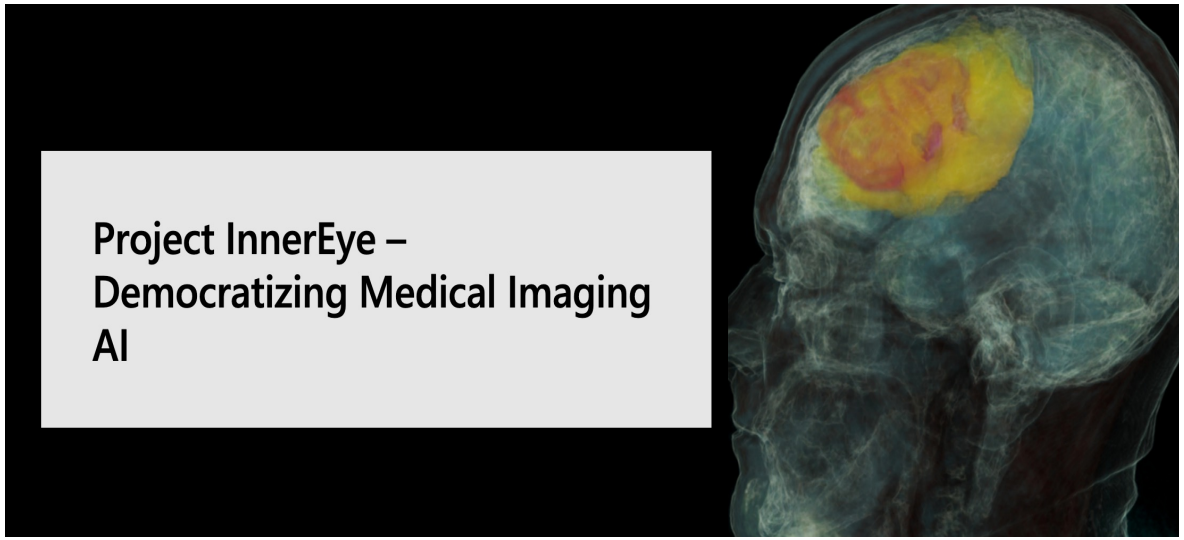
SIGuard: Guarding Secure Inference with Post Data Privacy

Xinqian Wang[*], Xiaoning Liu[*], Shangqi Lai[†], Xun Yi[*], Xingliang Yuan[‡]

* RMIT University, † CSIRO's Data61, ‡ The University of Melbourne

Machine Learning as a Service (MLaaS) offerings at the Cloud



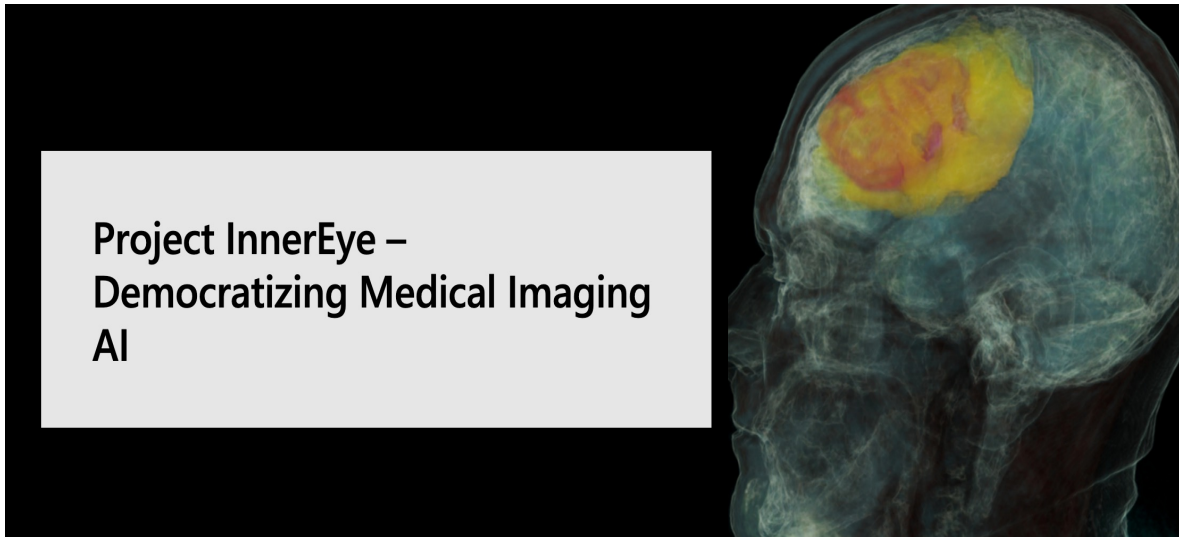
Google Vision API

Amazon SageMaker

[1] Microsoft Project InnerEye (2020). https://www.microsoft.com/en-us/research/project/medical-image-analysis/
[2] Google DeepMind Health (2020). https://deepmind.com/blog/announcements/deepmind-health-joins-google-health
[3] Kuna AI (2017). https://getkuna.com/ blogs / news / 2017 - 05 - 24 - introducing - kuna-ai.

Machine Learning as a Service (MLaaS) offerings at the Cloud

- Ready-made intelligence for a wide spectrum of applications
  - Medical imaging [1], clinical trial and research [2], home monitoring [3]



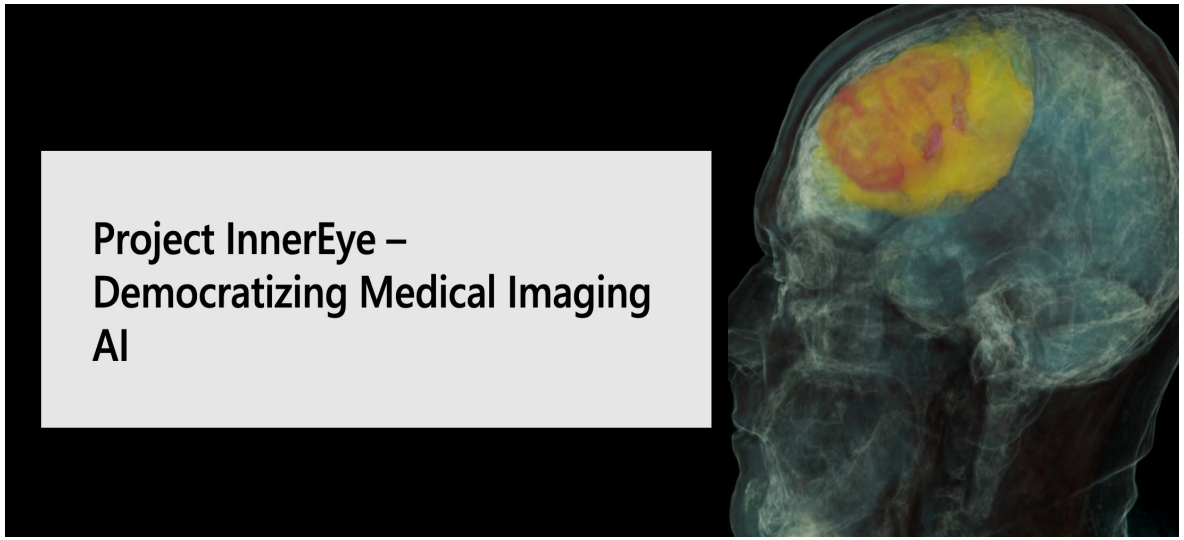
Google Vision API

Amazon SageMaker

[1] Microsoft Project InnerEye (2020). https://www.microsoft.com/en-us/research/project/medical-image-analysis/
[2] Google DeepMind Health (2020). https://deepmind.com/blog/announcements/deepmind-health-joins-google-health
[3] Kuna AI (2017). https://getkuna.com/ blogs / news / 2017 - 05 - 24 - introducing - kuna-ai.

Machine Learning as a Service (MLaaS) offerings at the Cloud

- Ready-made intelligence for a wide spectrum of applications
  - Medical imaging [1], clinical trial and research [2], home monitoring [3]
- Essential offerings for cloud service providers
  - Google Cloud Vision API, Amazon AWS SageMaker



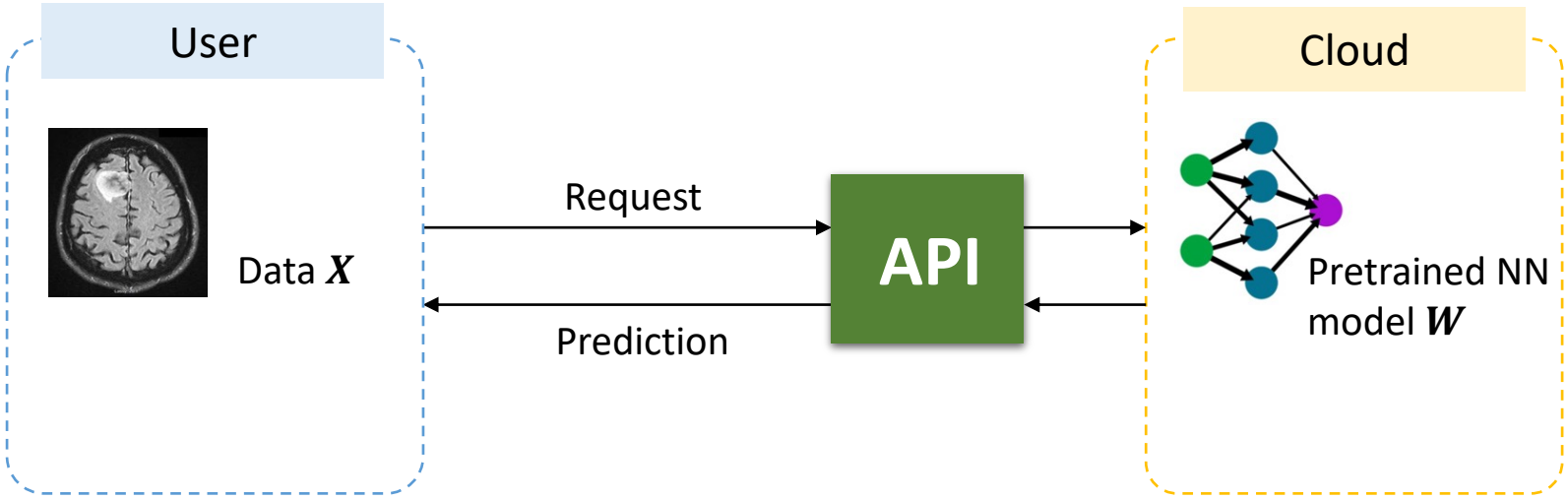
Google Vision API

Amazon SageMaker

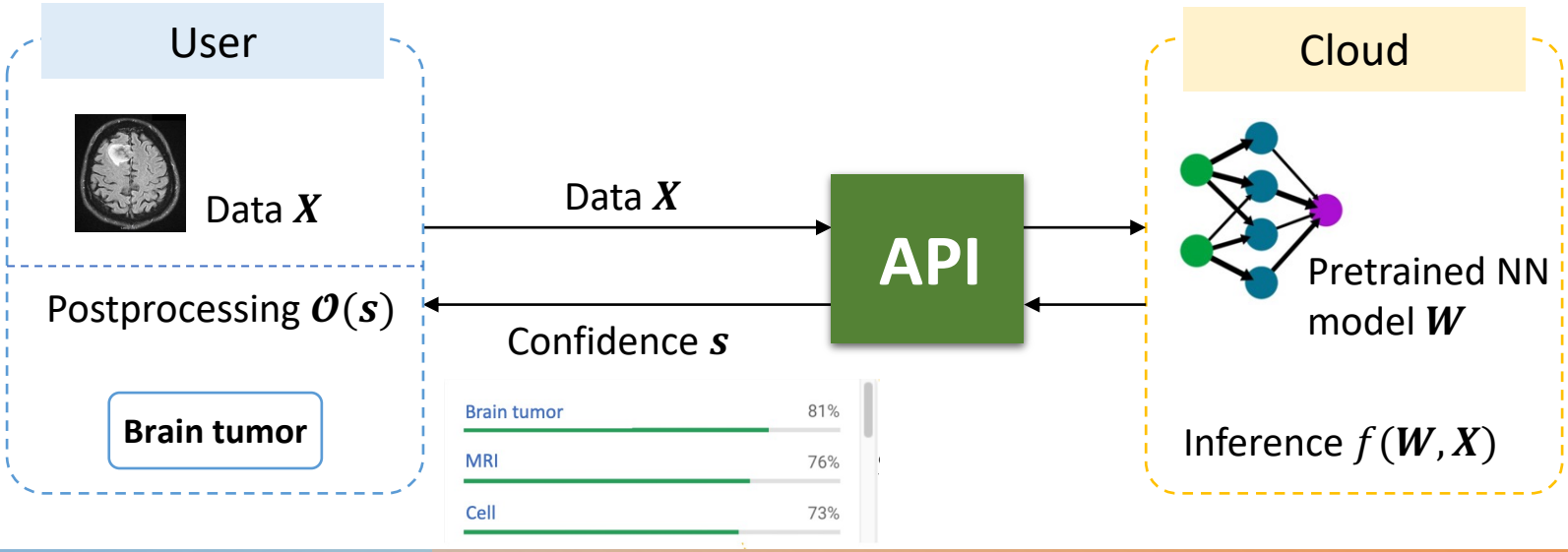
[1] Microsoft Project InnerEye (2020). https://www.microsoft.com/en-us/research/project/medical-image-analysis/
[2] Google DeepMind Health (2020). https://deepmind.com/blog/announcements/deepmind-health-joins-google-health
[3] Kuna AI (2017). https://getkuna.com/ blogs / news / 2017 - 05 - 24 - introducing - kuna-ai.

A typical workflow for MLaaS inference

- Cloud capitalizes on neural networks (NNs) to offer prediction services (e.g., medical diagnostics)
- User leverages the service to make a prediction over its data (e.g., brain MRI)

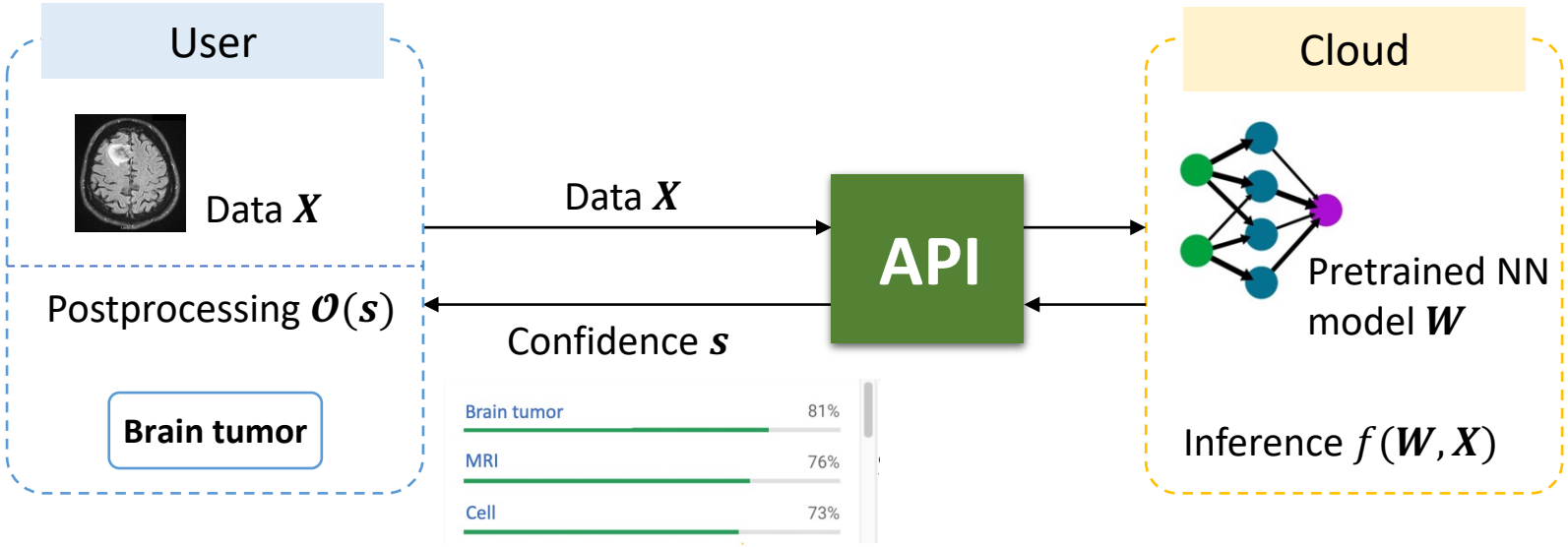


A typical workflow for MLaaS inference



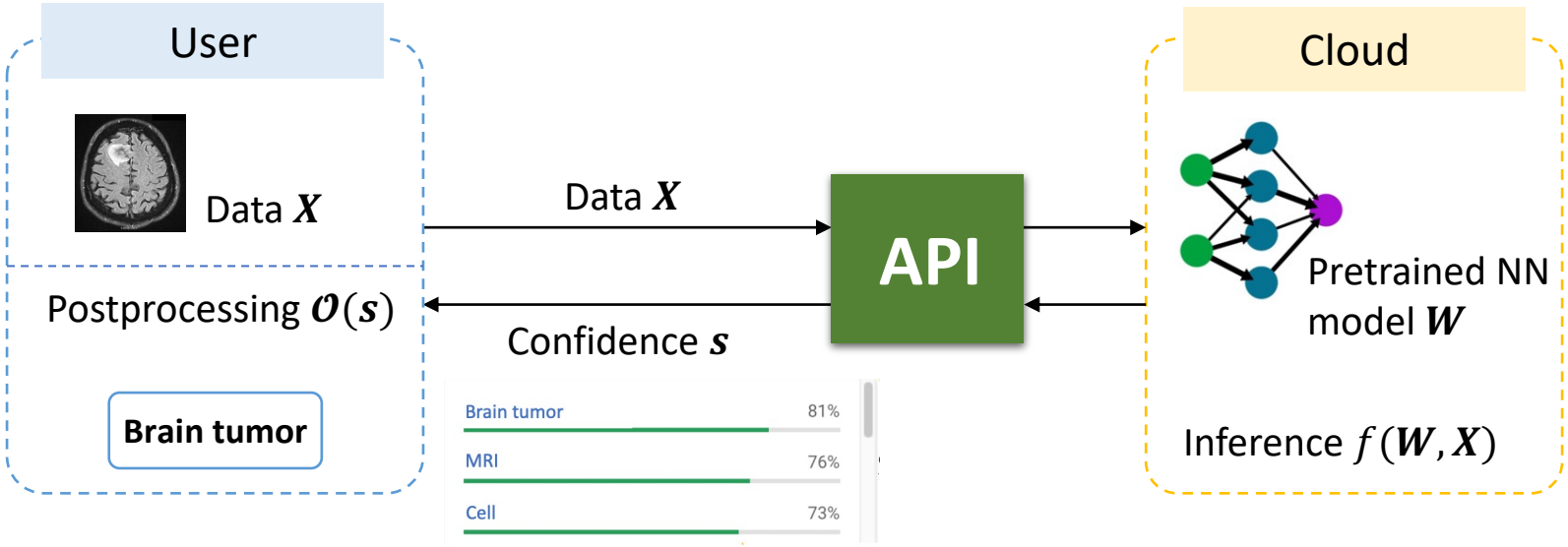
A typical workflow for MLaaS inference

- User feeds data to the model through the API



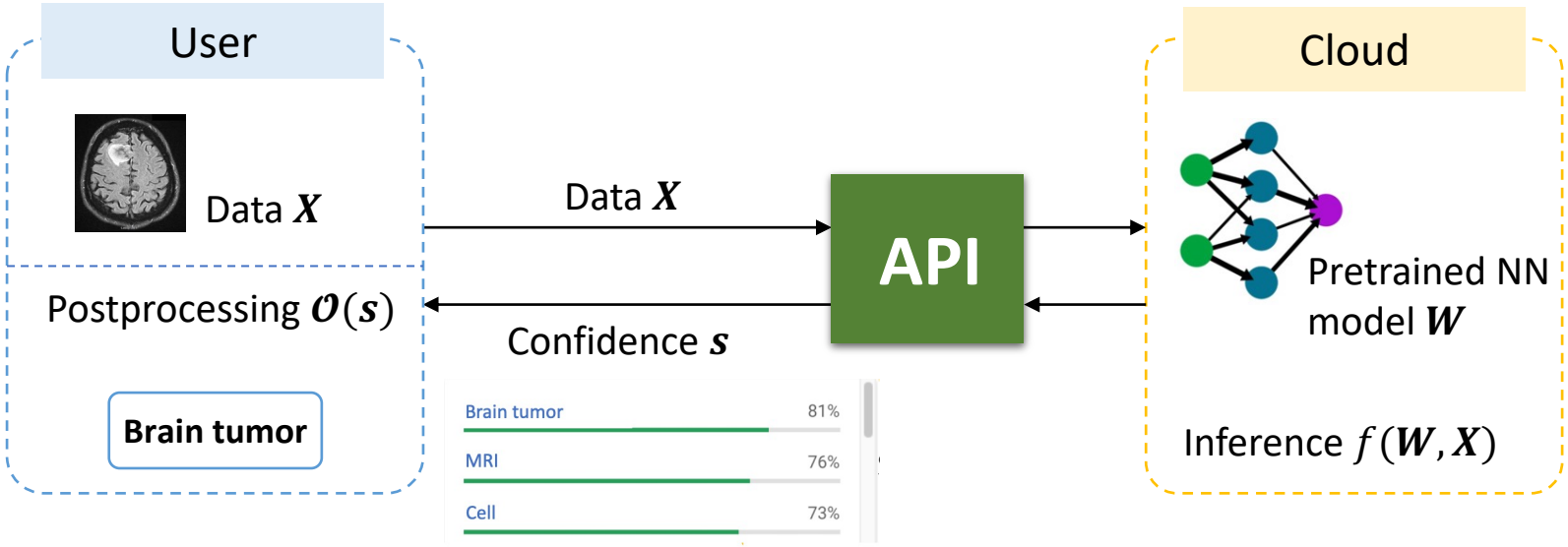
A typical workflow for MLaaS inference

- User feeds data to the model through the API
- Cloud hosts a pre-trained NN model and runs an inference function over user's data

