Hardware Software Co-optimizations for Efficient Privacy Preserving Computing in AIoT Devices

Weifeng Zhang
Alibaba Cloud Infrastructure

2021-11-03
Why data privacy and protection in AI?
Main approaches for privacy protection
Heterogeneous computing for AI acceleration
Speeding up privacy preserving computing for AI
With artificial intelligence quickly growing to maturity, more and more AI systems have been deployed in scenarios like logistic facilities, hospitals, streets and homes

- speech recognition in home voice assistants (Siri, Alexa, etc.)
- object detection in city traffic management
- 3D segmentation in medical diagnosis

Massive amount of data was generated and then processed at the edge and data centers

Edge Computing Devices
(IoT, ARM, RISC-V, DSP)

Cloud Data Centers
(CPUs, GPUs, FPGAs, NPUs)
Several challenges have arisen:

- Is the system vulnerable to various attacks?
- Is the system robust to small changes in the input data?
- Has the data been collected in an ethical manner?
- Does the system protect your data and personal identity?
- And more . . .

**HIPAA** 1996: Health Insurance Portability and Accountability ACT, US federal law for creation of national standards to protect patient health information

The EU General Data Protection Regulation (**GDPR**), which governs how personal data of individuals in the EU may be processed and transferred, went into effect on May 25, 2018

California Consumer Privacy Act of 2018 (**CCPA**), intended to enhance privacy rights and consumer protection for residents of California, US

China Personal Information Protection Regulation, effective on November 1st, 2021
New Focuses of AI Properties

The set of properties for AI systems needs to be extended beyond AI functionality and efficiency to achieve trustworthiness: security, privacy, robustness, reliability, interpretability, etc.

# Privacy Preserving Computing

<table>
<thead>
<tr>
<th>Technologies</th>
<th>Performance</th>
<th>Generality</th>
<th>Security</th>
<th>Root of Trust</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trusted Execution Environment (TEE)</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Yes</td>
<td>Hardware supported enclave for data isolation and protection</td>
</tr>
<tr>
<td>Trusted Computing Accelerator</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Yes</td>
<td>In early development stage</td>
</tr>
<tr>
<td>Cryptography</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Homomorphic Encryption (H.E.)</td>
<td>Extremely low</td>
<td>High</td>
<td>High</td>
<td>No</td>
<td>Only for limited types of operations, such as ones with linear property</td>
</tr>
<tr>
<td>Multi-Party Computation (MPC)</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>No</td>
<td>Difficult to scale</td>
</tr>
<tr>
<td>Federated Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decentralized iterative learning</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>May</td>
<td>For AI training via MPC, DP, H.E. or other secure hardware</td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential Privacy (DP)</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>No</td>
<td>Based on statistics for certain data distribution</td>
</tr>
<tr>
<td>Data Masking</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>May</td>
<td>Data Obfuscation, often leading to performance &amp; accuracy loss</td>
</tr>
</tbody>
</table>

*Make data available but invisible*
# Privacy Preserving Computing

<table>
<thead>
<tr>
<th>Technologies</th>
<th>Performance</th>
<th>Generality</th>
<th>Security</th>
<th>Root of Trust</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted Computing</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Yes</td>
<td>Hardware supported enclave for data isolation and protection</td>
</tr>
<tr>
<td>Trusted Computing Environment (TEE)</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Yes</td>
<td>In early development stage</td>
</tr>
<tr>
<td>Homomorphic Encryption (H.E.)</td>
<td>Extremely low</td>
<td>High</td>
<td>High</td>
<td>No</td>
<td>Only for limited types of operations, such as ones with linear property</td>
</tr>
<tr>
<td>Multi-Party Computation (MPC)</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>No</td>
<td>Difficult to scale</td>
</tr>
<tr>
<td>Decentralized iterative learning</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>May</td>
<td>For AI training via MPC, DP, H.E. or other secure hardware</td>
</tr>
<tr>
<td>Differential Privacy (DP)</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>No</td>
<td>Based on statistics for certain data distribution</td>
</tr>
<tr>
<td>Data Masking</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>May</td>
<td>Data Obfuscation, often leading to performance &amp; accuracy loss</td>
</tr>
</tbody>
</table>
What are the problems with TEE?

TEE is a trusted execution environment, where data is decrypted before execution and encrypted when coming into and going out the environment.

