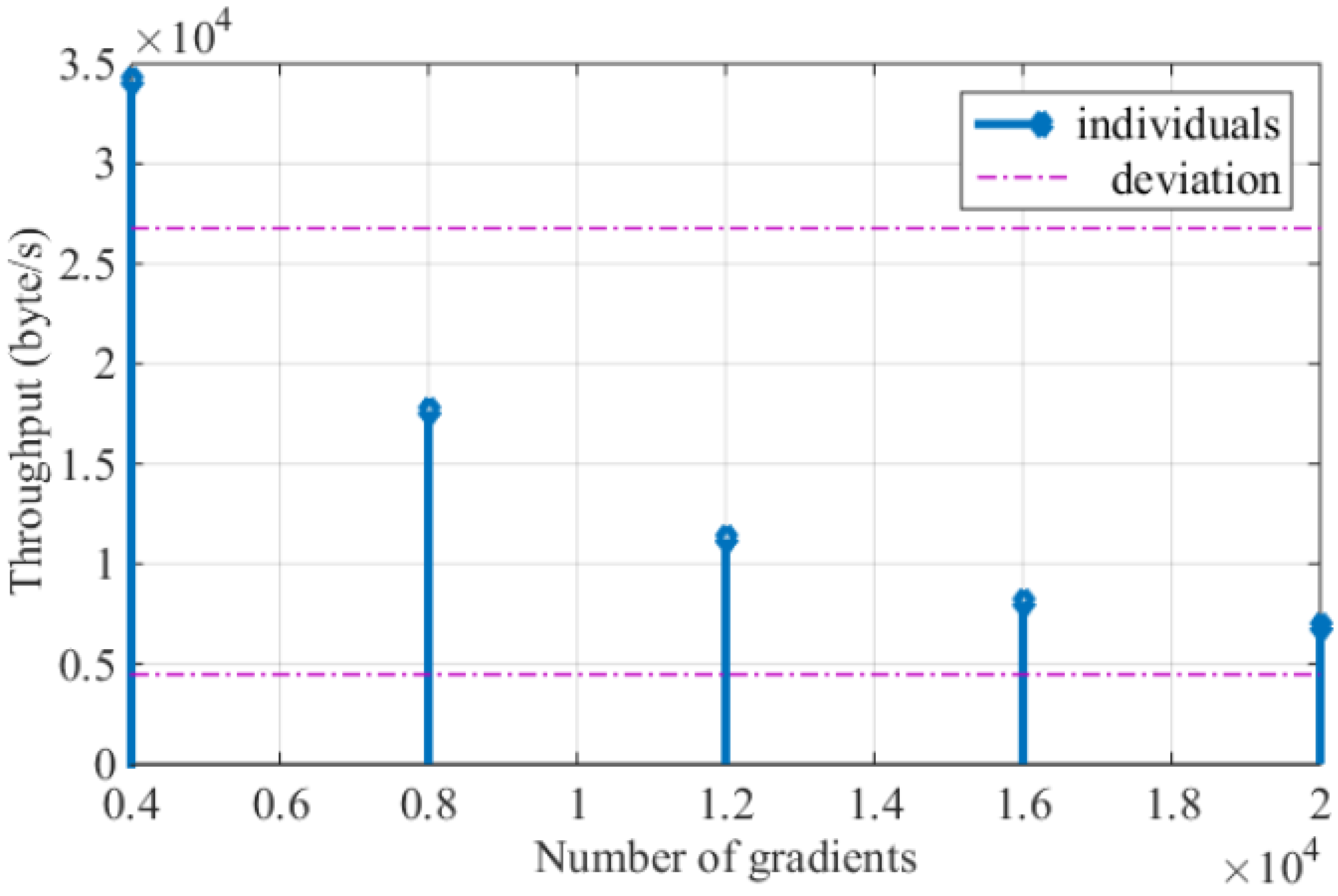


Fig. 5. Impact of No. of gradients on throughput.