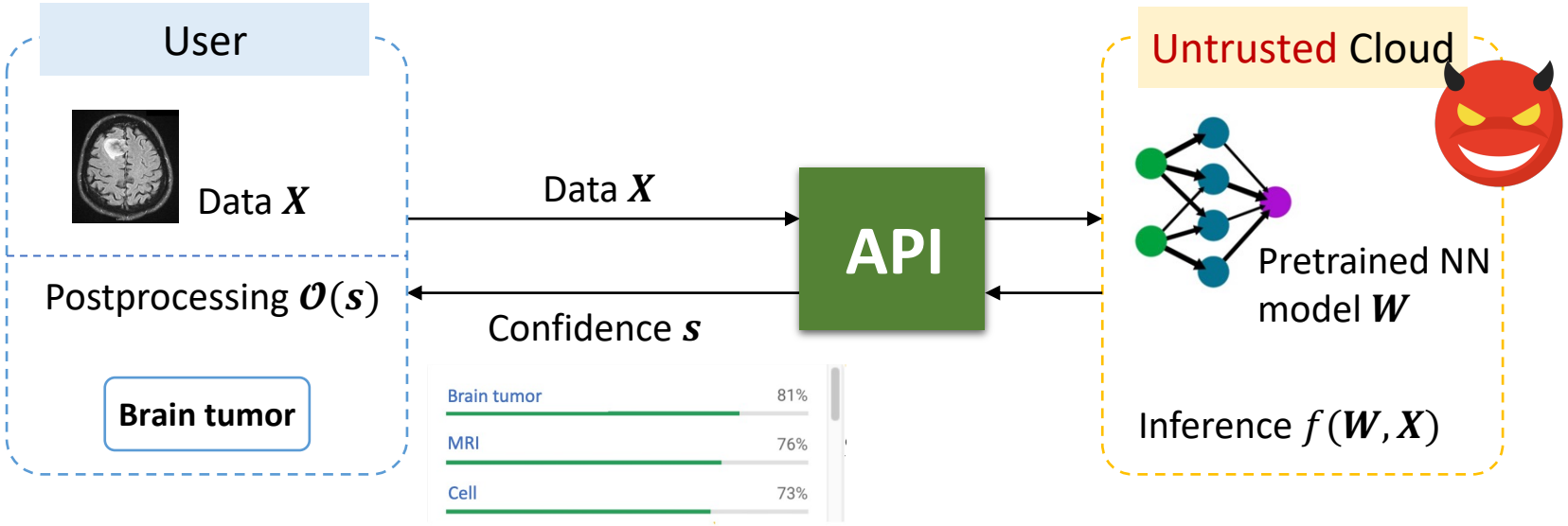


A typical workflow for MLaaS inference

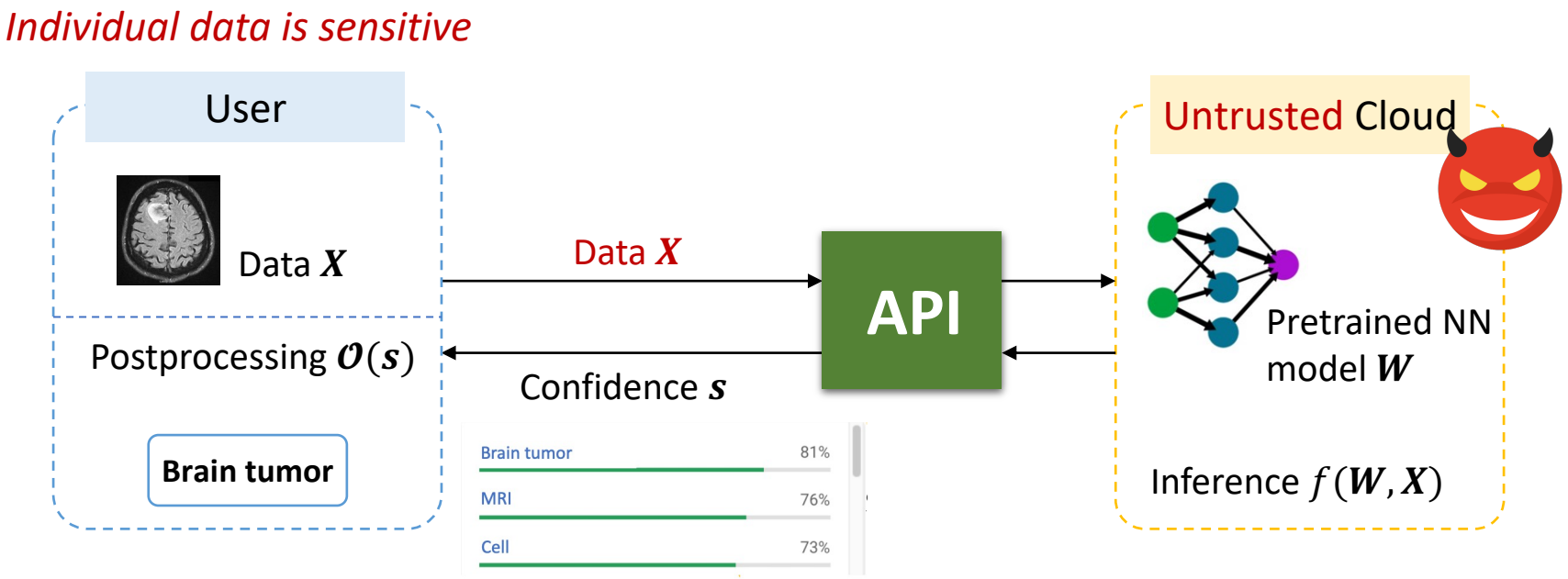
- User feeds data to the model through the API
- Cloud hosts a pre-trained NN model and runs an inference function over user's data
- User receives confidence vectors to choose a quality prediction



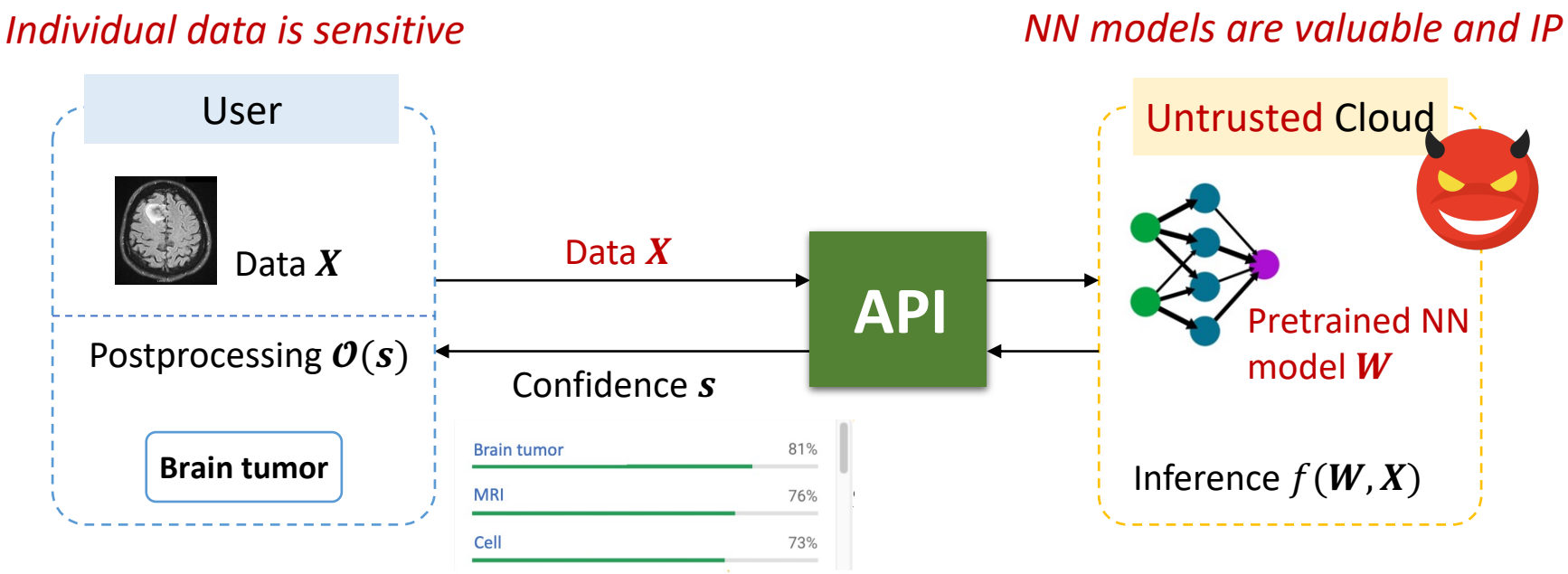
Privacy concerns in MLaaS inference service



Privacy concerns in MLaaS inference service

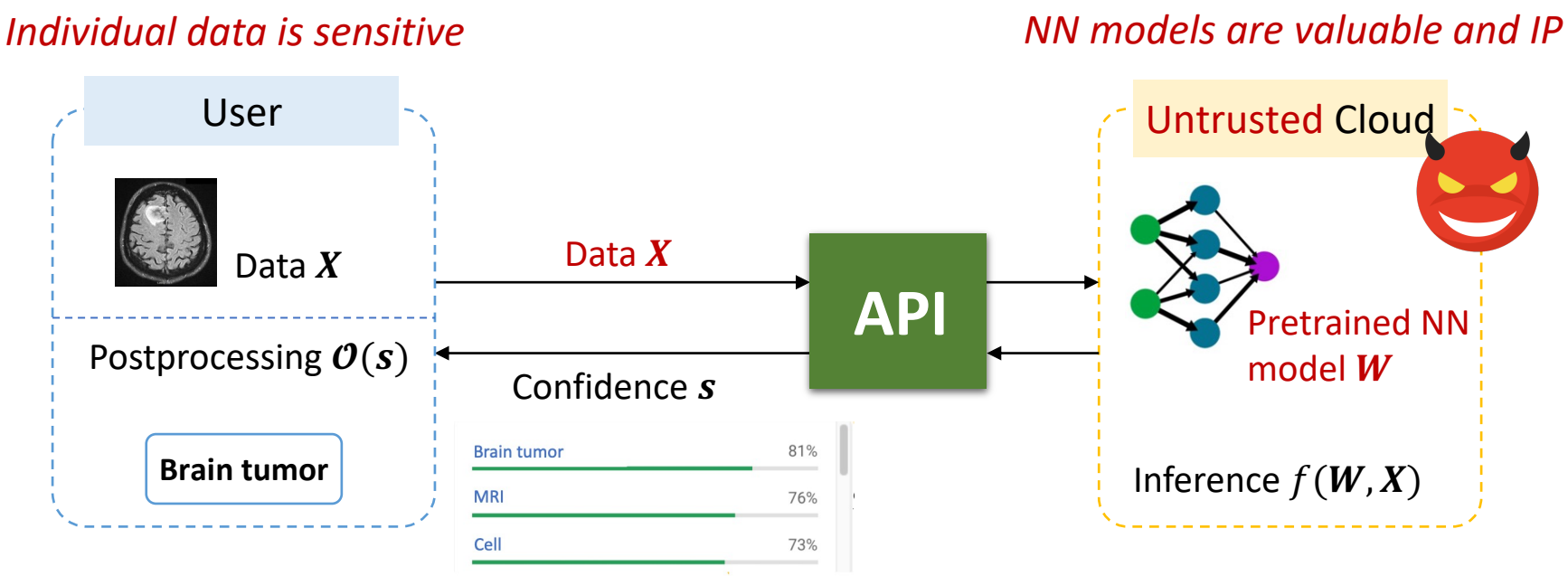


Privacy concerns in MLaaS inference service



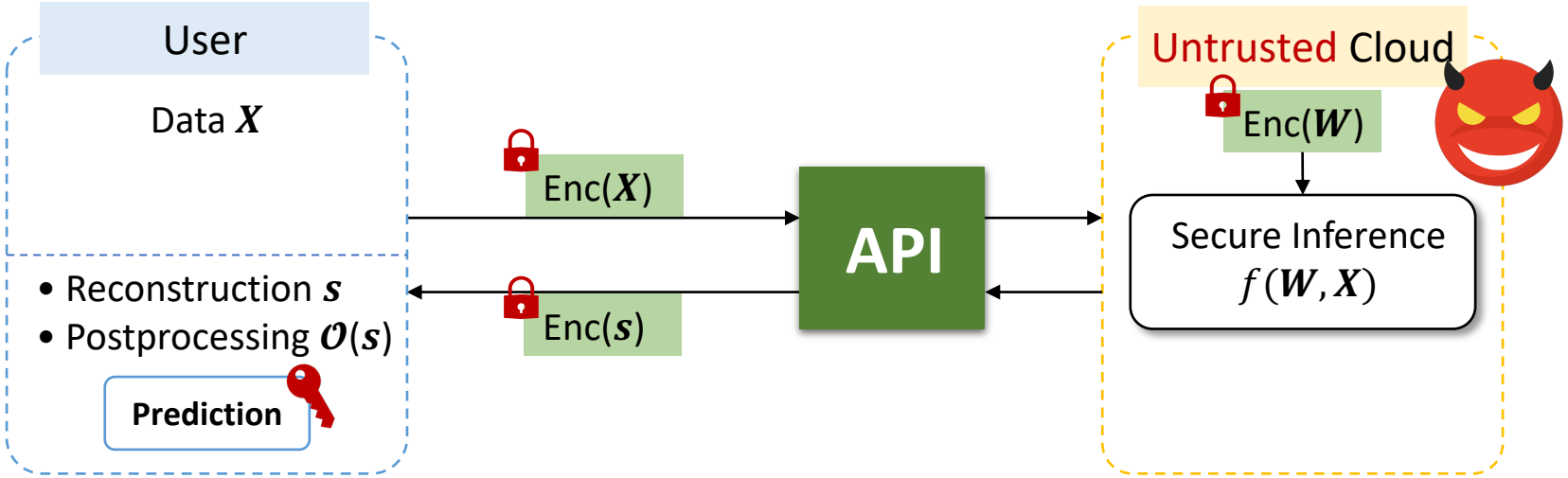
Privacy concerns in MLaaS inference service

- ***Input Privacy*** - sensitive information using proprietary model $\boldsymbol{W}$ to classify sensitive individual data $\boldsymbol{X}$



Secure inference to guarantee input privacy

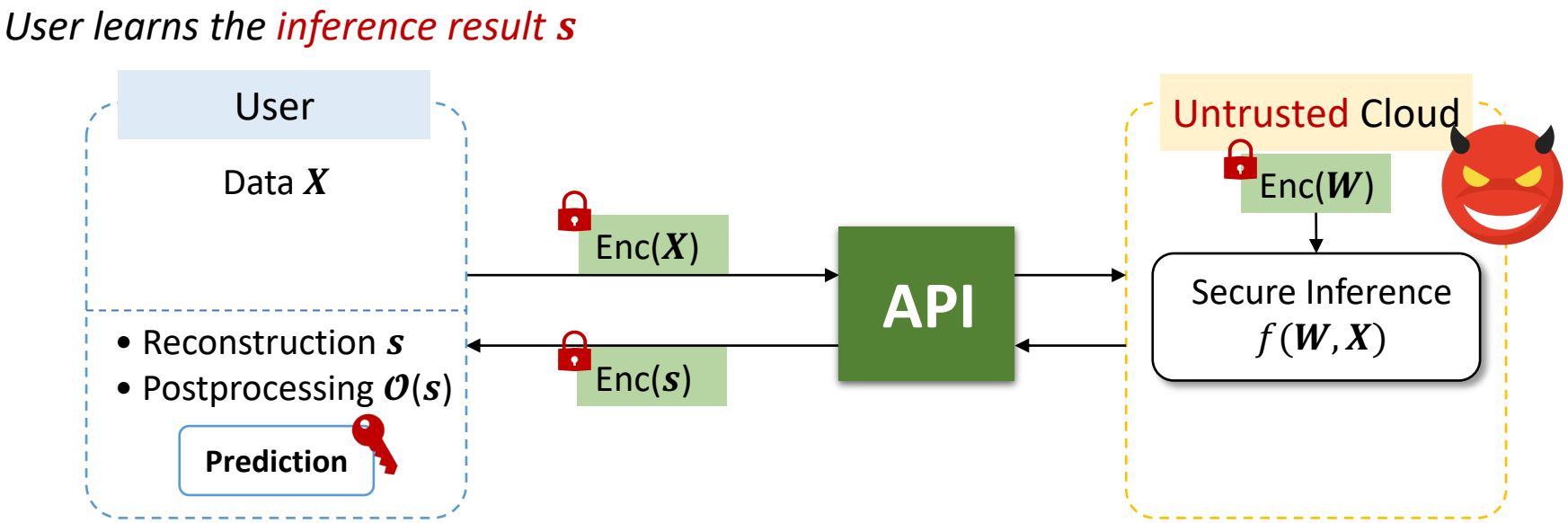
- Privacy-preserving machine learning (PPML) inference service, i.e., secure inference $f(\boldsymbol{W}, \boldsymbol{X})$
  - A user and a model owner upload encrypted data Enc($\boldsymbol{X}$) and model Enc($\boldsymbol{W}$) to the cloud
  - Opt for secure multiparty computation (MPC) [1, 2, 3] for efficiency



[1] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, “Delphi: A cryptographic inference service for neural networks,” in USENIX Security, 2020.
[2] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.
[3] Mohassel, Payman, and Peter Rindal. "ABY3: A mixed protocol framework for machine learning." Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018.

Secure inference to guarantee input privacy

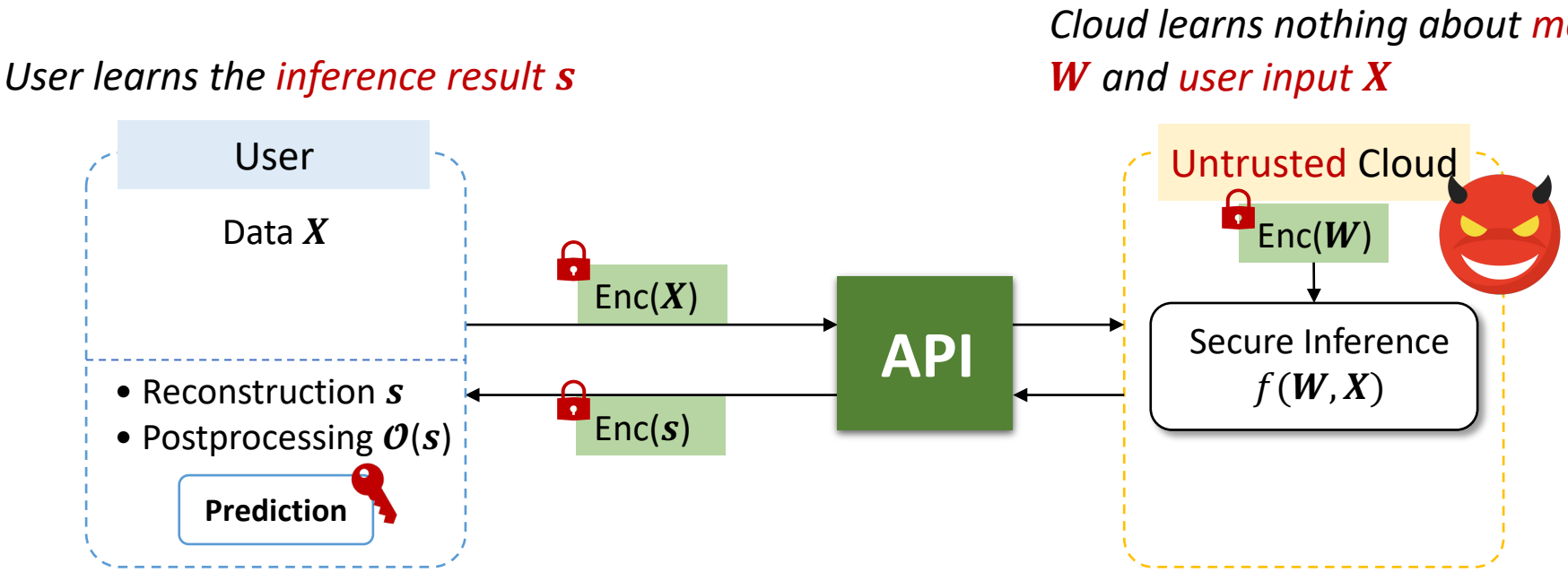
- Privacy-preserving machine learning (PPML) inference service, i.e., secure inference $f(\boldsymbol{W}, \boldsymbol{X})$
  - A user and a model owner upload encrypted data Enc($\boldsymbol{X}$) and model Enc($\boldsymbol{W}$) to the cloud
  - Opt for secure multiparty computation (MPC) [1, 2, 3] for efficiency



[1] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, "Delphi: A cryptographic inference service for neural networks," in USENIX Security, 2020.
[2] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.
[3] Mohassel, Payman, and Peter Rindal. "ABY3: A mixed protocol framework for machine learning." Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018.

Secure inference to guarantee input privacy

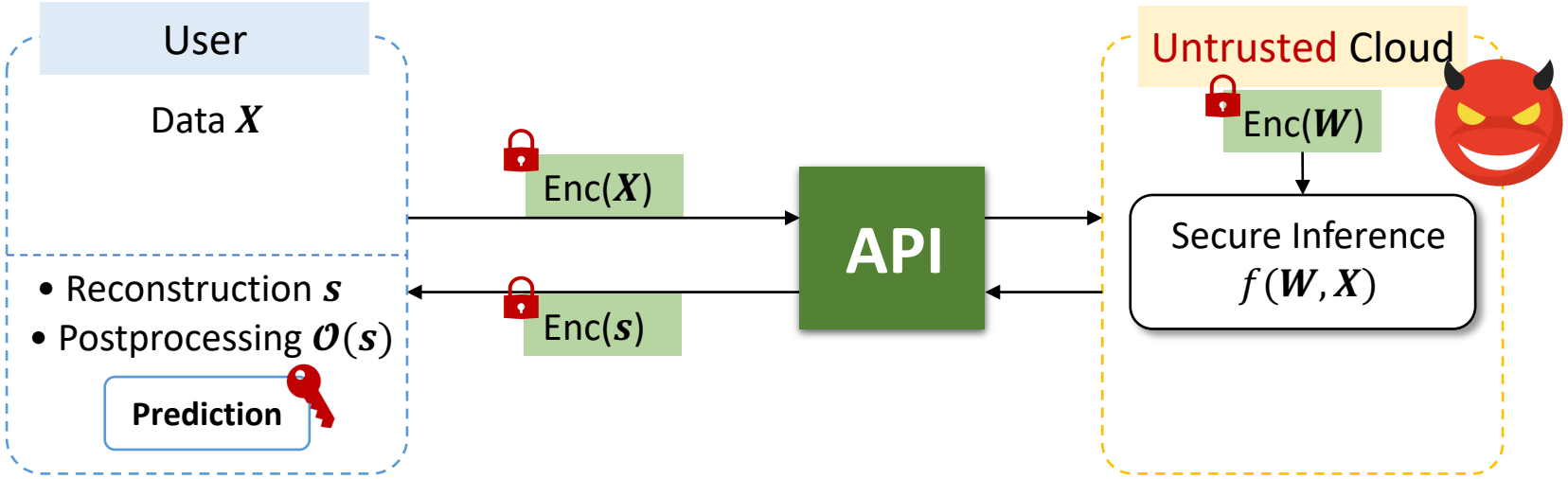
- Privacy-preserving machine learning (PPML) inference service, i.e., secure inference $f(\boldsymbol{W}, \boldsymbol{X})$
  - A user and a model owner upload encrypted data Enc($\boldsymbol{X}$) and model Enc($\boldsymbol{W}$) to the cloud
  - Opt for secure multiparty computation (MPC) [1, 2, 3] for efficiency



[1] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, “Delphi: A cryptographic inference service for neural networks,” in USENIX Security, 2020.
[2] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.
[3] Mohassel, Payman, and Peter Rindal. "ABY3: A mixed protocol framework for machine learning." Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018.