<table>
<thead>
<tr>
<th>Three criteria to measure TEE[2]: functionality, security, and deployability.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functionality</strong>: physical protection of the data throughout its entire life cycle, including protection of input/output, attestation, transformation and execution, and storage</td>
</tr>
<tr>
<td><strong>Security</strong>: mechanisms to avoid specific attacks on the system, such as data isolation, sanitization, damage control, etc.</td>
</tr>
<tr>
<td><strong>Deployability</strong>: cost, overhead, performance, scalability of the system, etc.</td>
</tr>
</tbody>
</table>

TEE usually has limited computation resource (compute, memory). Computation inside TEE won’t enjoy the performance benefit of powerful accelerators such as GPU and NPU.

- ARM TrustZone based TEE is typically bound to a single core with limited parallelism
- Intel SGX enclave may cause slowdown as much as 20~30X
- H.E. may suffer performance degradation up to 1000X

Why data privacy and protection in AI?

Main approaches for privacy protection

Heterogeneous computing for AI acceleration

Speeding up privacy preserving computing for AI
Heterogeneous computing has come to play.
AI training performance has achieved over 20X speedup, much beyond Moore’s Law.
Accelerating AI with Heterogeneous Computing

**Current solution for AI acceleration:**
- Fragmented solutions with incompatible algorithms, software stack, libraries and APIs
- Significant effort for software adaptation and optimizations
- Tight coupling with vendor specific solution and low code reuse, leading to high maintenance cost

**HALO / ODLA heterogeneous acceleration scheme:**
- **Code generated once and executed anywhere**
- Quickly support various new accelerator architecture and deploy to various scenarios
- Dynamical switch computation among heterogeneous devices with seamless migration
- Lower the execution cost
Heterogeneous Computing Paradigm for AI Acceleration

- Data parallelism
- Model parallelism
- Pipeline parallelism
- etc.

Optimization techniques: cost model for performance
- Model / task partitioning for parallelism
- Device placement (GPU, NPU, DSP, etc.)

Accelerating Trusted Computing with Untrusted Hardware Accelerators

Use the similar optimization scheme in heterogeneous computing to handle the new dimension of parallelism: trustworthiness
- Task partition for parallelism and privacy protection
- Device placement (trusted and untrusted devices)
- Target fragmented secure hardware

Model partitioning heuristics: performance centric ➔ privacy/security centric
Basic Ideas of Privacy-Aware Model Partitioning:
- Compute the “similarity” (or a range of similarities [min, max]) of feature maps with the original input for each layer.
- Find the layer where the similarity (or the minimal of similarities) appears to reach a minimal (plateau).
- Divide the model at this layer: prior layers as Model Part1 for TEE execution, and the rest as Model Part2 for REE acceleration.

Data similarity assessment:
- Kullback-Leibler (KL) Divergence to gauge the similarity of the middle feature map data with the original input.
- GAN based discrimination between the feature map data and the original input.
Privacy-Aware AI model partitioning based on KL divergence

Kullback-Leibler (KL) Divergence

- Compute similarity of distributions between each feature map and the original data input
- The lower the similarity is, the more difficult to recover (reverse engineer) the original data
Data Obfuscation

Pros:
- Sensitive data stays within the local storage
- Hard to reverse / recover the original data
- Key features remain during data obfuscation so it’s relatively easy to re-train

Cons:
- Model accuracy adversely affected
- Re-training typically required

Optimizations:
- Model restructuring with hardware software co-optimization: total FLOPS can be significantly reduced
- Strength reduction of the data masking operations for more efficient computation
Collaborative Edge-Cloud Training & Inference Paradigm

- Offload the computation to cloud due to insufficient resource at edge
- Data sharing & processing requirement at cloud - but can’t transport data directly

**Training session:**
- Partition the model to two portions: local and cloud
- Mask the data and send it to cloud for further training

**Inference session:**
- Do inference at edge, and send the masked output to cloud
- Do further inference on cloud using the masked input
Alibaba’s **SinianTrust** is a unified hardware software co-optimization framework to accelerate privacy-preserving computing based on **ODLA.trust** API.

- Intel SGX based TEE
- ARM Trustzone based TEE
- GPU / NPU / DSP for REE acceleration
- More coming ...
GitHub:

✓ HALO: heterogeneity-aware lowering & optimization
✓ ODLA: open deep learning API
✓ ODLA.trust: for TEE and privacy-preserving computing (coming soon)

https://github.com/alibaba/heterogeneity-aware-lowering-and-optimization