3. 6. Implementation and evaluation

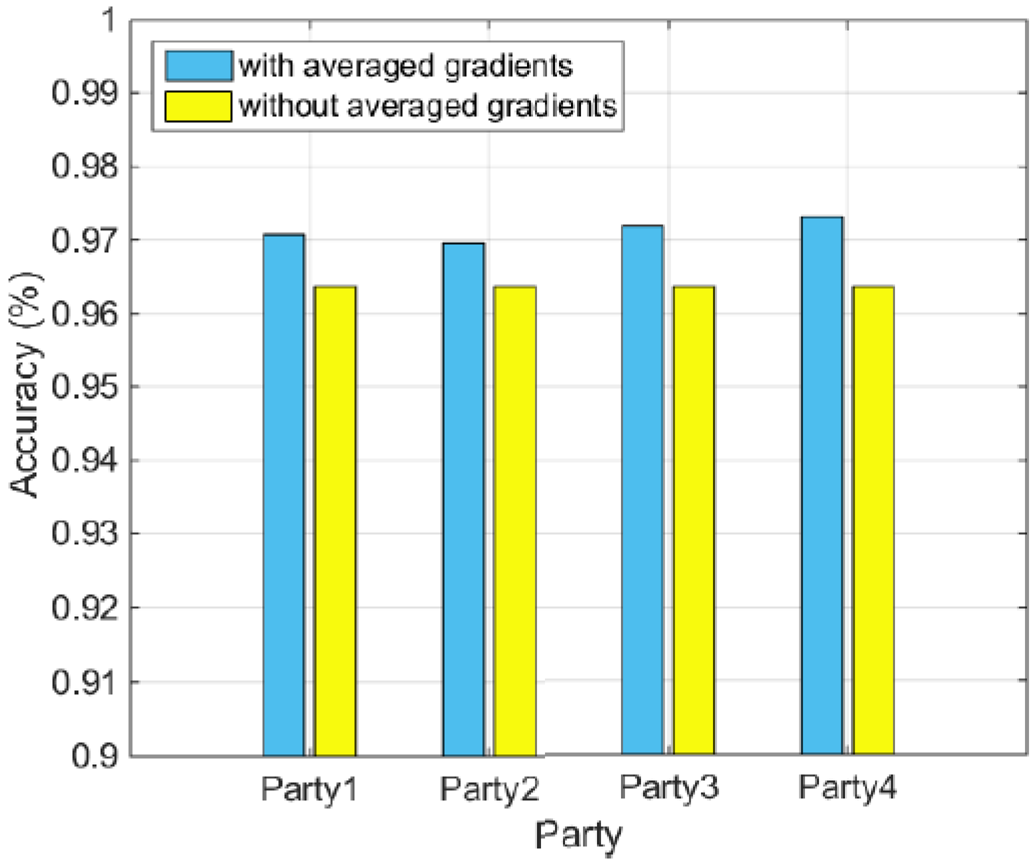


Fig. 6. Training accuracy for the case of four parties.