Secure inference does not protect post data information

- ***Privacy threats on post data*** - prediction API attacks $\mathcal{O}'$ exploit inference result $\boldsymbol{s}$ to learn training dataset information by querying the model [1, 2]

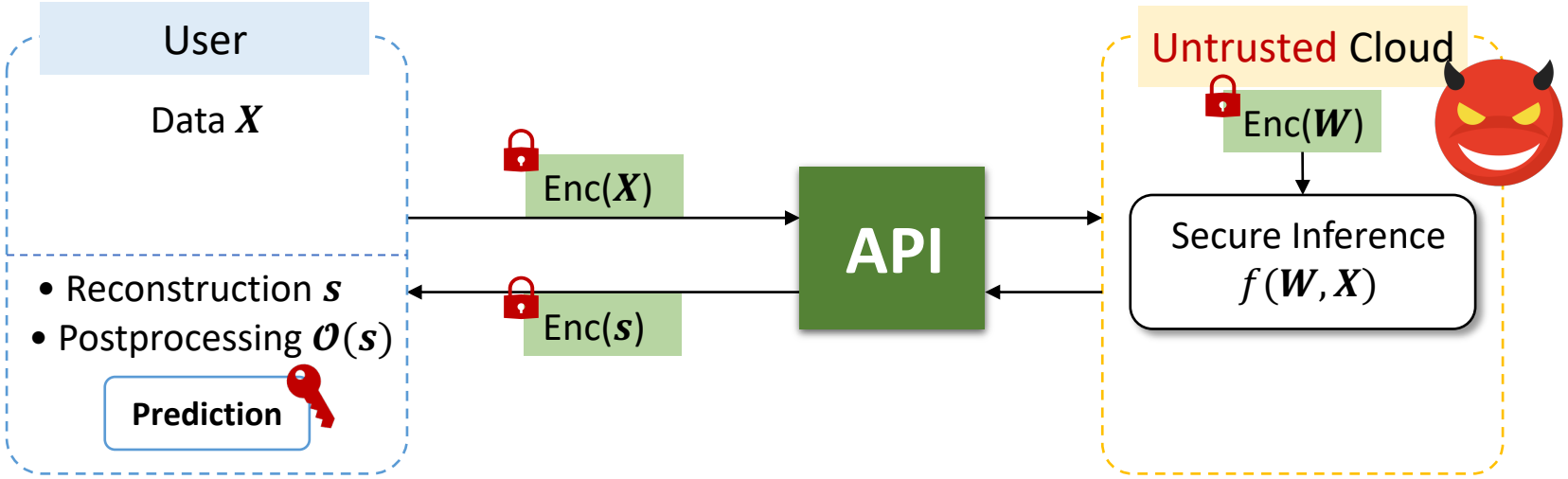


[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in S&P, 2017.
[2] H. Yu, K. Yang, T. Zhang, Y.-Y. Tsai, T.-Y. Ho, and Y. Jin, "Cloudleak: Large-scale deep learning models stealing through adversarial examples," in NDSS, 2020.

Secure inference does not protect post data information

- ***Privacy threats on post data*** - prediction API attacks $\mathcal{O}'$ exploit inference result $\boldsymbol{s}$ to learn training dataset information by querying the model [1, 2]

Output encodes knowledge of training dataset

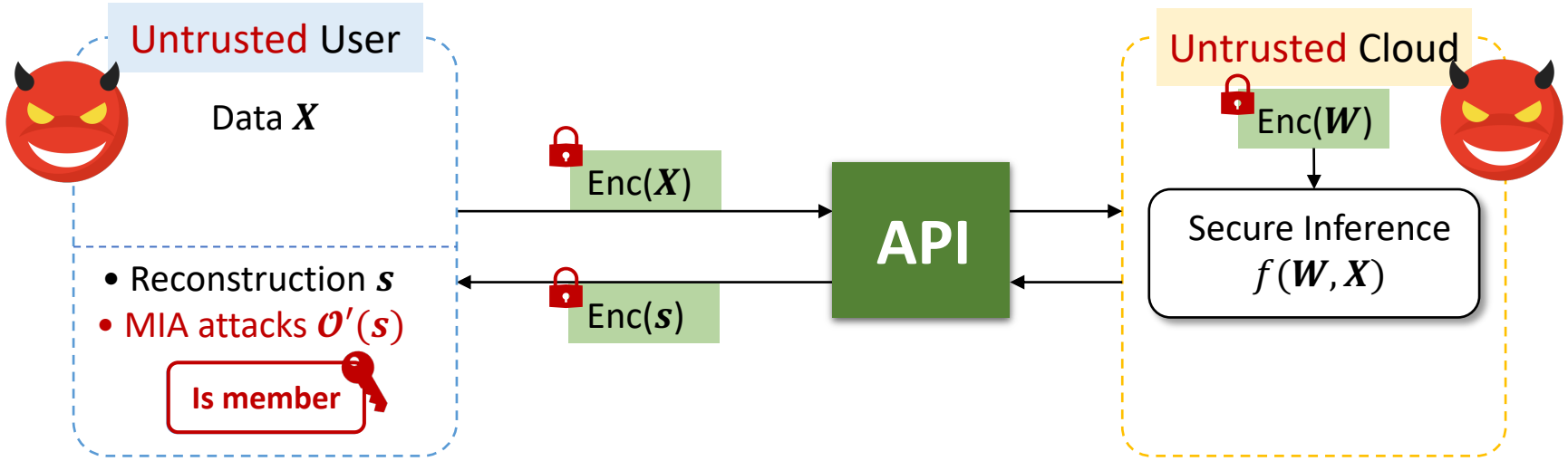


[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in S&P, 2017.
[2] H. Yu, K. Yang, T. Zhang, Y.-Y. Tsai, T.-Y. Ho, and Y. Jin, "Cloudleak: Large-scale deep learning models stealing through adversarial examples," in NDSS, 2020.

Secure inference does not protect post data information

- ***Privacy threats on post data*** - prediction API attacks $\mathcal{O}'$ exploit inference result $\boldsymbol{s}$ to learn training dataset information by querying the model [1, 2]

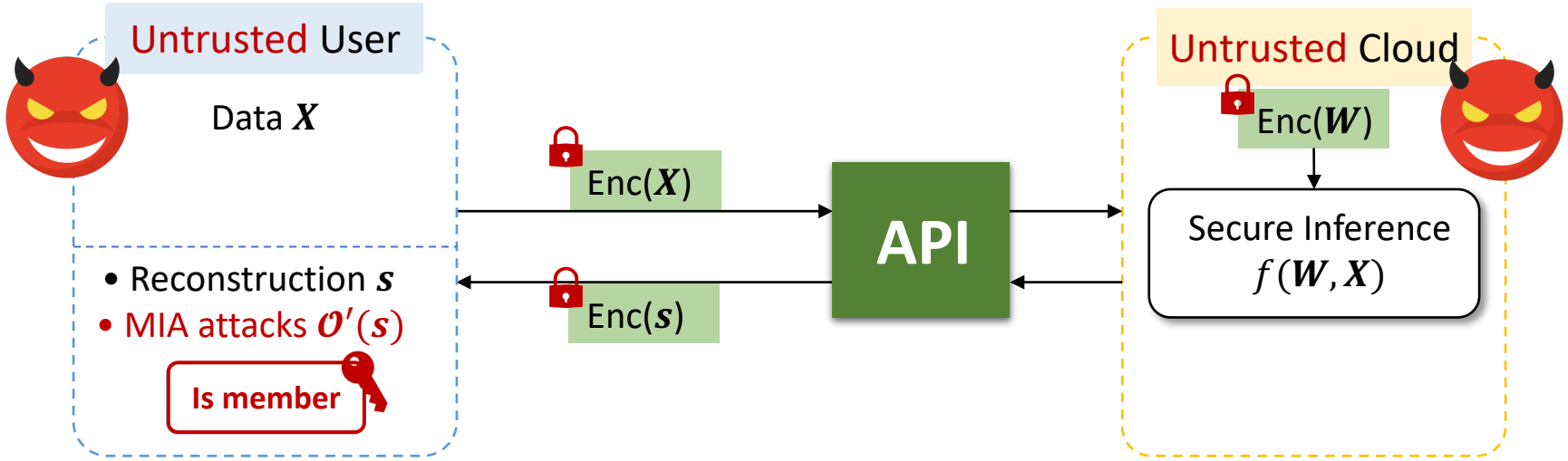
Output encodes knowledge of training dataset



[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in S&P, 2017.
[2] H. Yu, K. Yang, T. Zhang, Y.-Y. Tsai, T.-Y. Ho, and Y. Jin, "Cloudleak: Large-scale deep learning models stealing through adversarial examples," in NDSS, 2020.

Secure inference does not protect post data information

- ***Privacy threats on post data*** - Membership inference attacks (MIAs) [1] $\mathcal{O}'$ can exploit secure inference to infer whether the data $\boldsymbol{X}$ belongs to the encrypted model $\boldsymbol{W}$'s training dataset

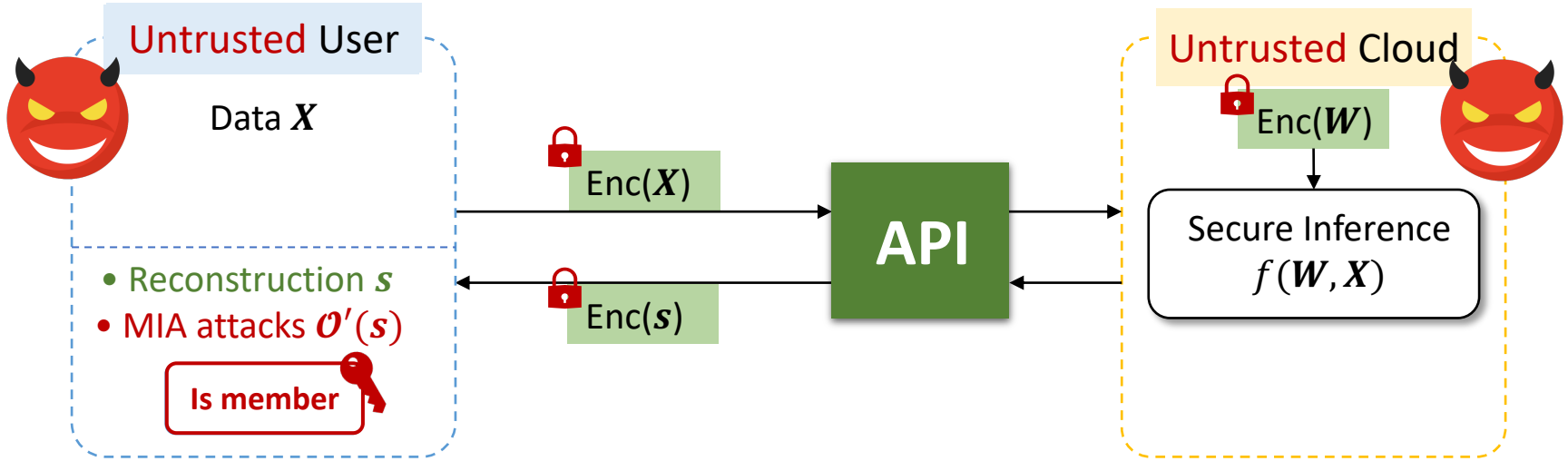


[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in S&P, 2017.

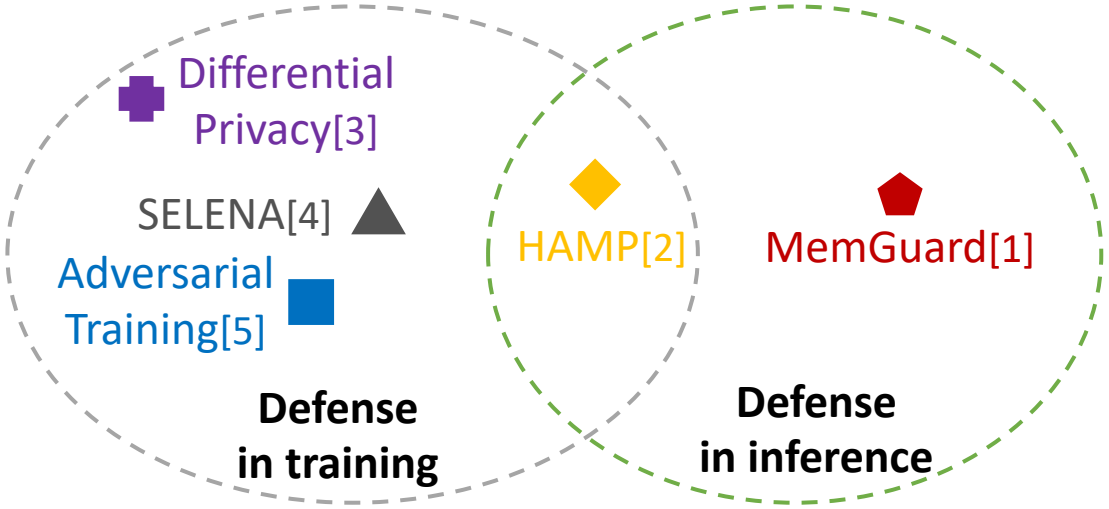
Secure inference does not protect post data information

- ***Privacy threats on post data in PPML*** - MIAs $\mathcal{O}'$ can exploit reconstructed inference result $\boldsymbol{s}$ to learn membership information by querying the encrypted model
- ***Output Privacy*** – sensitive information revealed from secure inference output (predictions)

Reconstructed output encodes knowledge of training dataset



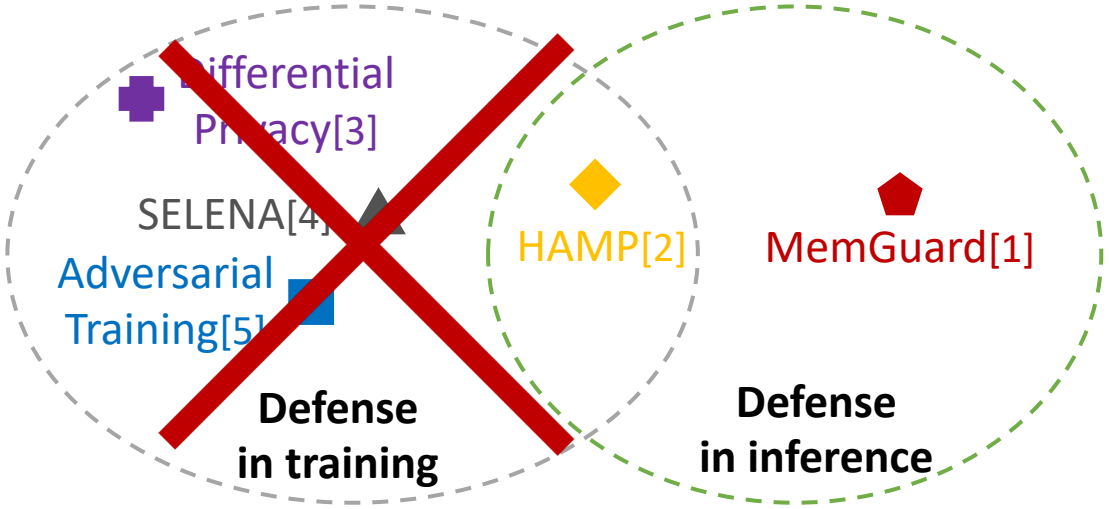
Existing MIA defenses in the plaintext domain



[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.
[3] Abadi, Martin, et al. "Deep learning with differential privacy." CCS 2016.
[4] Tang, Xinyu, et al. "Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture." USENIX Security 2022.
[5] Nasr, Milad, Reza Shokri, and Amir Houmansadr. "Machine learning with membership privacy using adversarial regularization." CCS 2018.

Existing MIA defenses in the plaintext domain

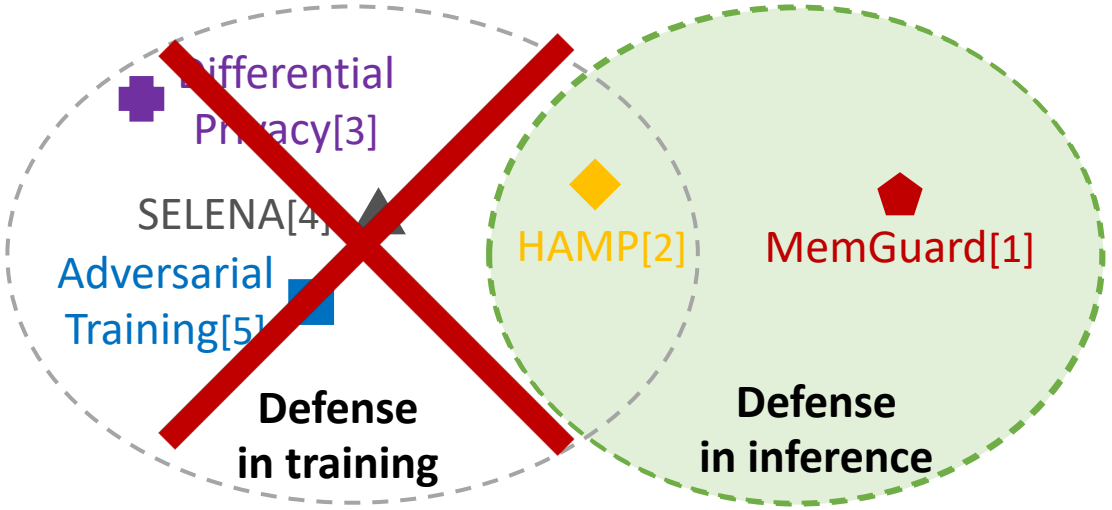
- Training time defenses use specific training approaches
  - However, they inherently reduce prediction accuracy, e.g., differentially private training [3]



[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.
[3] Abadi, Martin, et al. "Deep learning with differential privacy." CCS 2016.
[4] Tang, Xinyu, et al. "Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture." USENIX Security 2022.
[5] Nasr, Milad, Reza Shokri, and Amir Houmansadr. "Machine learning with membership privacy using adversarial regularization." CCS 2018.

Existing MIA defenses in the plaintext domain

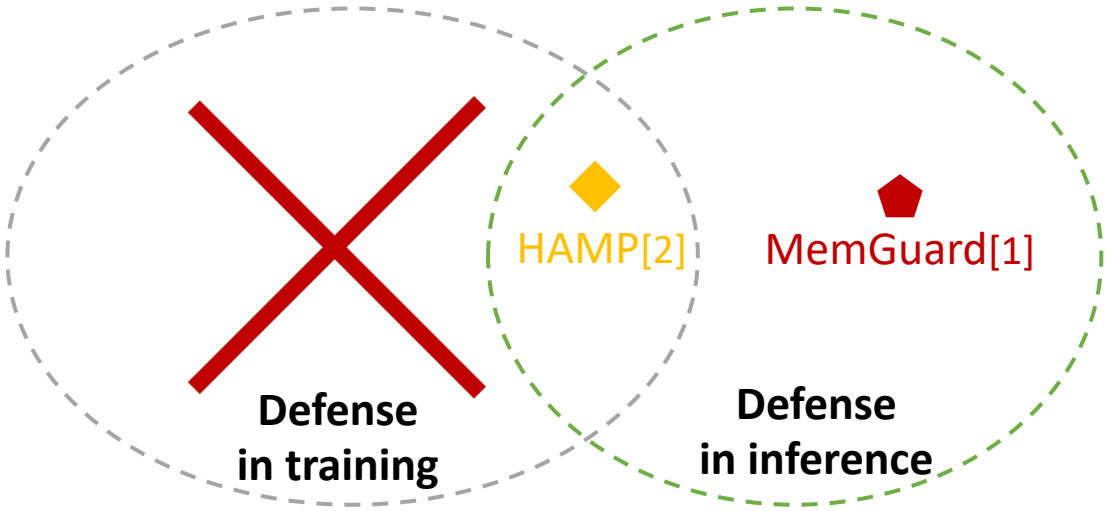
- Training time defenses use specific training approaches
  - However, they inherently reduce prediction accuracy, e.g., differentially private training [3]
  - Prefer inference time defense without accuracy loss



[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.
[3] Abadi, Martin, et al. "Deep learning with differential privacy." CCS 2016.
[4] Tang, Xinyu, et al. "Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture." USENIX Security 2022.
[5] Nasr, Milad, Reza Shokri, and Amir Houmansadr. "Machine learning with membership privacy using adversarial regularization." CCS 2018.