6. Implementation and evaluation

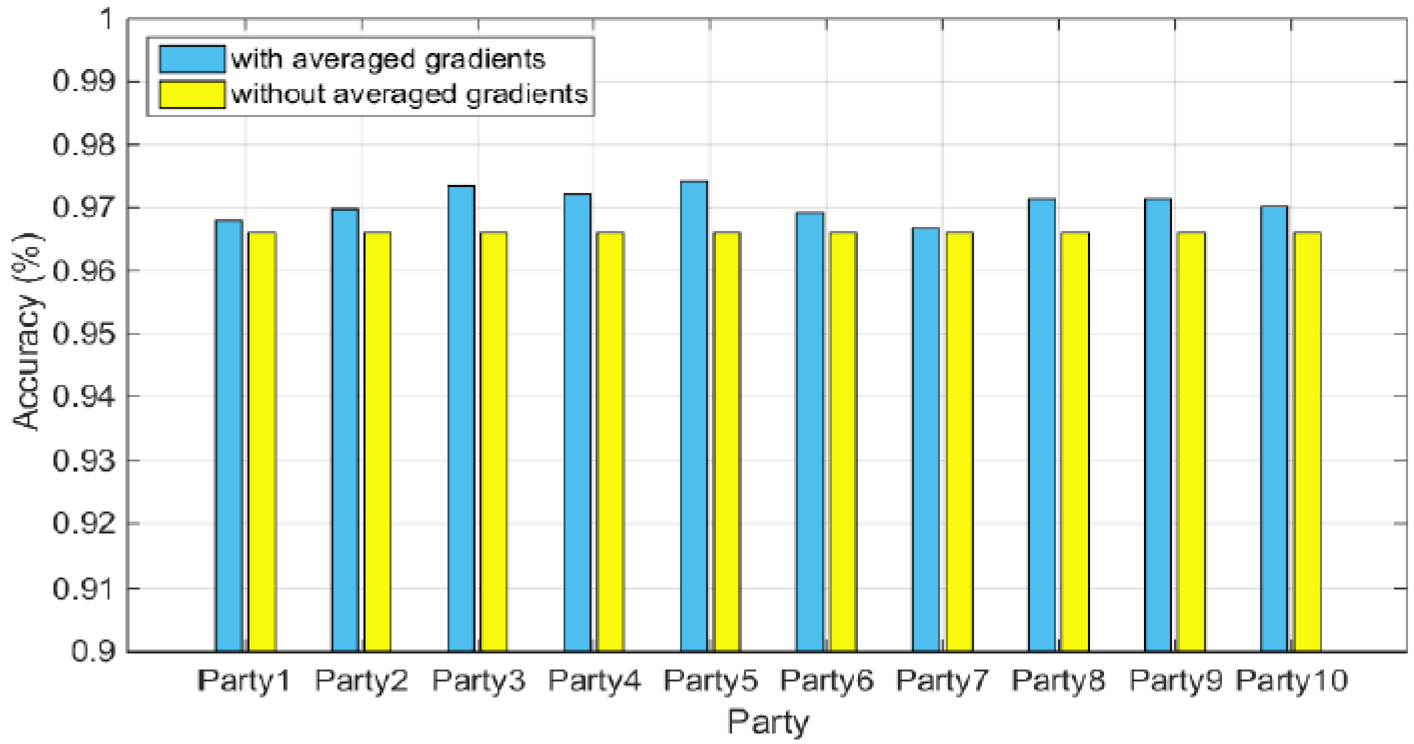


Fig. 7. Training accuracy for the case of ten parties.

6. Implementation and evaluation

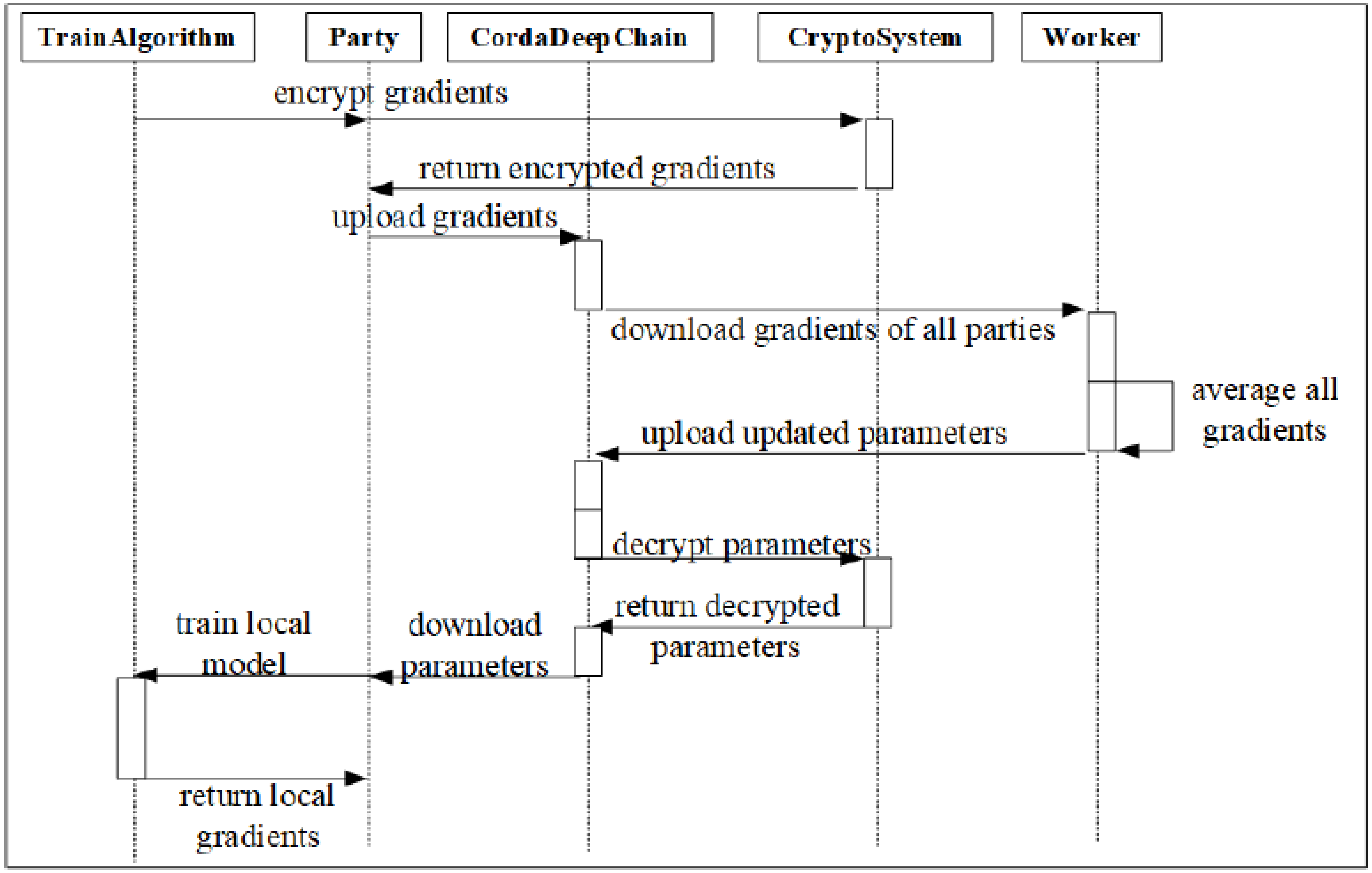


Fig. 8. Interaction in the entire collaborative training process.

6. Implementation and evaluation

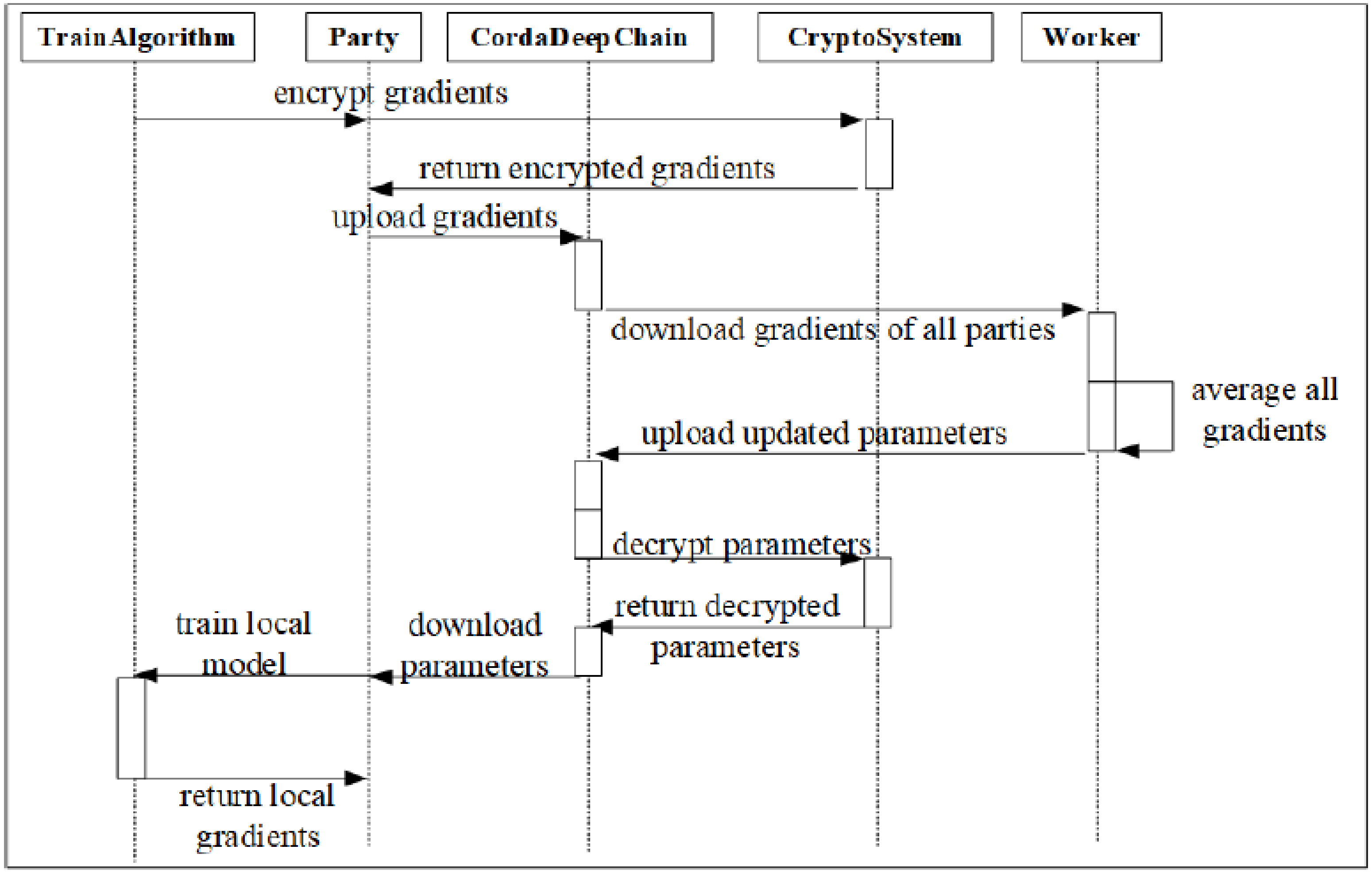


Fig. 8. Interaction in the entire collaborative training process.