Existing MIA defenses in the plaintext domain

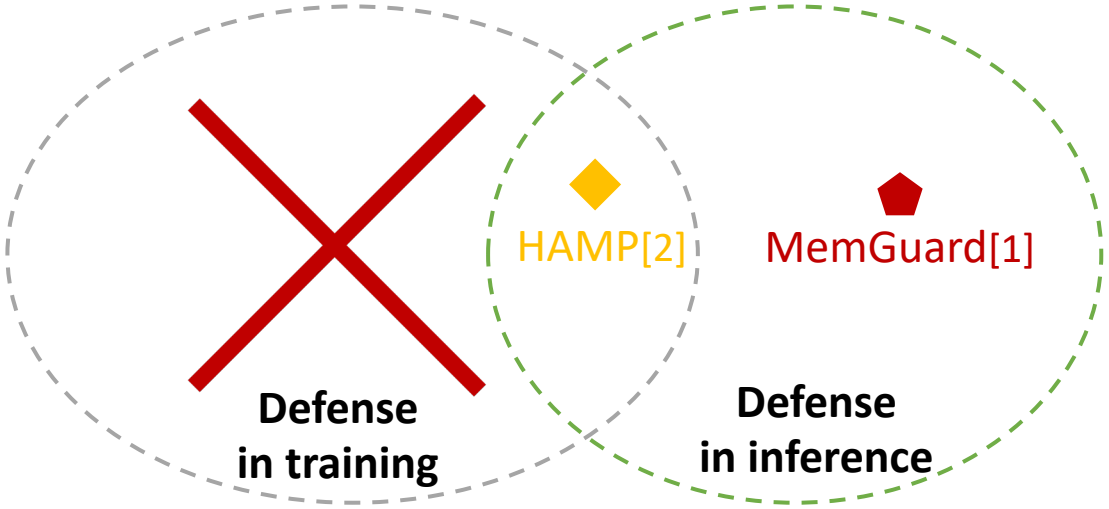
- Inference time defenses inject carefully crafted noises to perturb the inference output



[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

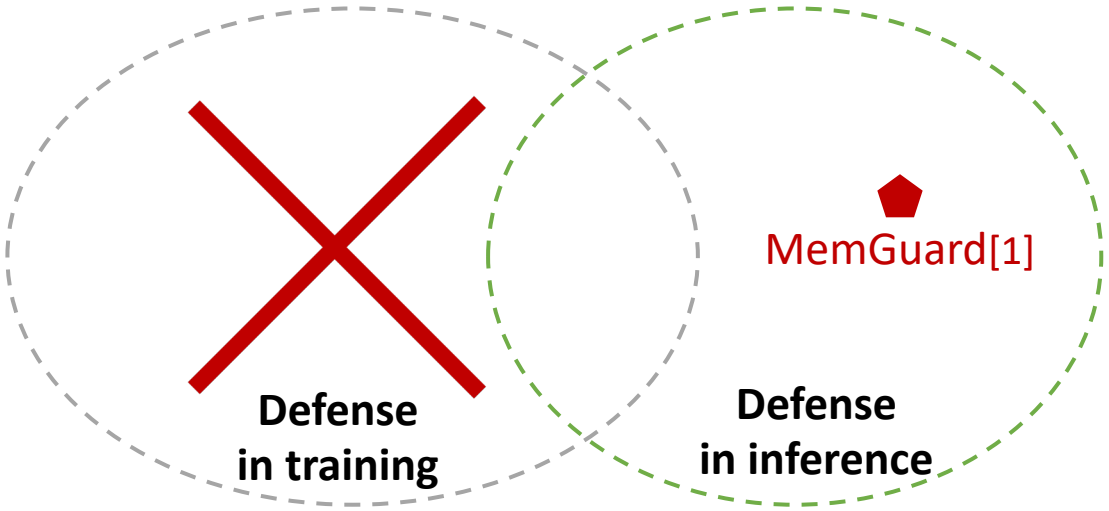
- Inference time defenses inject carefully crafted noises to perturb the inference output
  - HAMP [2] needs to apply defense in both training and inference, thus needs costly re-training



[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

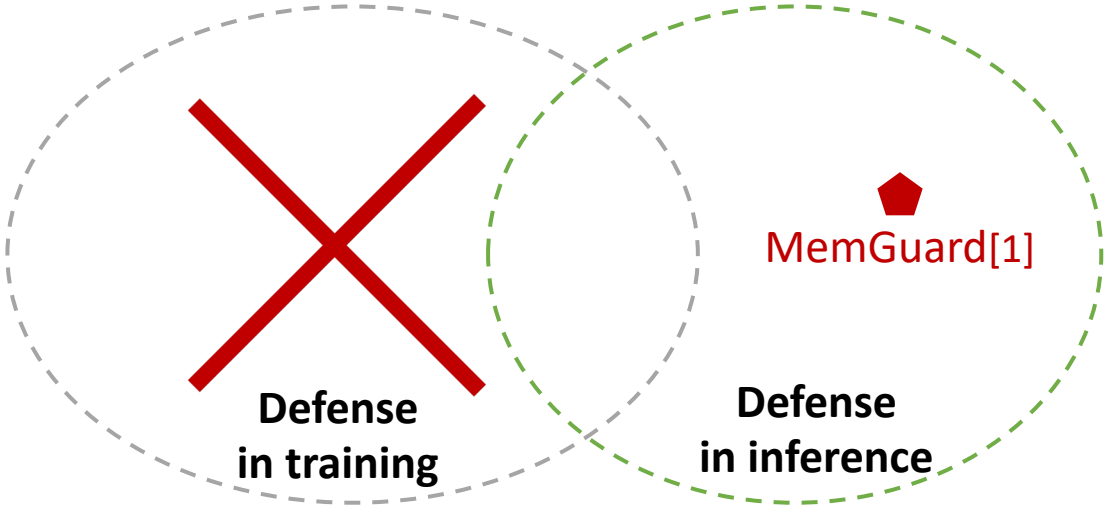
- Inference time defenses inject carefully crafted noises to perturb the inference output
  - HAMP [2] needs to apply defense in both training and inference, thus needs costly re-training



[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

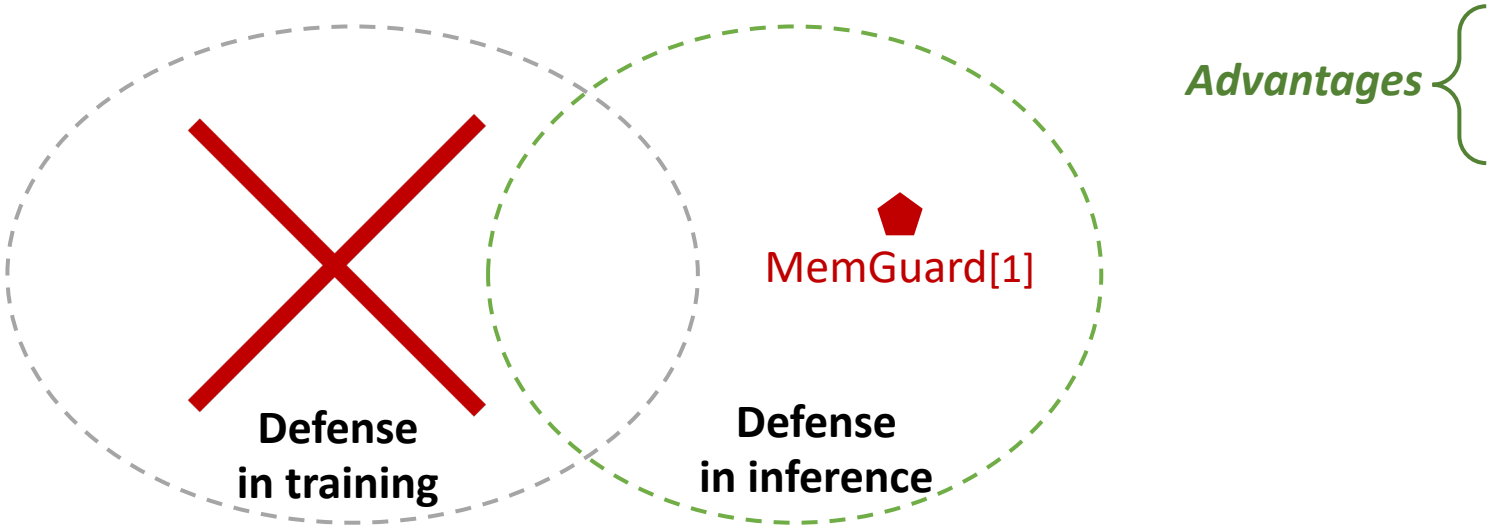
- Inference time defenses inject carefully crafted noises to perturb the inference output
  - HAMP [2] needs to apply defense in both training and inference, thus needs costly re-training
  - MemGuard [1] is chosen as our starting point



[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - HAMP [2] needs to apply defense in both training and inference, thus needs costly re-training
  - MemGuard [1] is chosen as our starting point

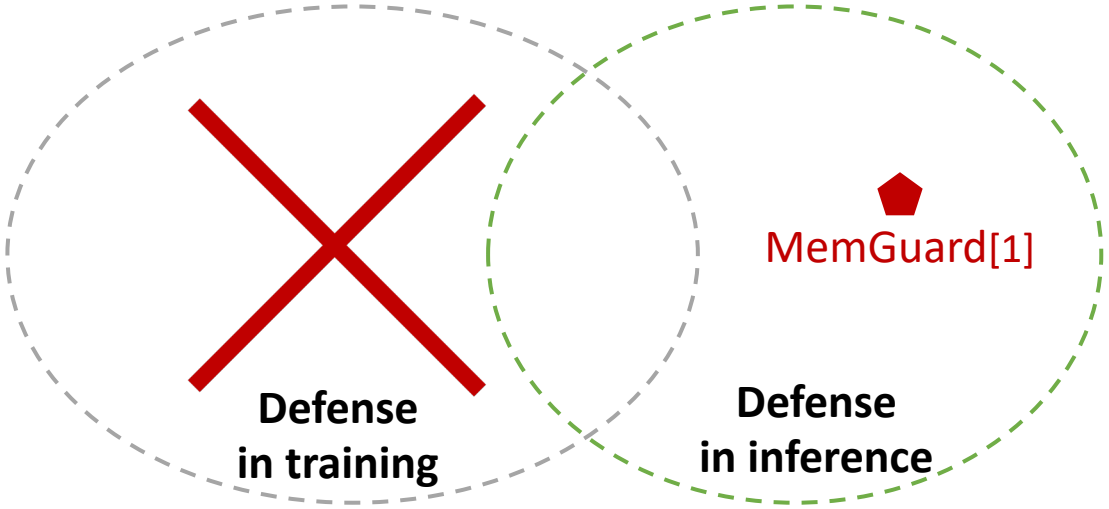


Advantages

[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - HAMP [2] needs to apply defense in both training and inference, thus needs costly re-training
  - MemGuard [1] is chosen as our starting point



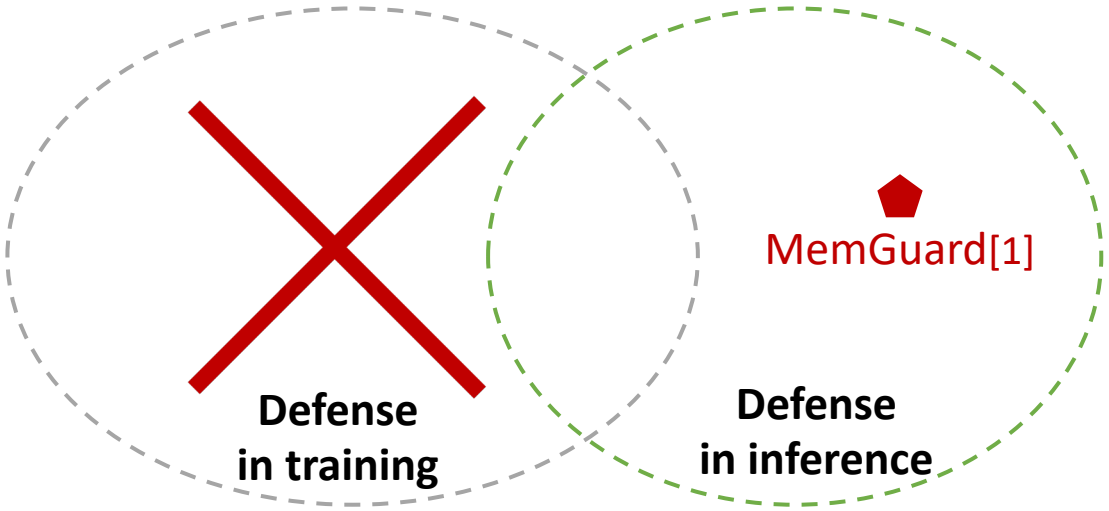
Advantages

- Preserve prediction accuracy

[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - HAMP [2] needs to apply defense in both training and inference, thus needs costly re-training
  - MemGuard [1] is chosen as our starting point



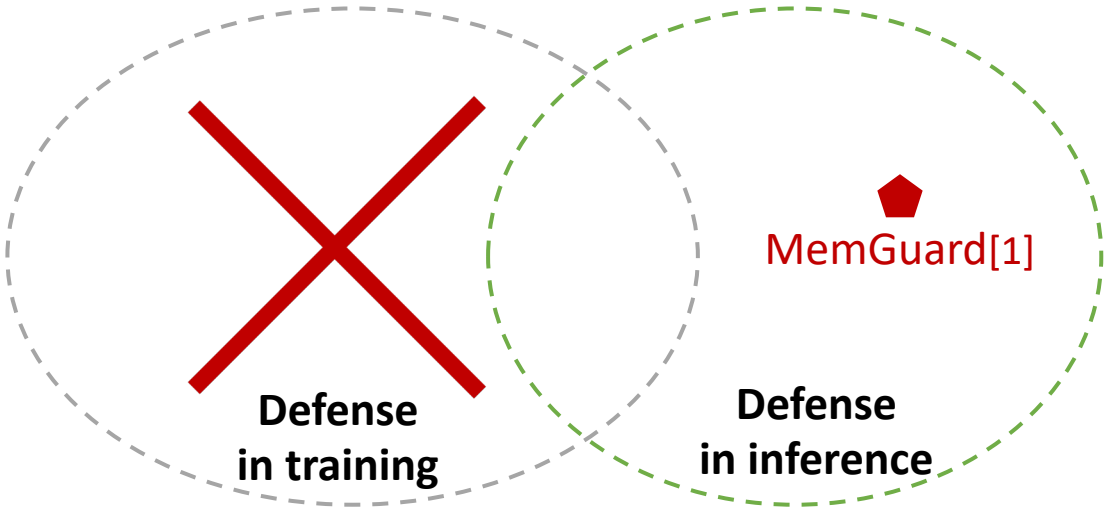
Advantages

- Preserve prediction accuracy
- No need for costly re-training

[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - HAMP [2] needs to apply defense in both training and inference, thus needs costly re-training
  - MemGuard [1] is chosen as our starting point



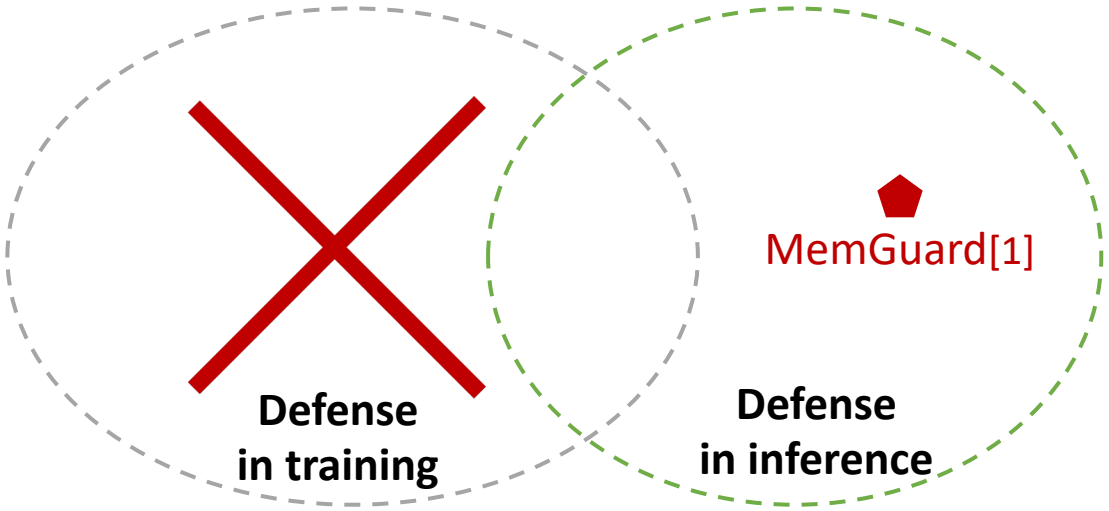
Advantages

- Preserve prediction accuracy
- No need for costly re-training
- Do not disrupt MLaaS pipeline

[1] Jia, Jinyuan, et al. "Memguard: Defending against black-box membership inference attacks via adversarial examples." CCS 2019.
[2] Chen, Zitao, and Karthik Pattabiraman. "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction." NDSS. 2023.

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output

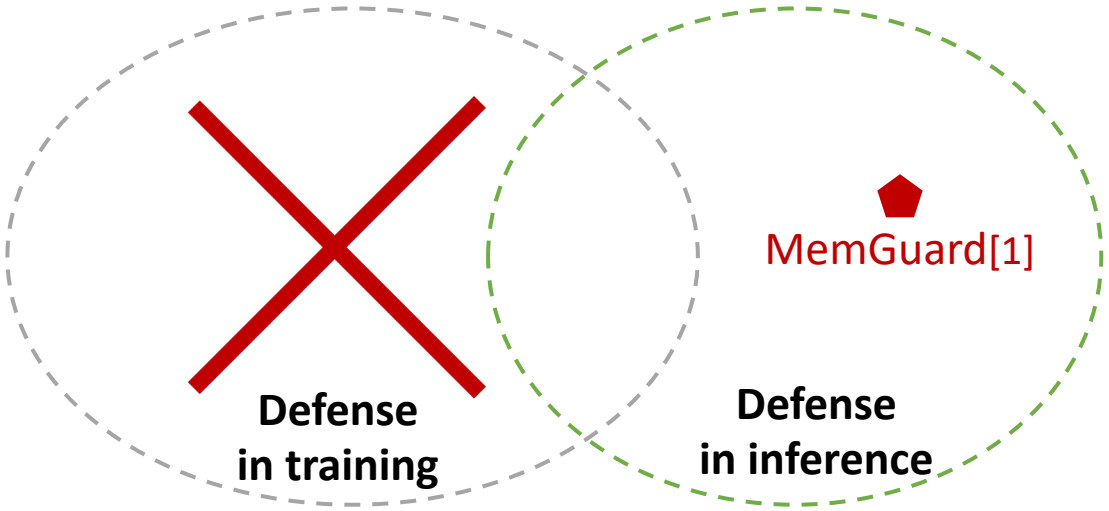


Advantages

- Preserve prediction accuracy
- No need for costly re-training
- Do not disrupt MLaaS pipeline

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - However, it cannot directly be adopted in secure inference

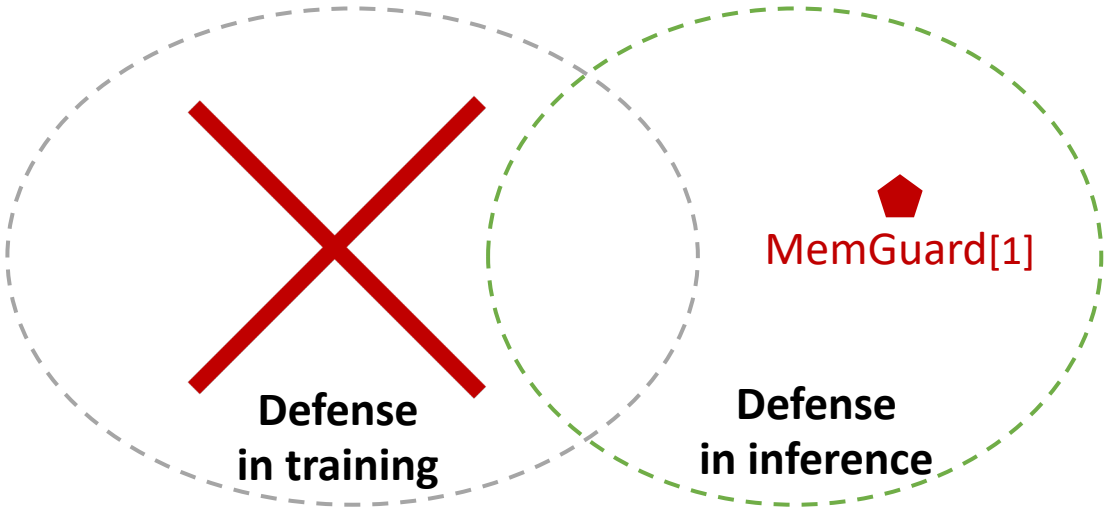


Advantages

- Preserve prediction accuracy
- No need for costly re-training
- Do not disrupt MLaaS pipeline

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - However, it cannot directly be adopted in secure inference



Advantages

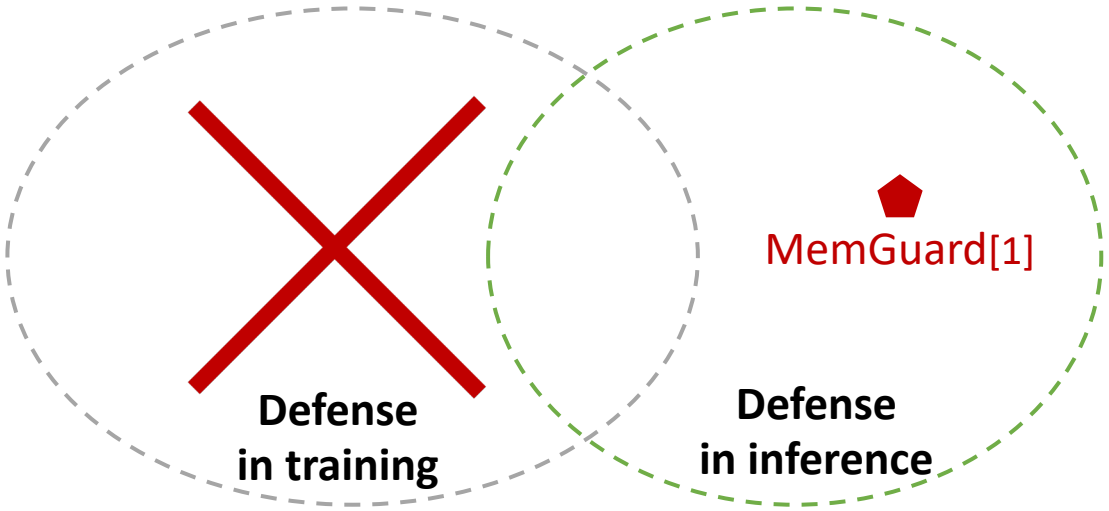
- Preserve prediction accuracy
- No need for costly re-training
- Do not disrupt MLaaS pipeline

Obstacles

- Process with cleartext data

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - However, it cannot directly be adopted in secure inference



Advantages

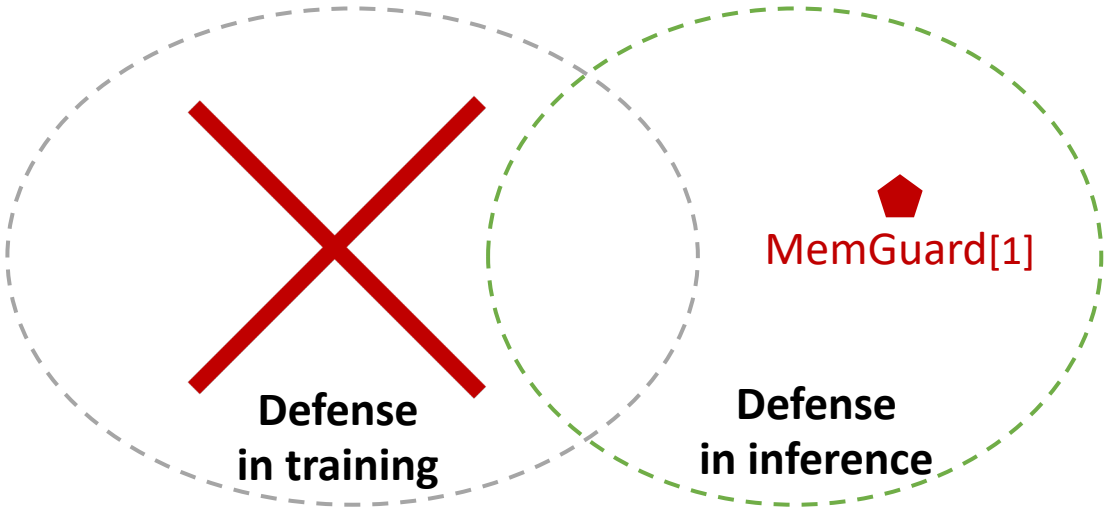
- Preserve prediction accuracy
- No need for costly re-training
- Do not disrupt MLaaS pipeline

Obstacles

- Process with cleartext data
- Threat model differs from PPML

Existing MIA defenses in the plaintext domain

- Inference time defenses inject carefully crafted noises to perturb the inference output
  - However, it cannot directly be adopted in secure inference



Advantages

- Preserve prediction accuracy
- No need for costly re-training
- Do not disrupt MLaaS pipeline

Obstacles

- Process with cleartext data
- Threat model differs from PPML
- Complex and recursive algorithm

Our research

Building **SIGuard** framework

Secure **I**nference **Guard**

guarding output privacy

integrated into versatile MPC-based secure inference

Our philosophy

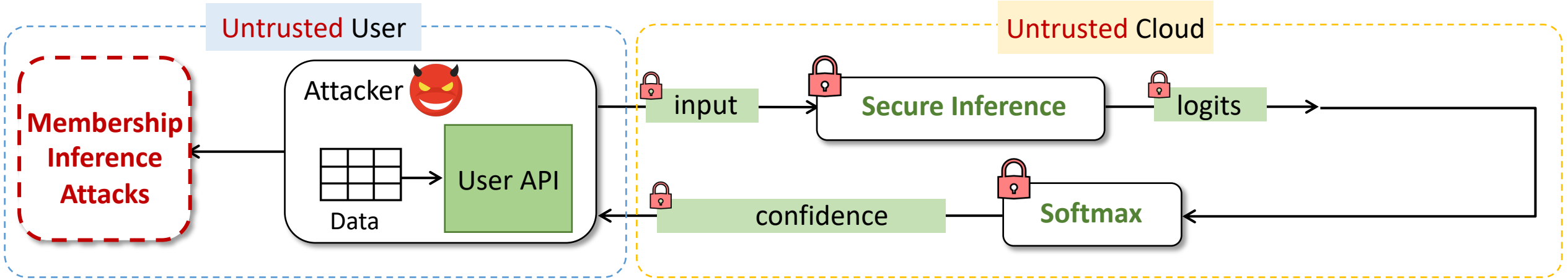
Harnessing insights of MPC techniques and machine learning

stringent and provable security guarantees

preserving model prediction accuracy

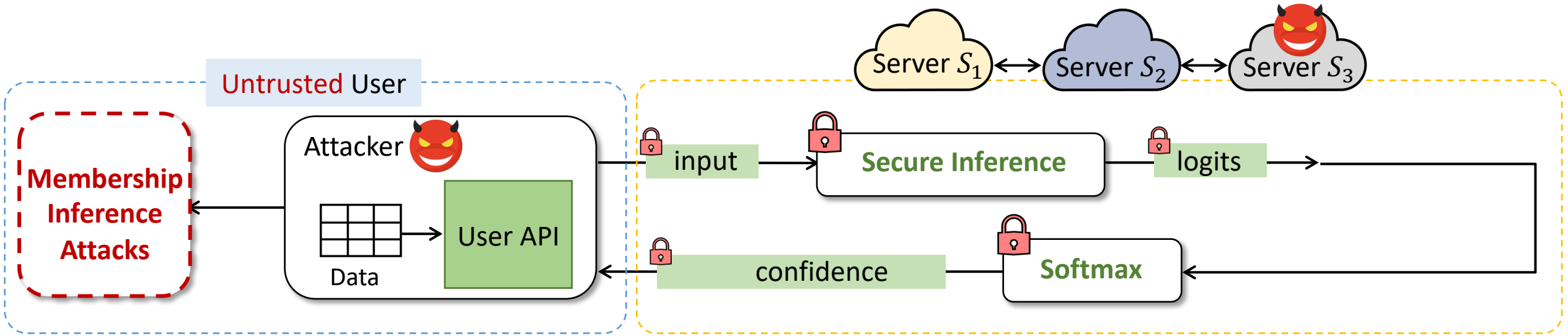
Our approach

- 3-party MPC techniques to achieve privacy guarantees
- SIGuard protocol plugs into secure inference



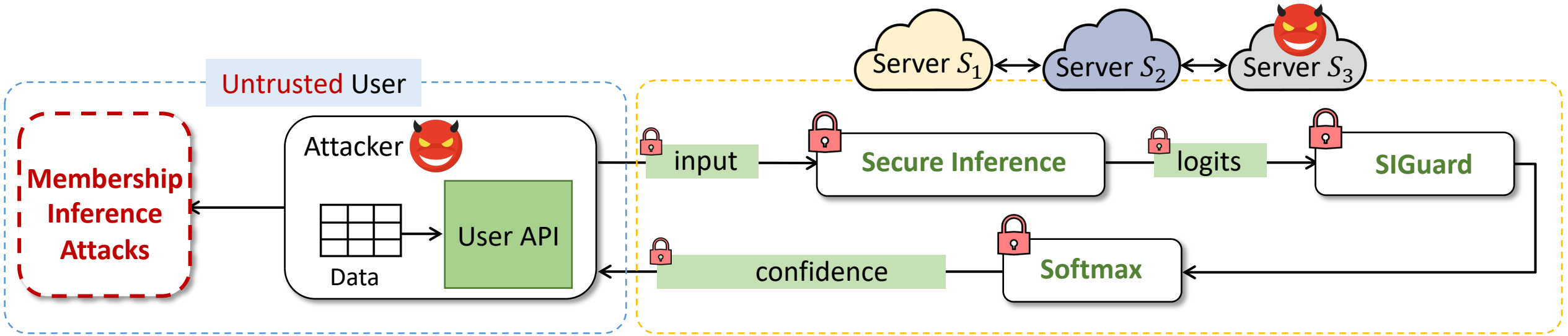
Our approach

- 3-party MPC techniques to achieve privacy guarantees
- SIGuard protocol plugs into secure inference



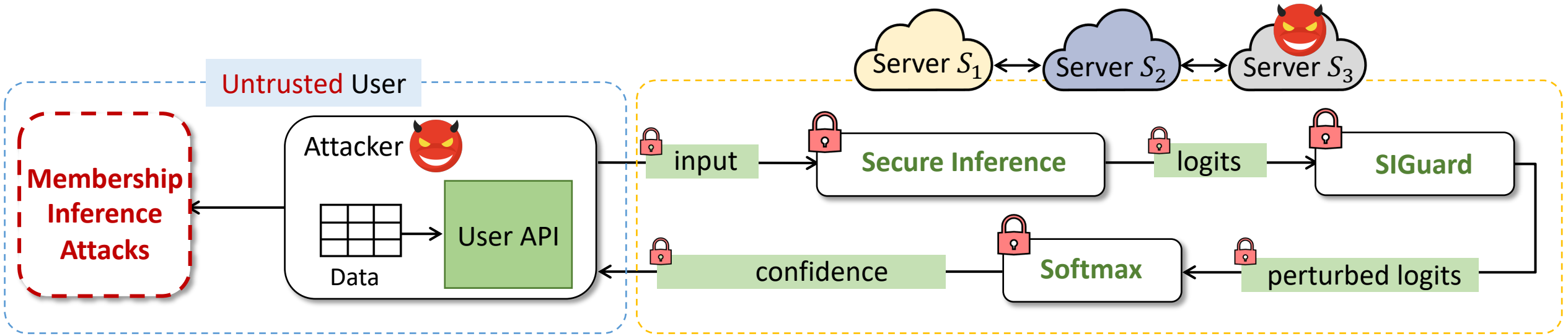
Our approach

- 3-party MPC techniques to achieve privacy guarantees
- SIGuard protocol plugs into secure inference



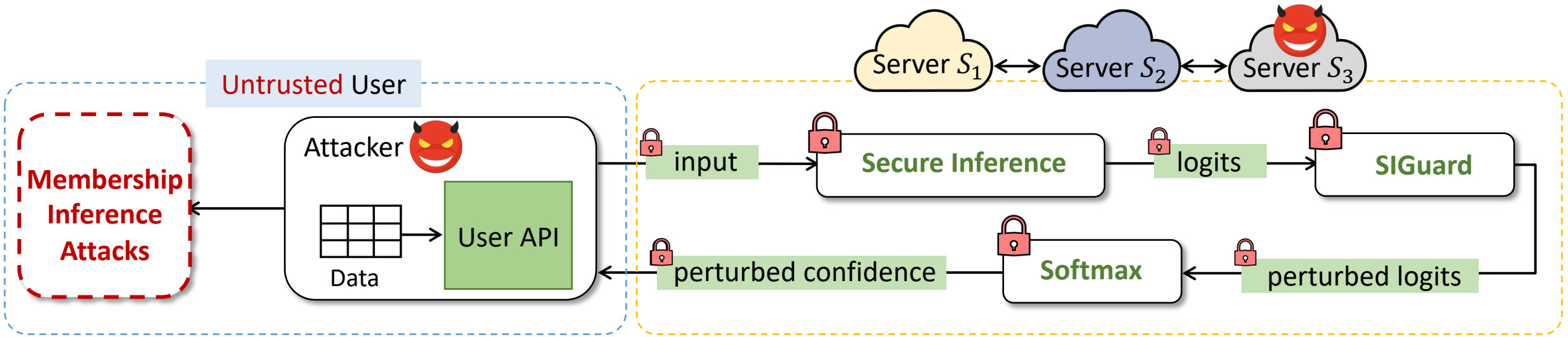
Our approach

- 3-party MPC techniques to achieve privacy guarantees
- SIGuard protocol plugs into secure inference



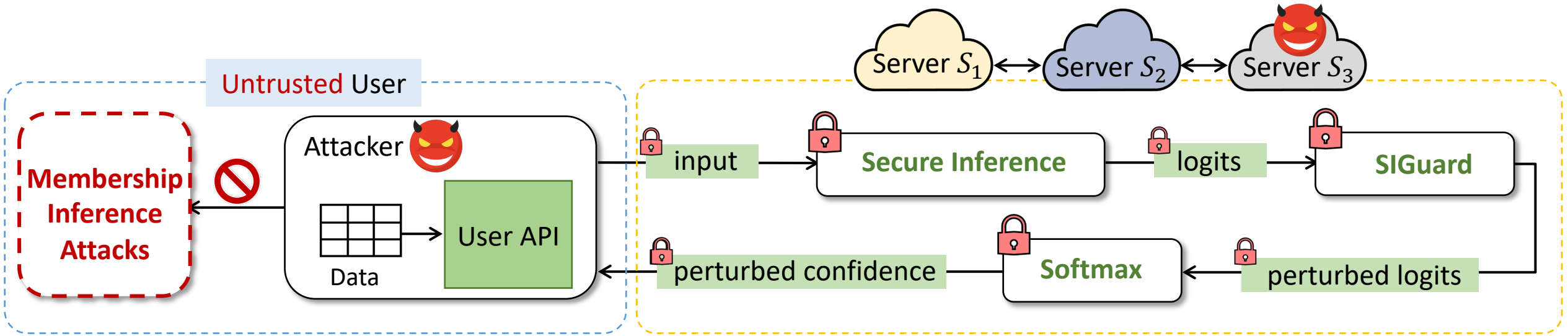
Our approach

- 3-party MPC techniques to achieve privacy guarantees
- SIGuard protocol plugs into secure inference



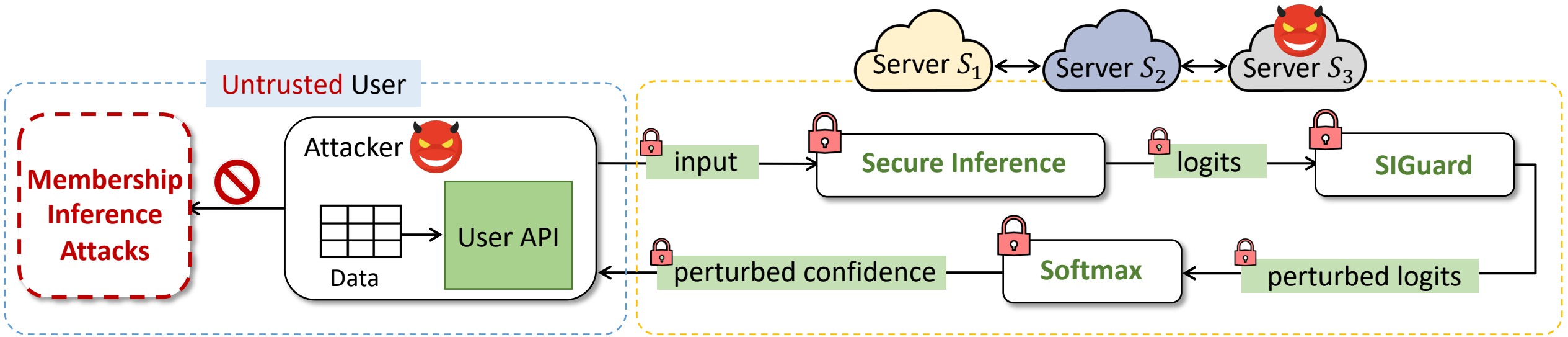
Our approach

- 3-party MPC techniques to achieve privacy guarantees
- SIGuard protocol plugs into secure inference



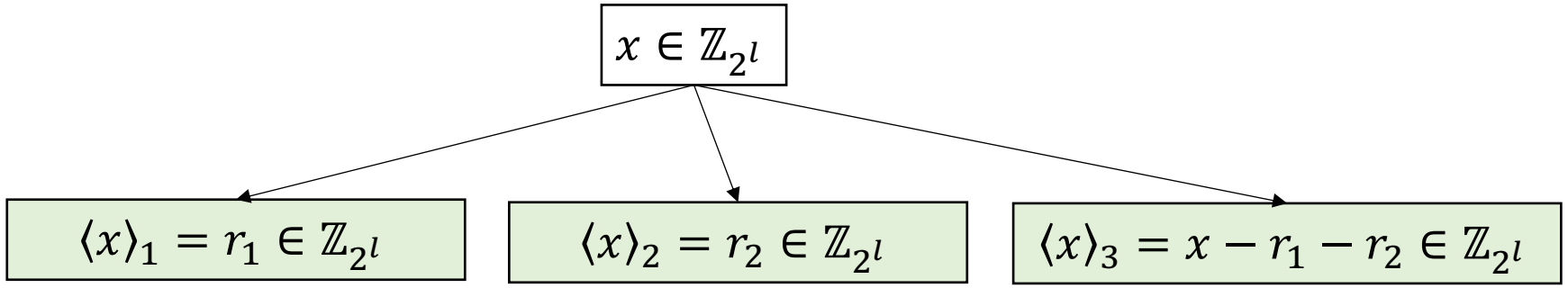
Our approach

- Independent corruption
- Collusion – broaden attack surface
  - MIA adversary corrupting user, colludes with secure inference adversary corrupting one cloud server



MPC techniques - Replicate secret sharing (RSS)

- Replicate secret sharing encrypts private data as secret shares



Party P_1

$(\langle x \rangle_1, \langle x \rangle_2)$

Party P_2

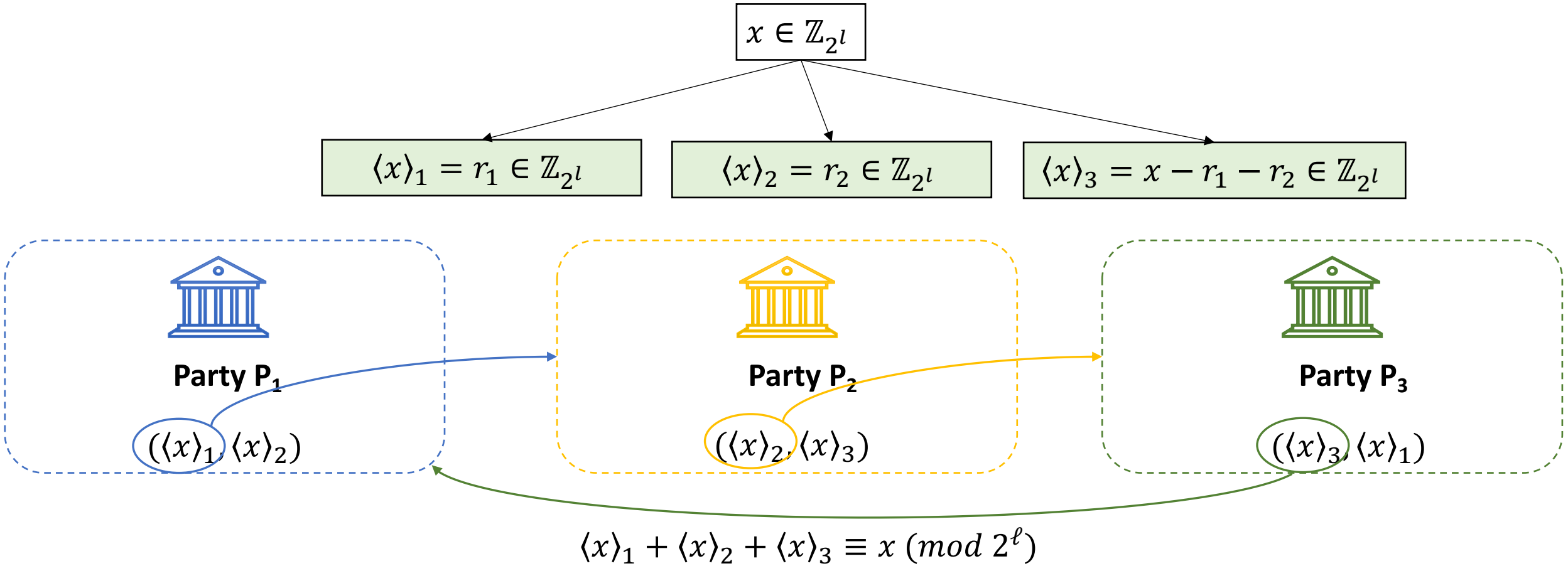
$(\langle x \rangle_2, \langle x \rangle_3)$

Party P_3

$(\langle x \rangle_3, \langle x \rangle_1)$

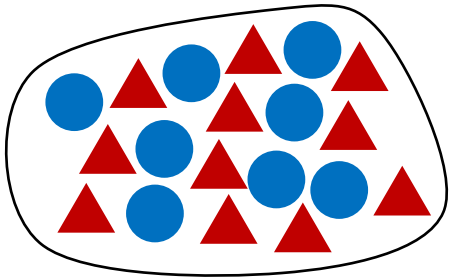
MPC techniques - Replicate secret sharing (RSS)

- Replicate secret sharing encrypts private data as secret shares



Machine learning - MIAs

- The MIA adversary trains a membership (binary) classifier $f(\vec{s}) = \begin{cases} 1 & x \in M\text{'s training dataset} \\ 0 & x \notin M\text{'s training dataset} \end{cases}$

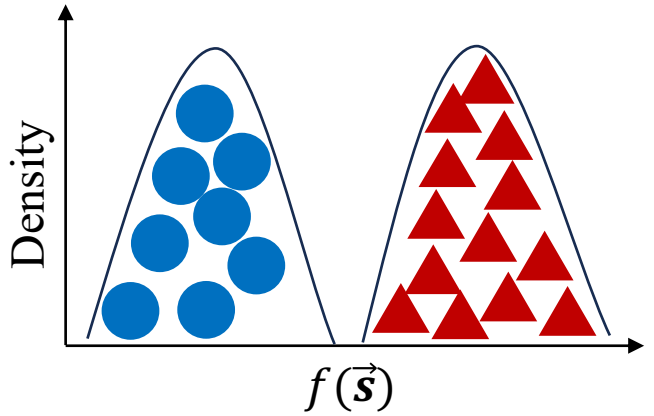


Membership dataset D

f

Ideal attack

100% membership classify accuracy!



Ground-truth

Member prediction $\vec{s}$

Non-member prediction $\vec{s}$

Machine learning – MemGuard

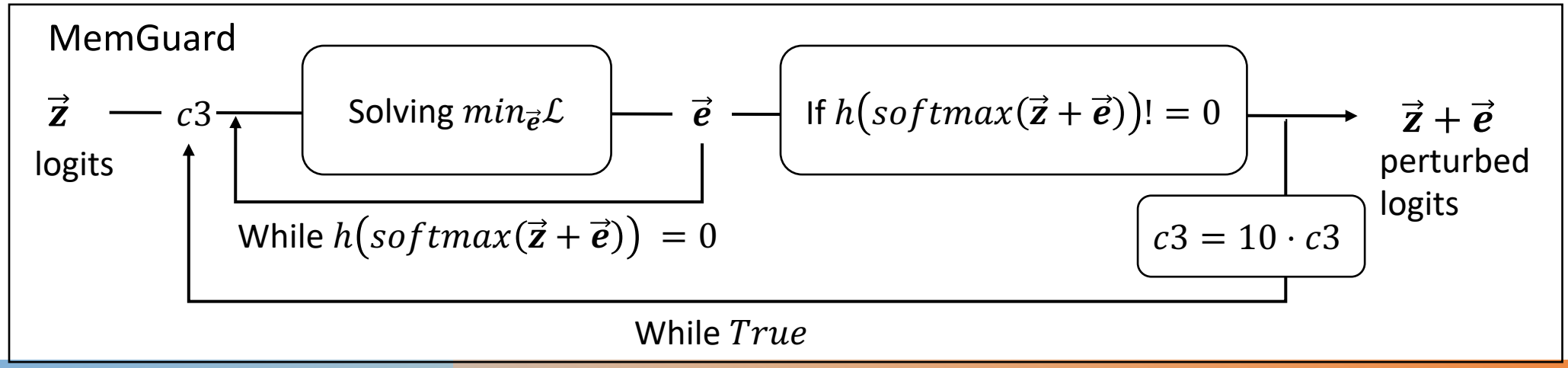
- Insight: MIAs classifier is vulnerable to *adversarial examples*, fool MIAs by perturbing confidence to adversarial example

Machine learning – MemGuard

- Insight: MIAs classifier is vulnerable to *adversarial examples*, fool MIAs by perturbing confidence to adversarial example
- Approach: adding a noise vector $\vec{e}$ to perturb logits $\vec{z}$, while keeping final prediction unchanged $\vec{s}' = \vec{s}$

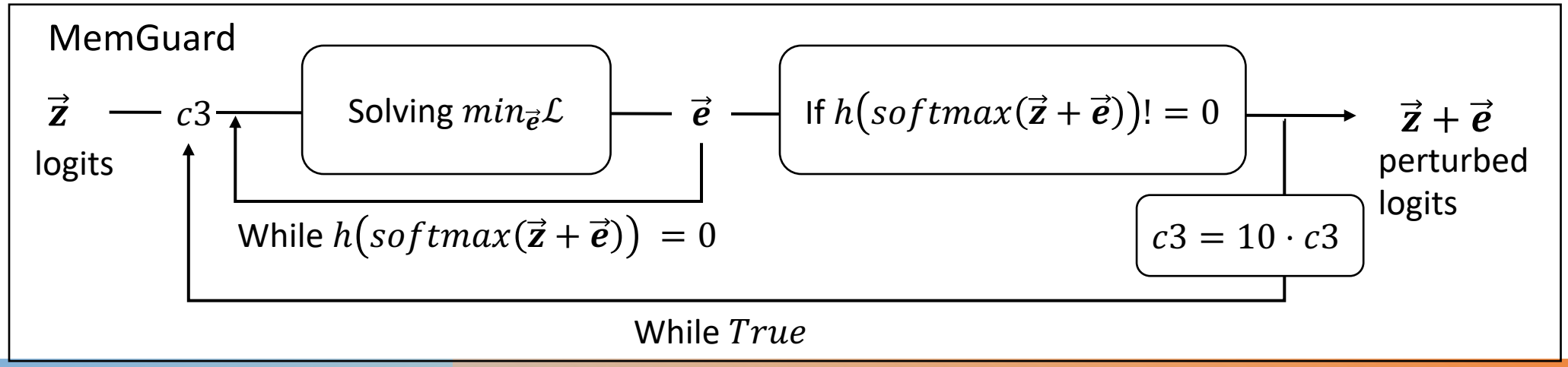
Machine learning – MemGuard

- Insight: MIAs classifier is vulnerable to *adversarial examples*, fool MIAs by perturbing confidence to adversarial example
- Approach: adding a noise vector $\vec{e}$ to perturb logits $\vec{z}$, while keeping final prediction unchanged $\vec{s}' = \vec{s}$
- Workflow:
  - Iteratively optimizes the noise vector $\vec{e}$ by minimizing the loss $\mathcal{L}$ ($c3$ controls optimization pace)



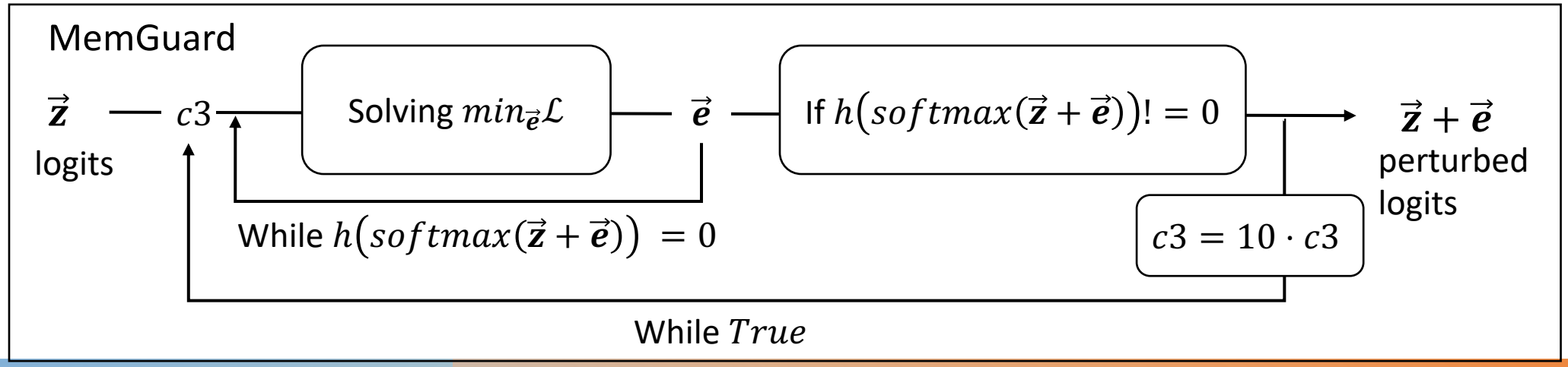
Machine learning – MemGuard

- Insight: MIAs classifier is vulnerable to *adversarial examples*, fool MIAs by perturbing confidence to adversarial example
- Approach: adding a noise vector $\vec{e}$ to perturb logits $\vec{z}$, while keeping final prediction unchanged $\vec{s}' = \vec{s}$
- Workflow:
  - Iteratively optimizes the noise vector $\vec{e}$ by minimizing the loss $\mathcal{L}$ ($c3$ controls optimization pace)
  - Generate perturbed prediction (confidence) $\vec{s}' = softmax(\vec{z} + \vec{e})$



Machine learning – MemGuard

- Insight: MIAs classifier is vulnerable to *adversarial examples*, fool MIAs by perturbing confidence to adversarial example
- Approach: adding a noise vector $\vec{e}$ to perturb logits $\vec{z}$, while keeping final prediction unchanged $\vec{s}' = \vec{s}$
- Workflow:
  - Iteratively optimizes the noise vector $\vec{e}$ by minimizing the loss $\mathcal{L}$ ($c3$ controls optimization pace)
  - Generate perturbed prediction (confidence) $\vec{s}' = softmax(\vec{z} + \vec{e})$
  - Terminate when $h(\vec{s}') = \mathbf{0}$, i.e., classifier h couldn't determine the membership of perturbed prediction and noise $\vec{e}$ can't be further minimized

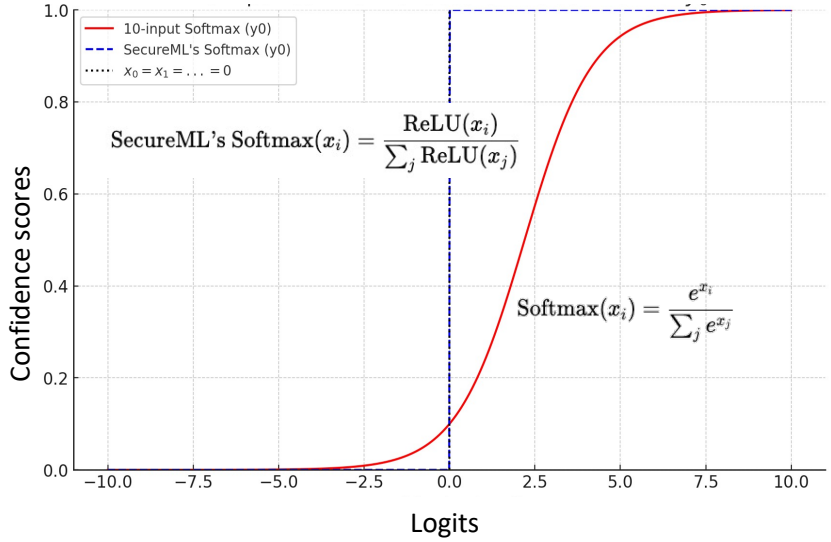


Observation 1: Perturbations from MPC nonlinear approximations

[1] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.

Observation 1: Perturbations from MPC nonlinear approximations

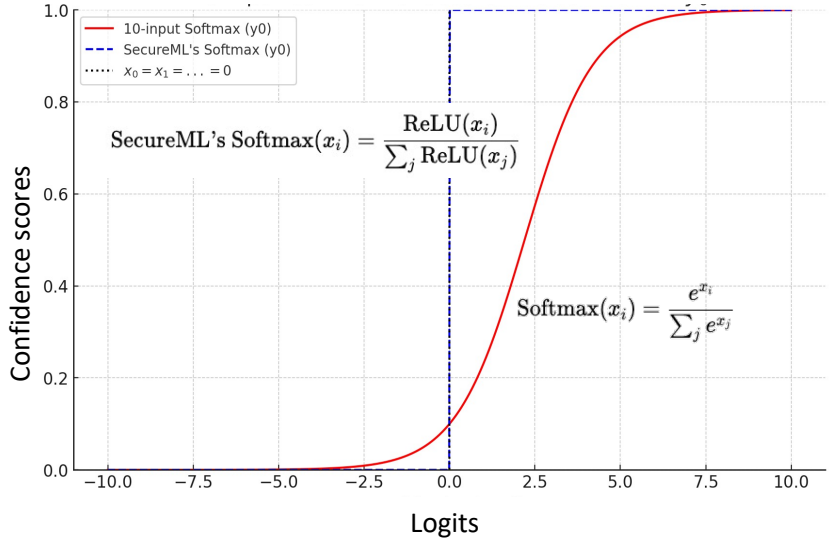
- MPC requires to approximate non-linear functions, e.g., softmax (division, exp)
  - E.g., SecureML [1] approximates softmax using ReLU



[1] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.

Observation 1: Perturbations from MPC nonlinear approximations

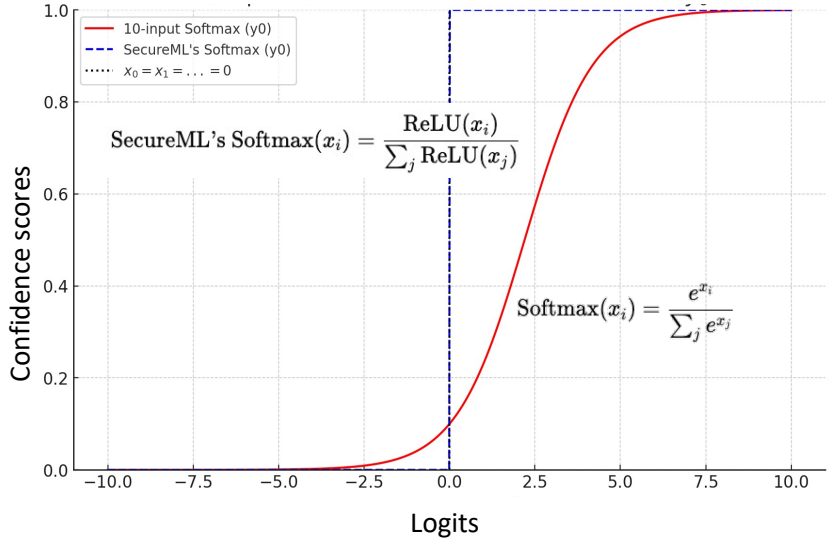
- MPC requires to approximate non-linear functions, e.g., softmax (division, exp)
  - E.g., SecureML [1] approximates softmax using ReLU
- Approximated softmax in secure inference perturbs confidence scores



[1] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.

Observation 1: Perturbations from MPC nonlinear approximations

- MPC requires to approximate non-linear functions, e.g., softmax (division, exp)
  - E.g., SecureML [1] approximates softmax using ReLU
- Approximated softmax in secure inference perturbs confidence scores

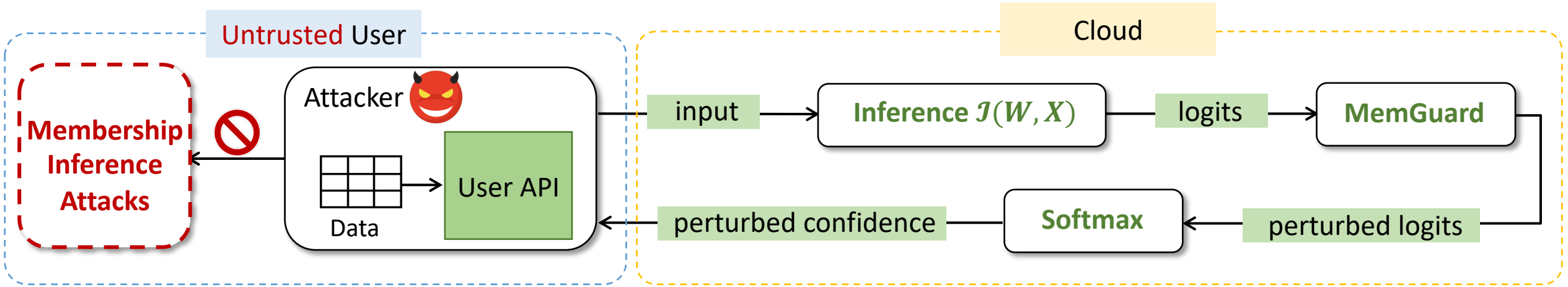


***RQ1**: Whether MIAs can still exploit secure inference with perturbed predictions?*

[1] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.

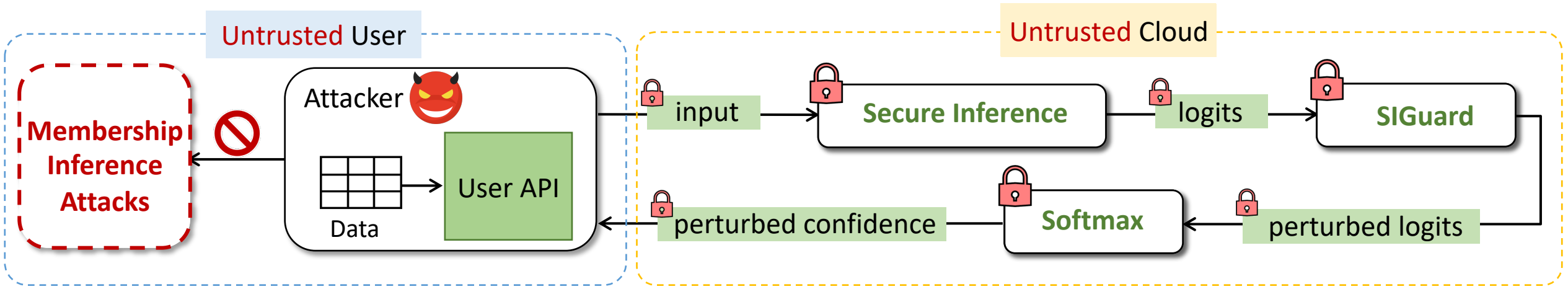
Observation 2: Secure post inference defense just like MemGuard

- Inspired by MemGuard [1], SIGuard's protocol injects carefully crafted perturbations into the encrypted predictions without harming the inference accuracy
- Takes encrypted logits, and finally returns encrypted perturbed confidence to the user



Observation 2: Secure post inference defense just like MemGuard

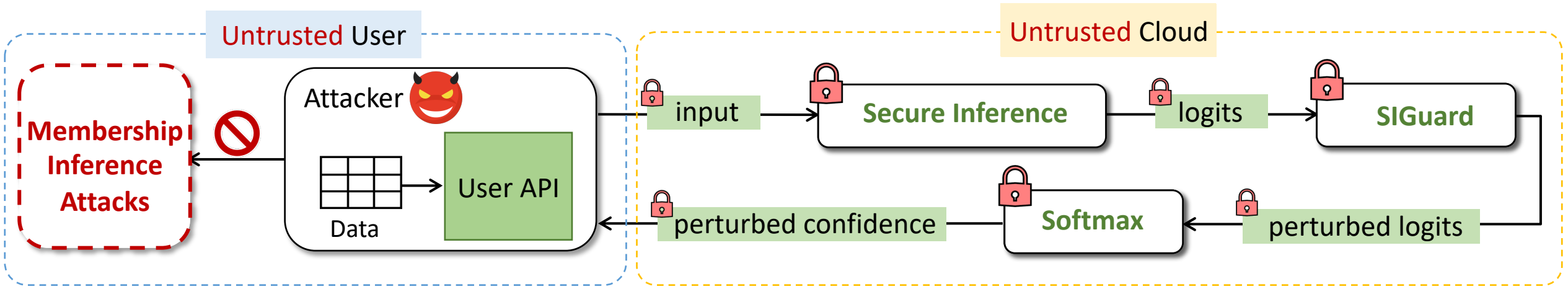
- Inspired by MemGuard [1], SIGuard's protocol injects carefully crafted perturbations into the encrypted predictions without harming the inference accuracy
- Takes encrypted logits, and finally returns encrypted perturbed confidence to the user



Observation 2: Secure post inference defense just like MemGuard

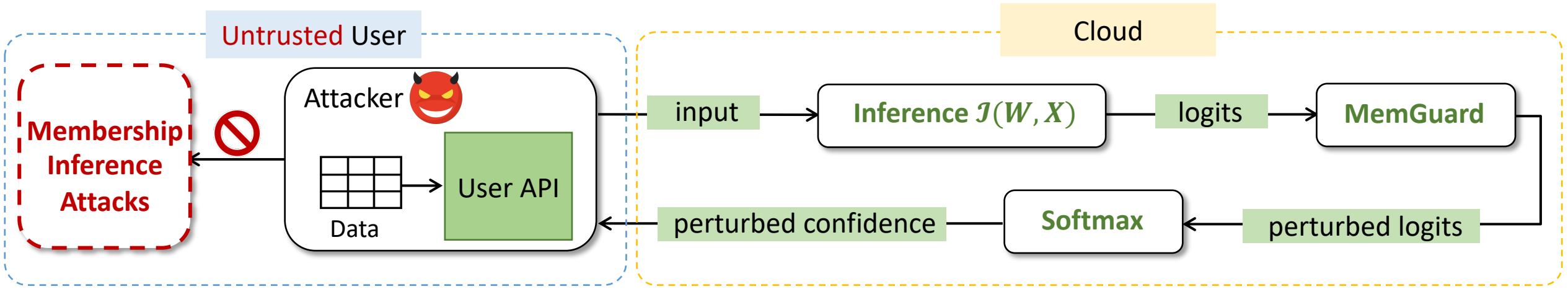
- Inspired by MemGuard [1], SIGuard's protocol injects carefully crafted perturbations into the encrypted predictions without harming the inference accuracy
- Takes encrypted logits, and finally returns encrypted perturbed confidence to the user

***RQ2**: How to efficiently realize SIGuard with MPC?*



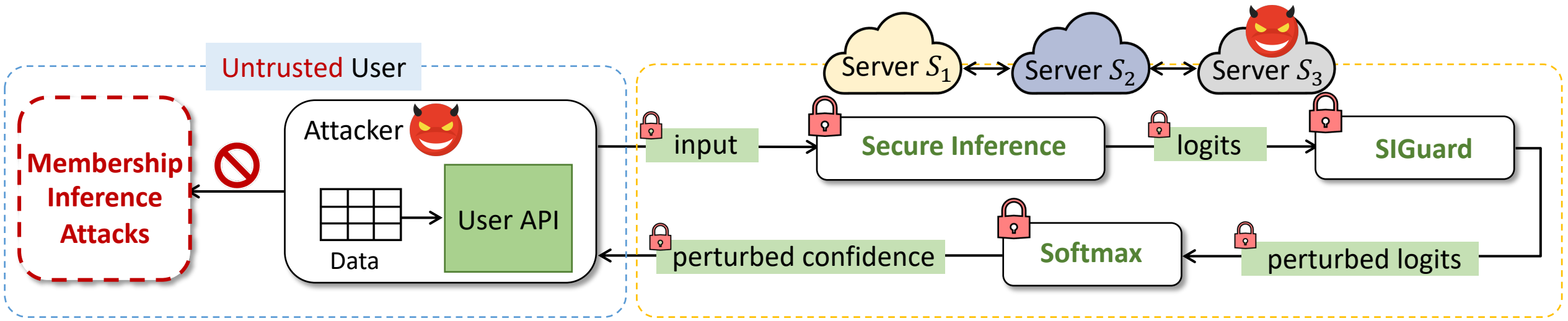
Observation 3: Broaden MIAs attack surface

- MIAs in plaintext can only access perturbed confidence scores



Observation 3: Broaden MIAs attack surface

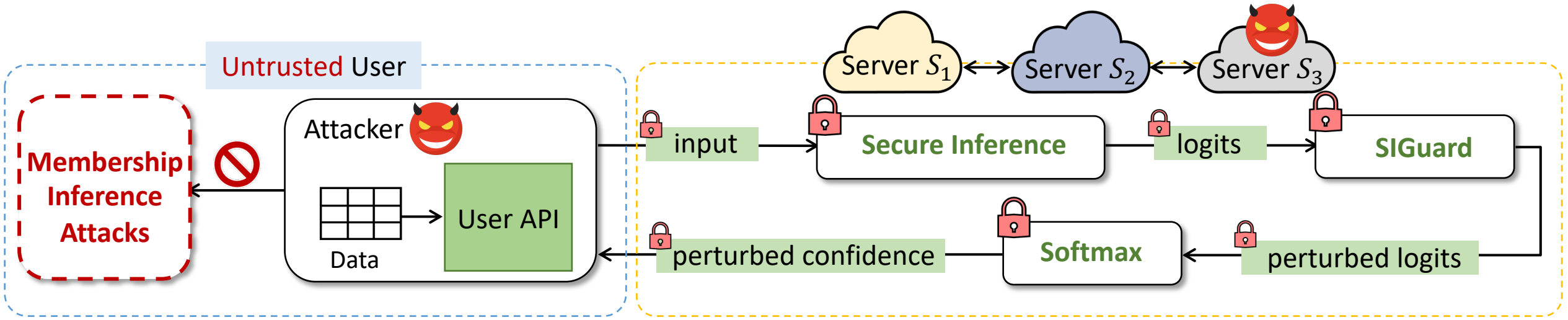
- MIAs in secure inference can obtain knowledge of how the secure defense mechanism operates
  - When the corrupted user colludes with one corrupted cloud server
  - Attempt to leverage auxiliary knowledge to bypass defense



Observation 3: Broaden MIAs attack surface

- MIAs in secure inference can obtain knowledge of how the secure defense mechanism operates
  - When the corrupted user colludes with one corrupted cloud server
  - Attempt to leverage auxiliary knowledge to bypass defense

***RQ3**: How to mitigate the leakages and achieve the stringent privacy guarantee?*



RQ1: Evaluating membership inference against secure inference

5 datasets
- CIFAR-10
- CIFAR-100
- CH-MINIST
- Location30
- Texas100

5 MIAs
- NN-based
- Confidence-based
- Entropy-based
- Modified entropy-based
- LiRA

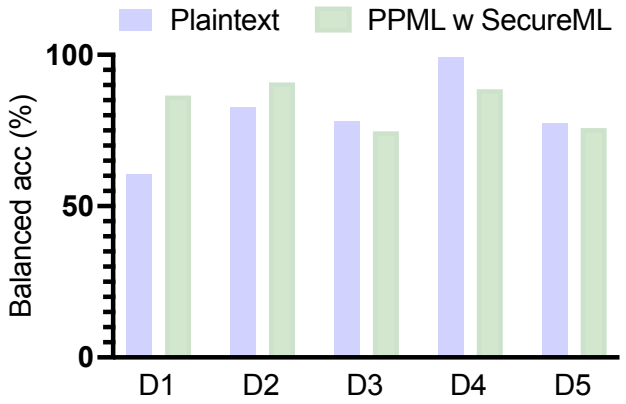
2 metrices
- Balanced accuracy
- TPR at low FPR

4 softmax approximations
- SecureML [1]: $relu(\vec{z}[i])/\sum relu(\vec{z}[j])$
- CrypTen [2]: $\exp(\vec{z}[i]) = \lim_{n\to\infty}(1+\vec{z}[i]/2^n)^n$
- Piranha [3]: $\exp(\vec{z}[i]-\vec{z}_{max}) = \begin{cases} 0.5(\vec{z}[i]-\vec{z}_{max})+1, \vec{z}[i]-\vec{z}_{max} \geq -2 \\ 0, otherwise \end{cases}$
- AS19 [4]: $\exp(\vec{z}[i]) = 2^{\vec{z}[i]\cdot\log_2 e}$

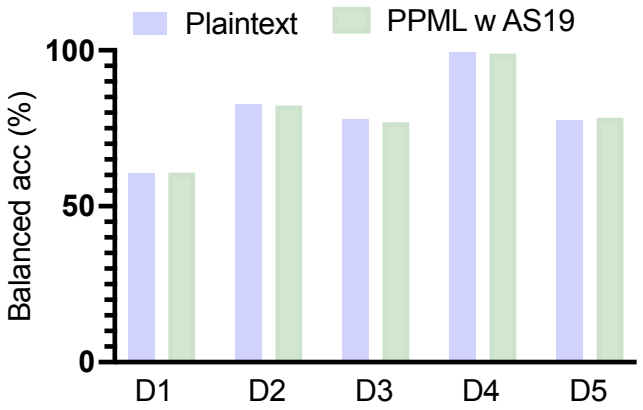
[1] Mohassel, Payman, and Yupeng Zhang. "Secureml: A system for scalable privacy-preserving machine learning." S&P, 2017.
[2] Knott, Brian, et al. "Crypten: Secure multi-party computation meets machine learning." NeuIPS, 2021.
[3] Watson, Jean-Luc, Sameer Wagh, and Raluca Ada Popa. "Piranha: A {GPU} platform for secure computation." USENIX Security. 2022.
[4] Aly, Abdelrahaman, and Nigel P. Smart. "Benchmarking privacy preserving scientific operations" ACNS, 2019.

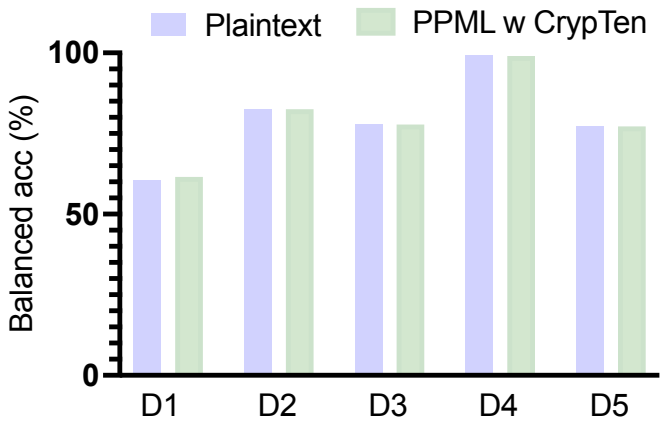
RQ1: Evaluating membership inference against secure inference

- The risk of MIAs against secure inference remains significant, and in some cases may be even higher.



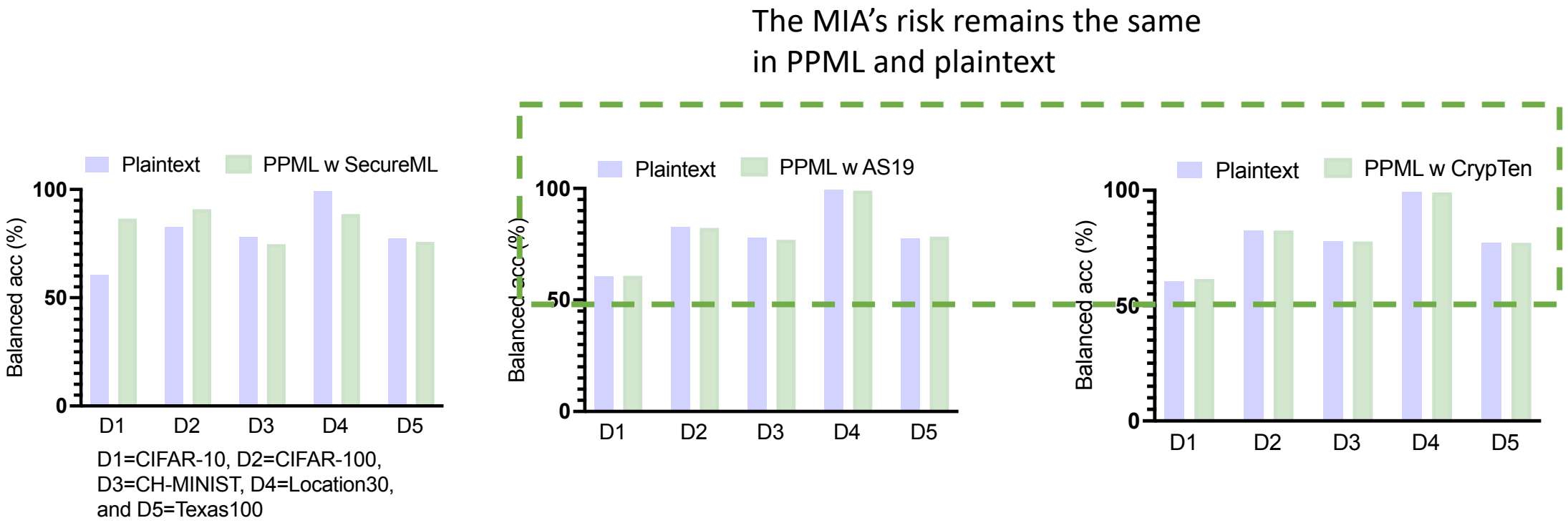
D1=CIFAR-10, D2=CIFAR-100,
D3=CH-MINIST, D4=Location30,
and D5=Texas100





RQ1: Evaluating membership inference against secure inference

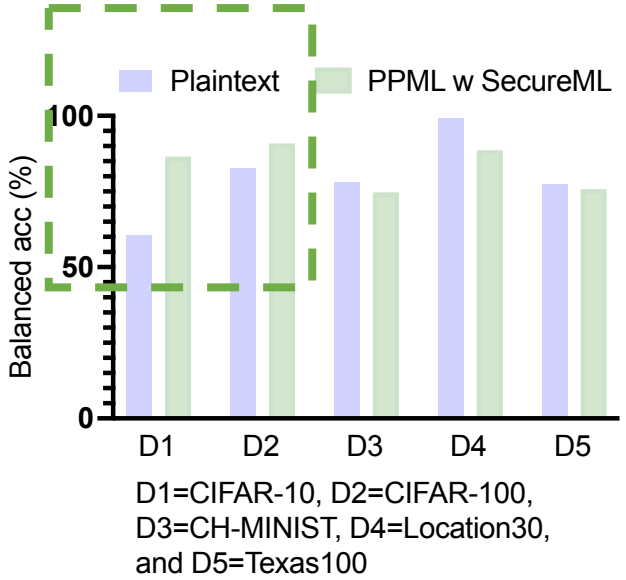
- The risk of MIAs against secure inference remains significant, and in some cases may be even higher.



RQ1: Evaluating membership inference against secure inference

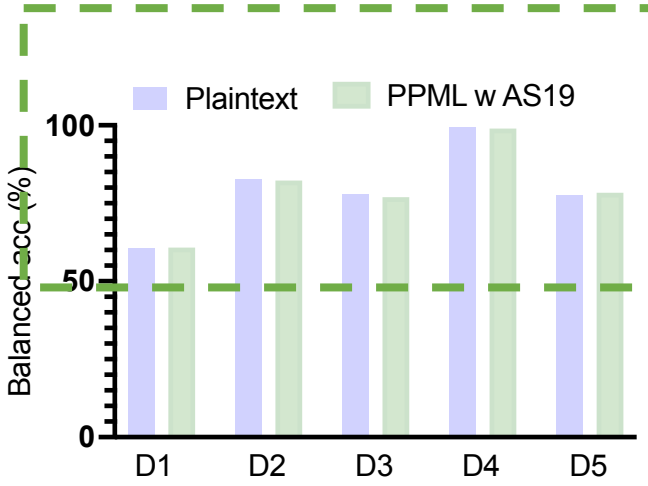
- The risk of MIAs against secure inference remains significant, and in some cases may be even higher.

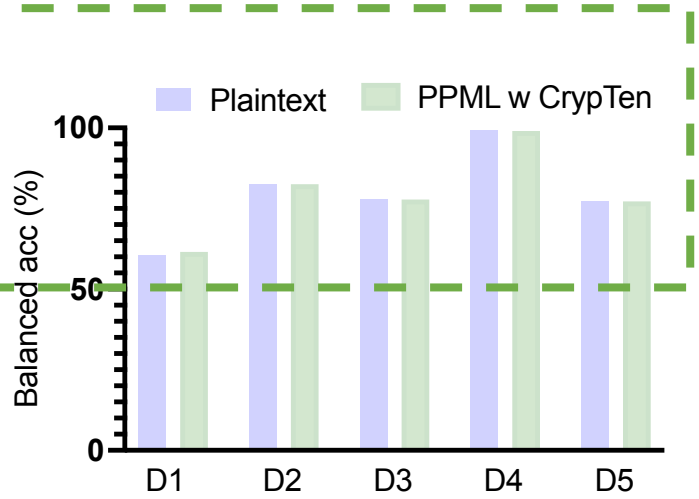
Sometimes, MIA can achieve higher accuracy on the PPML setting



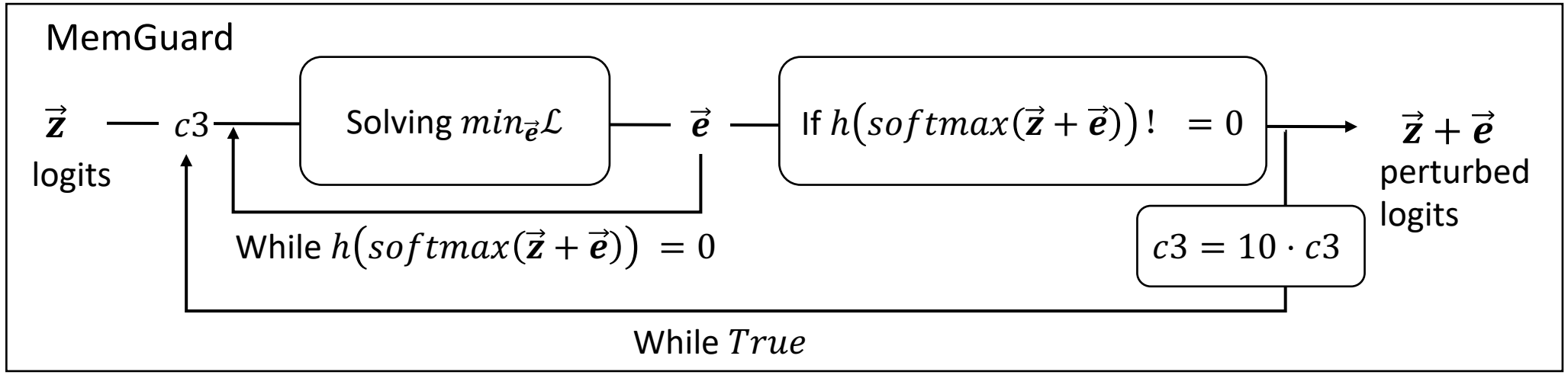
D1=CIFAR-10, D2=CIFAR-100,
D3=CH-MINIST, D4=Location30,
and D5=Texas100

The MIA's risk remains the same in PPML and plaintext



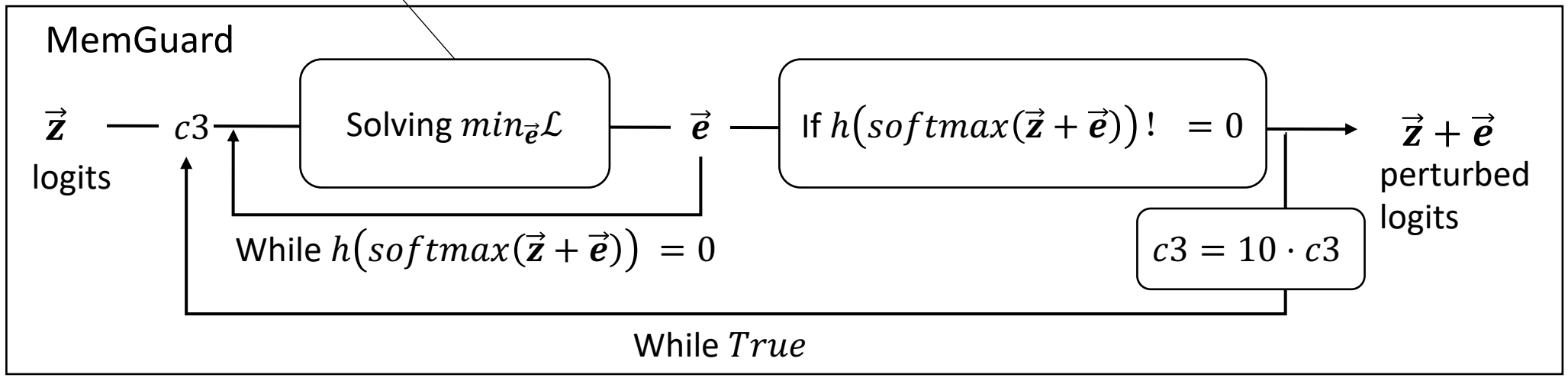


RQ2: Efficient realization of SIGuard



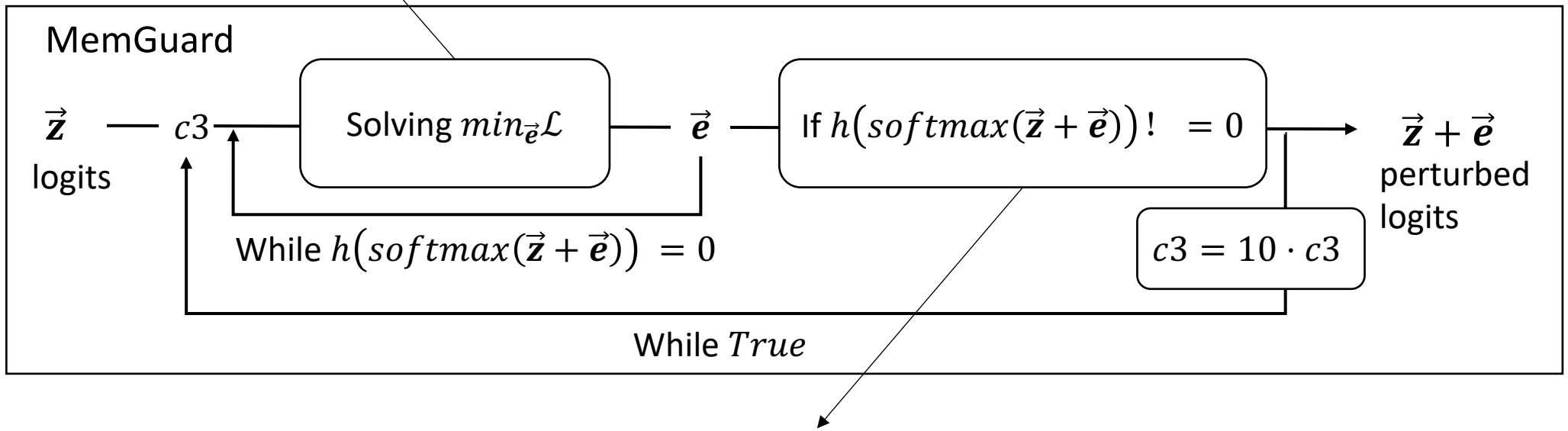
RQ2: Efficient realization of SIGuard

Secure Noise Optimization protocol aims to find an optimal noise vector to perturb the secret-shared confidence vector.



RQ2: Efficient realization of SIGuard

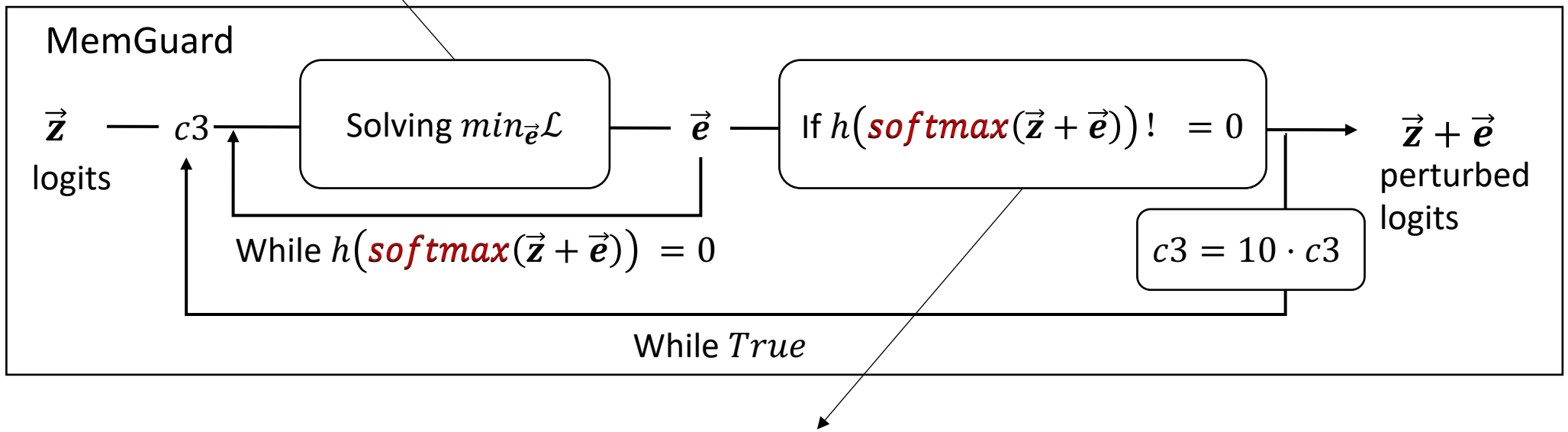
Secure Noise Optimization protocol aims to find an optimal noise vector to perturb the secret-shared confidence vector.



Secure Noise Validation protocol aims to validate that whether the secret-shared noise vector is carefully crafted to preserve the inference accuracy.

RQ2: Efficient realization of SIGuard

Secure Noise Optimization protocol aims to find an optimal noise vector to perturb the secret-shared confidence vector.

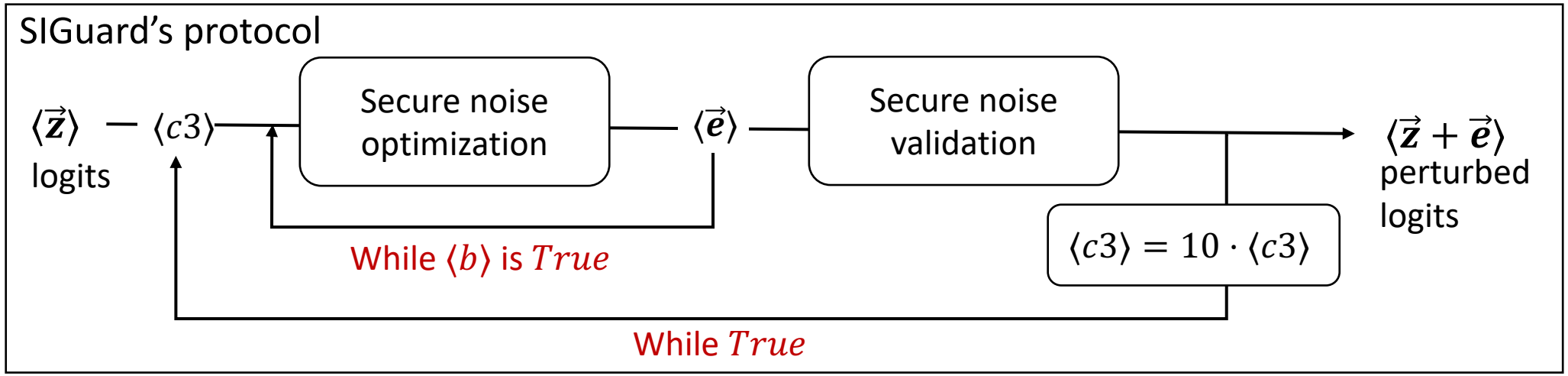


Secure Noise Validation protocol aims to validate that whether the secret-shared noise vector is carefully crafted to preserve the inference accuracy.

Softmax needs a careful design for softmax approximation – whether it affects defense performance

RQ3: Stringent privacy with optimal efficiency

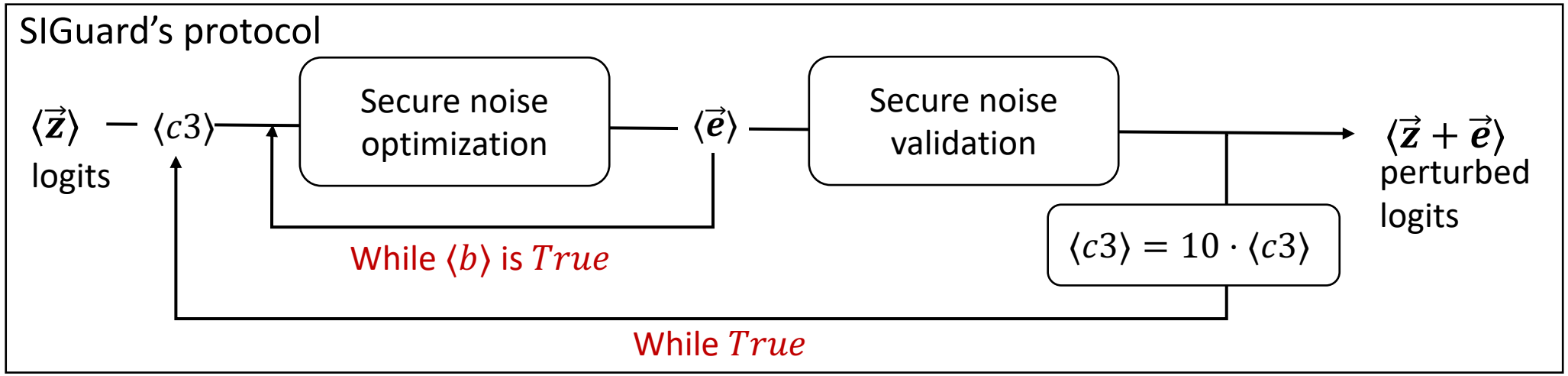
- MPC cannot directly compute `While` loop in the encrypted domain
  - The secret-shared condition needs to be revealed to terminate the loop



[1] J. D. Nielsen and M. I. Schwartzbach, "A domain-specific programming language for secure multiparty computation," in PLAS, 2007.

RQ3: Stringent privacy with optimal efficiency

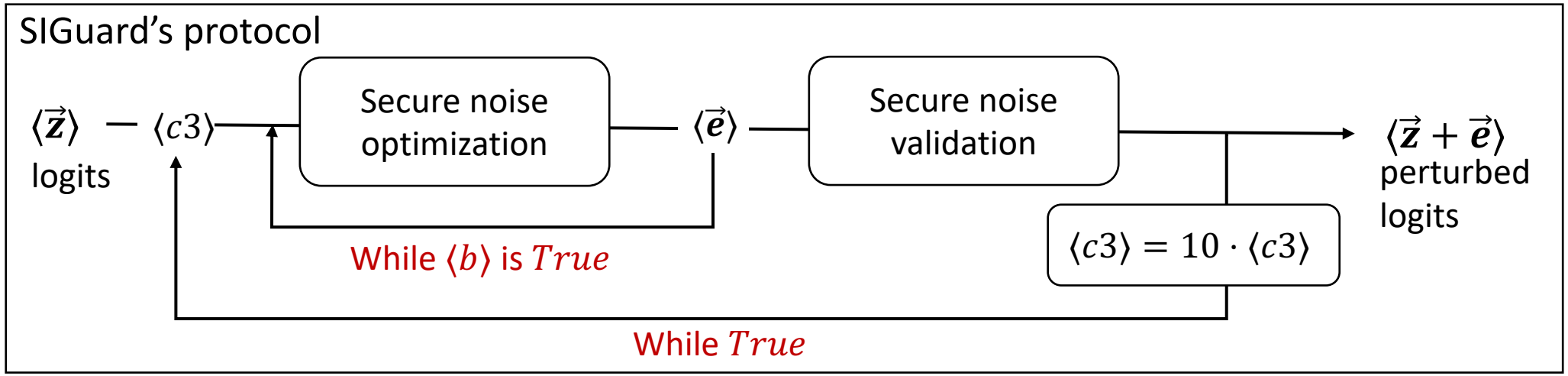
- MPC cannot directly compute `While` loop in the encrypted domain
  - The secret-shared condition needs to be revealed to terminate the loop
- Revealing the condition lets MIA know the number of iterations to find the optimal noise



[1] J. D. Nielsen and M. I. Schwartzbach, "A domain-specific programming language for secure multiparty computation," in PLAS, 2007.

RQ3: Stringent privacy with optimal efficiency

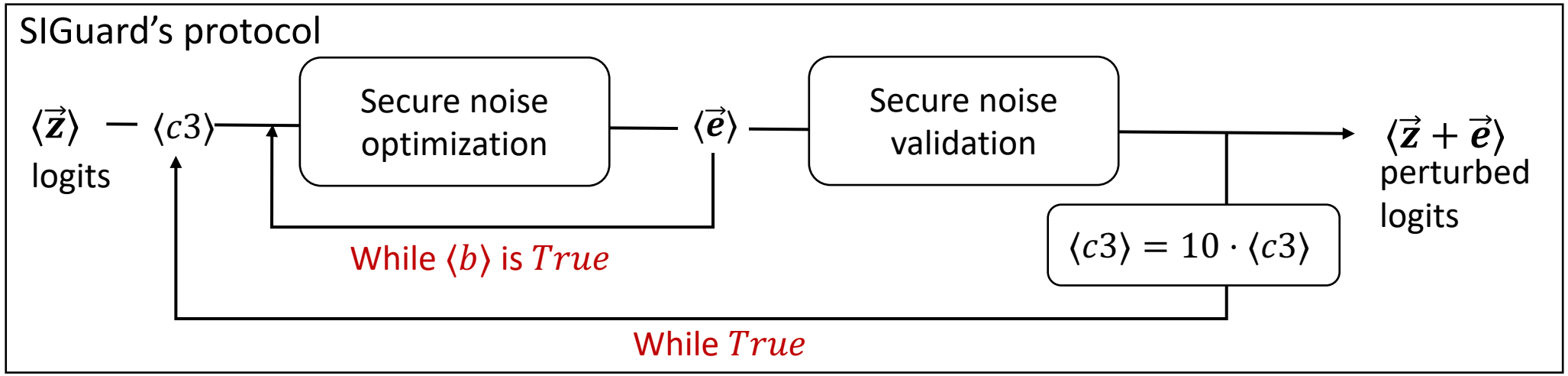
- MPC cannot directly compute `While` loop in the encrypted domain
  - The secret-shared condition needs to be revealed to terminate the loop
- Revealing the condition lets MIA know the number of iterations to find the optimal noise
  - Broadened attack surface: MIA colludes with secure inference adversary
  - Condition $h\big(softmax(\vec{\boldsymbol{z}} + \vec{\boldsymbol{e}})\big) = 0$
- *Would revealing the number of iterations increase MIA's attack performance?*



[1] J. D. Nielsen and M. I. Schwartzbach, "A domain-specific programming language for secure multiparty computation," in PLAS, 2007.

RQ3: Stringent privacy with optimal efficiency

- MPC cannot directly compute `While` loop in the encrypted domain
  - The secret-shared condition needs to be revealed to terminate the loop
- Revealing the condition lets MIA know the number of iterations to find the optimal noise
  - Broadened attack surface: MIA colludes with secure inference adversary
  - Condition $h\big(softmax(\vec{\boldsymbol{z}} + \vec{\boldsymbol{e}})\big) = 0$
- *Would revealing the number of iterations increase MIA's attack performance?*



[1] J. D. Nielsen and M. I. Schwartzbach, "A domain-specific programming language for secure multiparty computation," in PLAS, 2007.

Refinement I: Mitigating potential leakages from iterations

- The revealed number of iterations is linked with data sample's membership information
- Even if the condition is protected, MIA still observes functionality to learn the number of iterations due to collusion

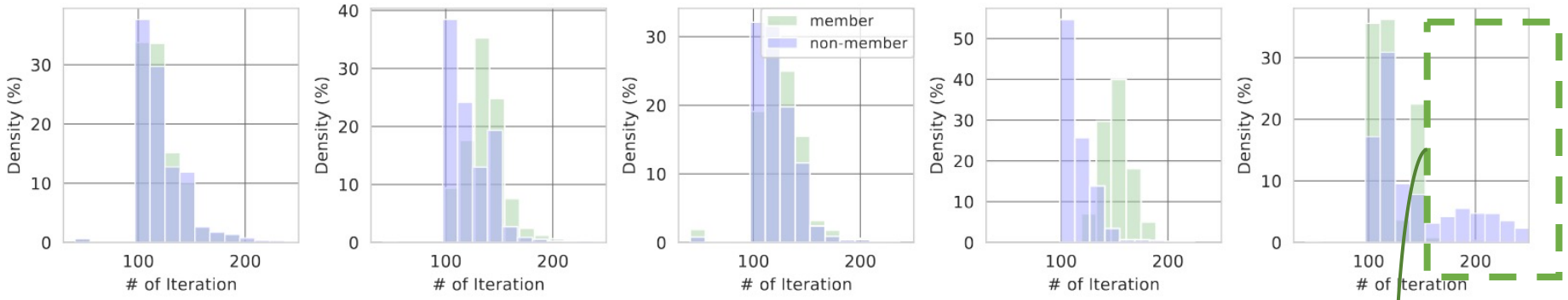
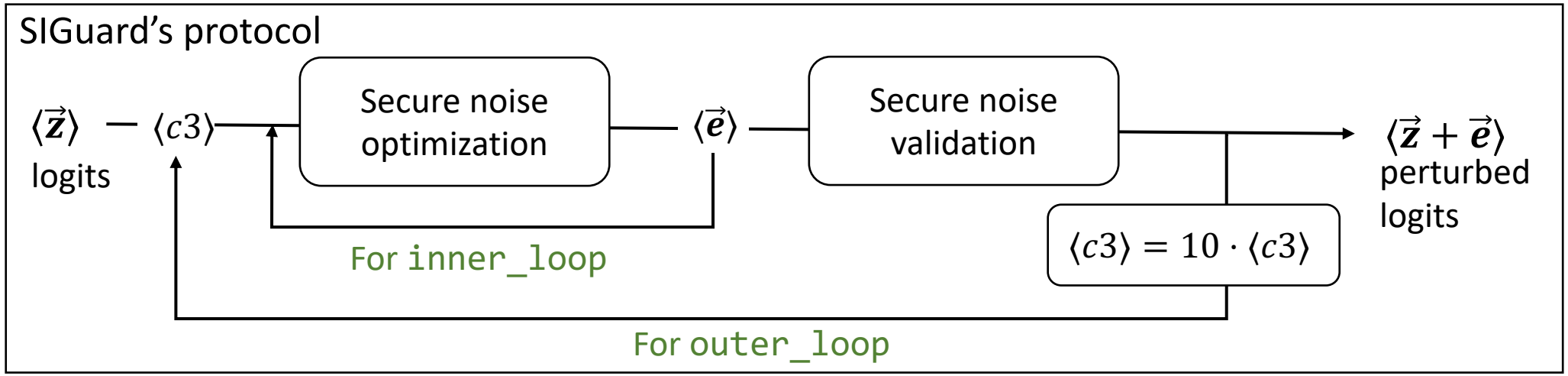


Fig. 4: Comparison of iteration frequency between member and non-member samples.

No member data when #iter >= 160
Must be non-member

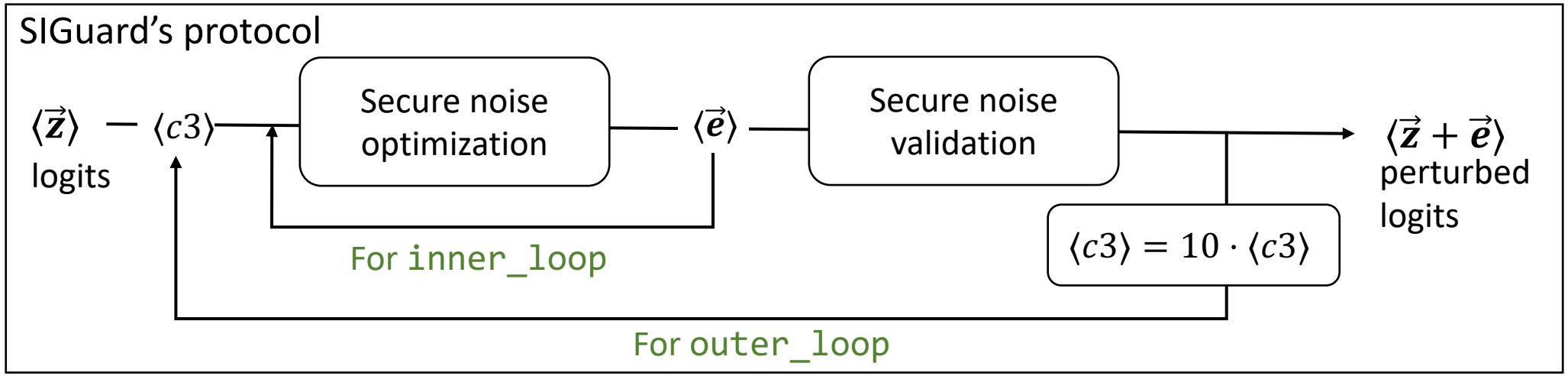
Refinement I: Mitigating potential leakages from iterations

- Difference of total iteration numbers is caused by each sample's optimization hardness
- Convert `While` to `For`
- Use a fixed number of iterations (`inner_loop`, `outer_loop`)



Refinement II: Balanced efficiency & defense effectiveness

- Less iterations: faster v.s. ↓defense effectiveness (↓accuracy)
- More iterations: heavier MPC computation v.s. ↑defense effectiveness
- *How to find suitable `inner_loop`, `outer_loop` balance the efficiency and defense?*

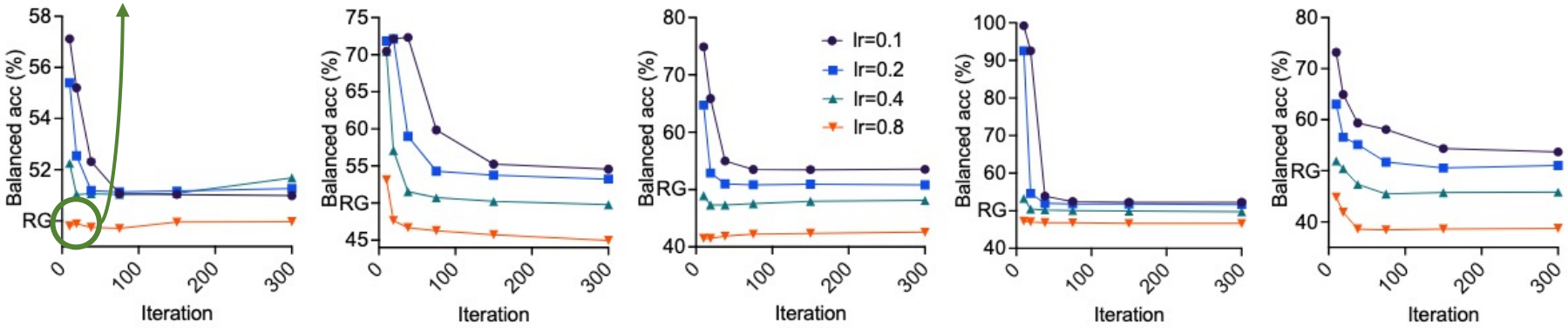


Refinement II: Balanced efficiency & defense effectiveness

- `outer_loop` is fixed to 3

`inner_loop` = 10 (MemGuard sets 300)
lr = 0.8
defense near 50%

Compared with our basic SIGuard
Runtime: basic 64s → 1.1s **58x** savings
Bandwidth: basic 301MB → 5.4MB **55x** savings



(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST (d) Location30 (e) Texas100

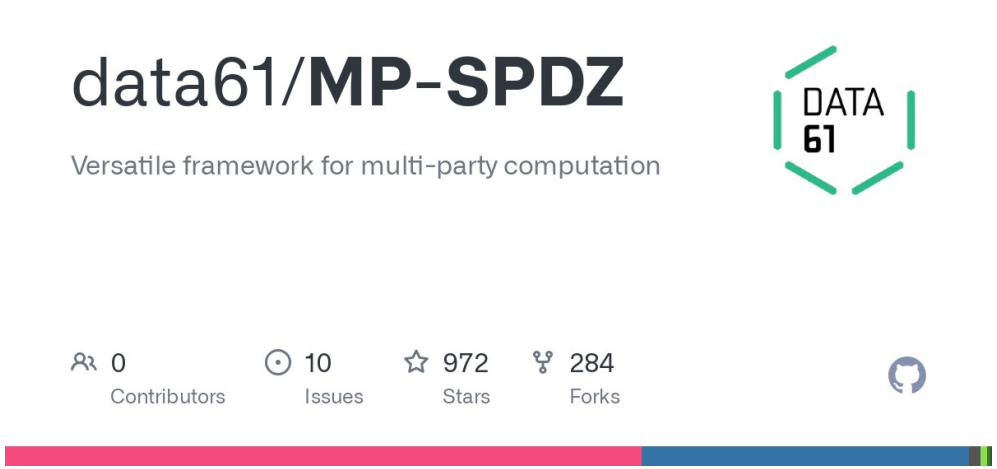
Fig. 5: Comparison of SIGuard's defense performance under different learning rates and iterations ($inner\_loop \in \{10, 19, 38, 75, 150, 300\}$).

Implementations

- We leverage the MP-SPDZ framework as the skeleton of SIGuard.

Q1:*Can SIGuard effectively mitigate the risk of MIAs in secure inference?*

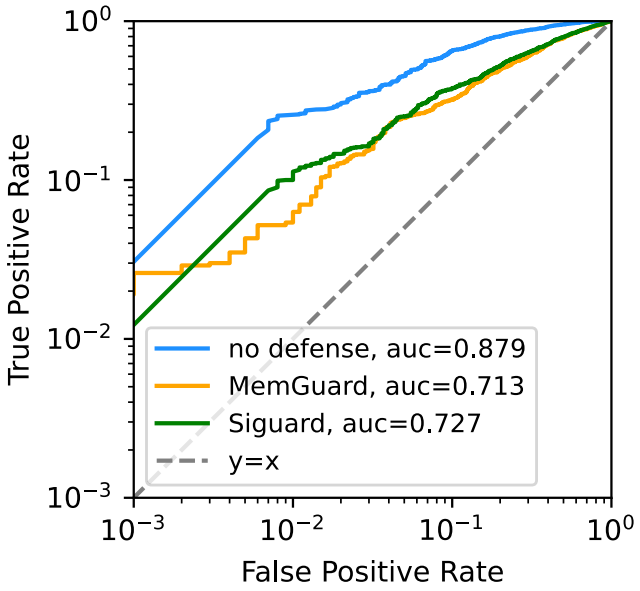
Q2:*Is SIGuard efficient to be integrated into secure inference pipeline?*



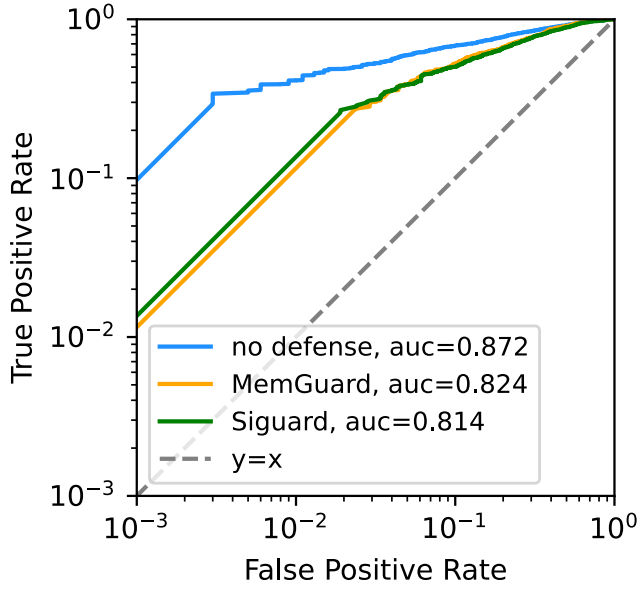
Defense performance of SIGuard

SIGuard can effectively mitigate MIAs from balanced accuracy and TPR @ FPR.

Dataset	Group	NN-based	Entropy-based
CIFAR-10	No defend	57.60%	56.80%
	SIGuard	50.30%	52.00%
CIFAR-100	No defend	68.85%	65.10%
	SIGuard	54.45%	53.45%
CH-MINIST	No defend	76.95%	70.50%
	SIGuard	50.90%	48.90%
Location30	No defend	98.93%	92.40%
	SIGuard	51.86%	50.00%
Texas100	No defend	78.00%	73.00%
	SIGuard	53.90%	52.45%



CIFAR-100



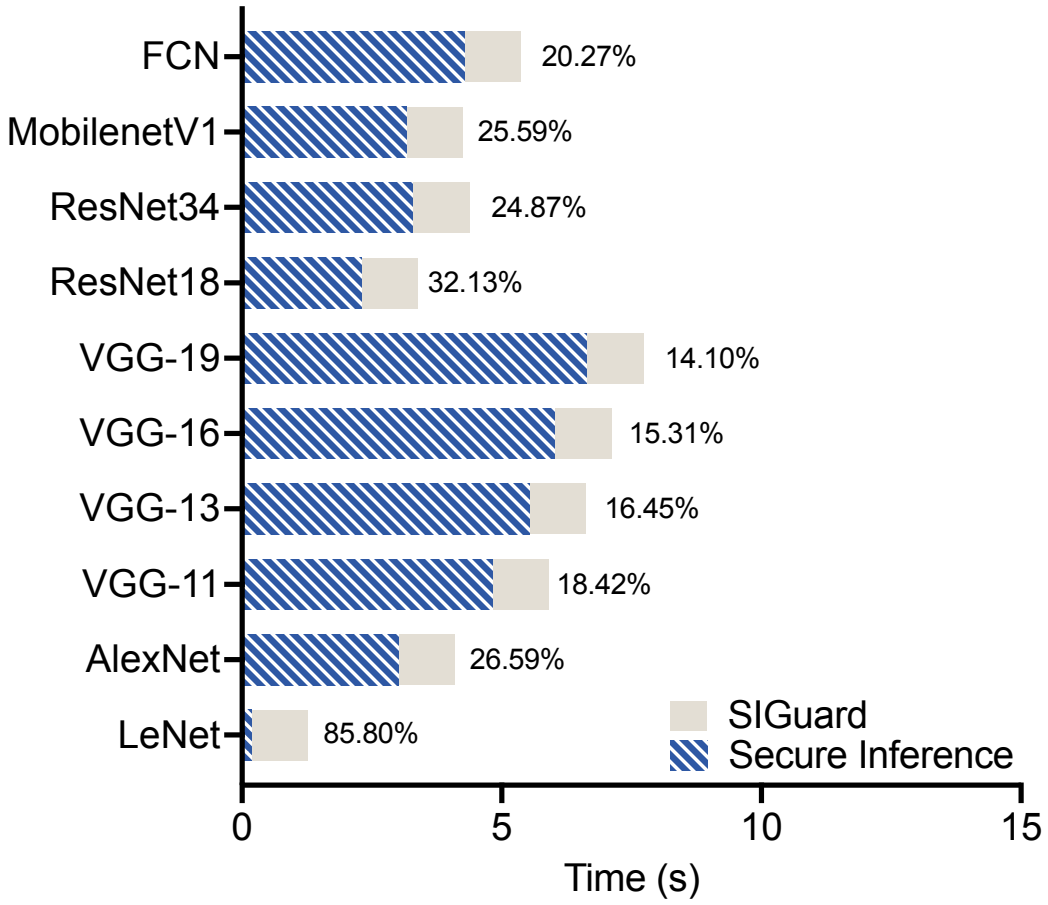
Texas-100

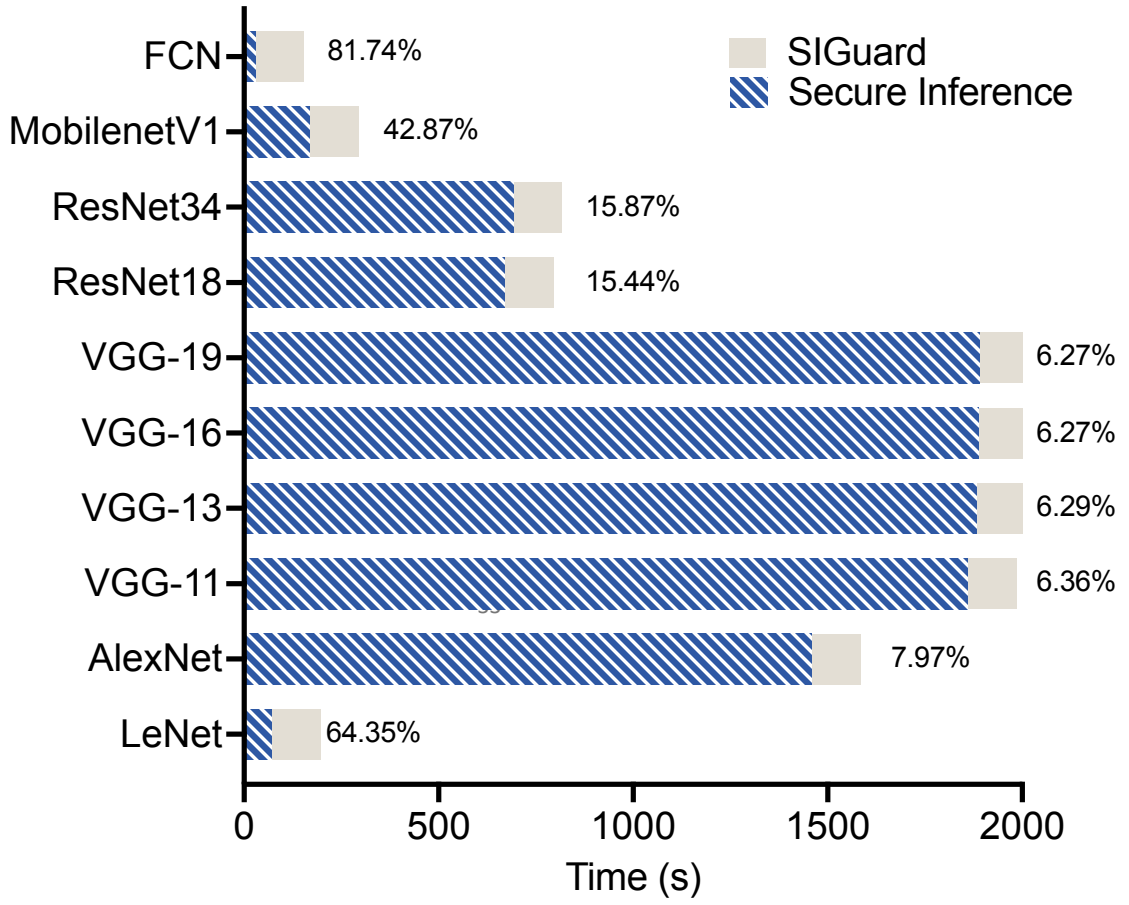
SIGuard also has degradation on the LiRA attack.

Close to random guessing of 50%

SIGuard's efficiency

SIGuard can effectively defeat MIAs for secure inference without introducing dominant overhead.





LAN

WAN

Thanks for listening!

Questions?

Contact me:

xinqian.wang@rmit.edu.au