6. Implementation and evaluation

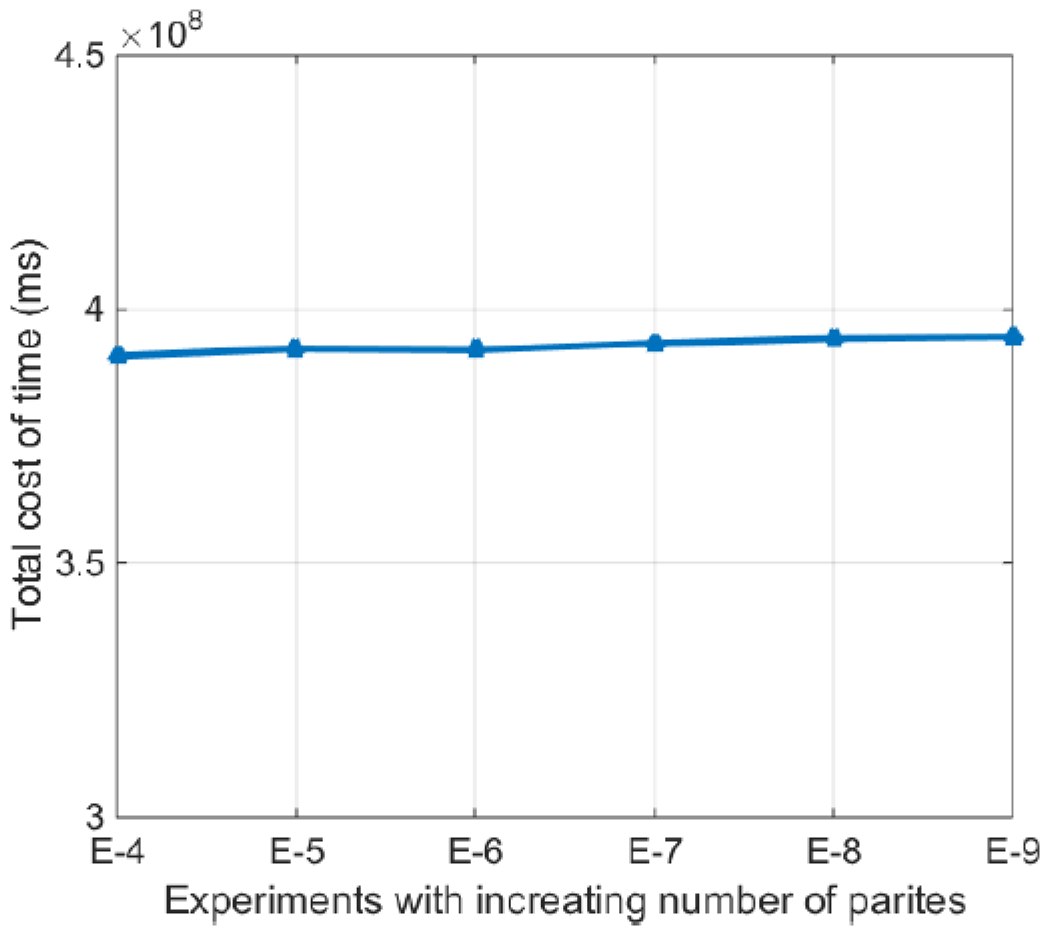


Fig. 9. Total cost of time in different experiments with increasing number of parites.

6. Conclusion

- Presented DeepChain, a robust and fair decentralized platform based on Blockchain for secure collaborative deep training.
- Achieve three security goals, namely confidentiality, auditability, and fairness.
- Discussed the significance of DeepChain in a long-term way.

Reference

- Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., & Luo, W. (2018). Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. Cryptology ePrint Archive, Report 2018/679.

Opinion

- Due to network bandwidth and data privacy concerns, it is impractical and often unnecessary to send all the data to a remote cloud.
- As a result, research organizations estimate that over 90% of the data will be stored and processed locally [3].
- Local data storing and processing with global coordination is made possible by the emerging technology of mobile edge computing (MEC).
- To analyze large amounts of data and obtain useful information for the detection, classification, and prediction of future events, dachine learning techniques.
- However, the distributed deep learning on MEC tend to be challenging due to the following reason.

✓ Single point of failure,

✓ Privacy leakage

✓ Training data insufficiency

✓ Data poisoning attacks

Can support a secure distributed learning at the device level and provide data integrity and confidentially in the IoT network.

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory