Poster: Cryptographic Inferences for Video Deep Neural Networks

Bingyu Liu
Illinois Institute of Technology
bliu40@hawk.iit.edu

Rujia Wang
Illinois Institute of Technology
rwang67@iit.edu

Zhongjie Ba
Zhejiang University
zhongjieba@zju.edu.cn

Shanglin Zhou
University of Connecticut
shanglin.zhou@uconn.edu

Caiwen Ding
University of Connecticut
caiwen.ding@uconn.edu

Yuan Hong
University of Connecticut
yuan.hong@uconn.edu

ABSTRACT

Deep neural network (DNN) services have been widely deployed in many different domains. For instance, a client may send its private input data (e.g., images, texts and videos) to the cloud for accurate inferences with pre-trained DNN models. However, significant privacy concerns would emerge in such applications due to the potential data or model sharing. *Secure inferences* with cryptographic techniques have been proposed to address such issues, and the system can perform *secure two-party inferences* between each client and cloud. However, most of existing cryptographic systems only focus on DNNs for extracting 2D features for image inferences, which have major limitations on latency and scalability for extracting spatio-temporal (3D) features from videos for accurate inferences. To address such critical deficiencies, we design and implement the first cryptographic inference system, `Crypto3D`, which privately infers videos on 3D features with rigorous privacy guarantees. We evaluate `Crypto3D` and benchmark with the state-of-the-art systems on privately inferring videos in the UCF-101 and HMDB-51 datasets with C3D and I3D models. Our results demonstrate that `Crypto3D` significantly outperforms existing systems (*substantially extended to inferences with 3D features*): execution time: 186.89× vs. CryptoDL (3D), 63.75× vs. HEANN (3D), 61.52× vs. MP-SPDZ (3D), 45× vs. E2DM (3D), 3.74× vs. Intel SGX (3D), and 3× vs. Gazelle (3D); accuracy: 82.3% vs. below 70% for all of them.

CCS CONCEPTS

• **Security and privacy** → *Privacy-preserving protocols*;

KEYWORDS

Privacy; Secure Multiparty Computation; Deep Neural Network

ACM Reference Format:
Bingyu Liu, Rujia Wang, Zhongjie Ba, Shanglin Zhou, Caiwen Ding, and Yuan Hong. 2022. Poster: Cryptographic Inferences for Video Deep Neural Networks. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3548606.3563543

CCS '22, November 7–11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9450-5/22/11.
https://doi.org/10.1145/3548606.3563543

Table 1: Comparison of secure inferences (HE: Homomorphic Encryption, GC: Garbled Circuits, SS: Secret Sharing, TEE: Trusted Execution Environment, Mix: Mixed MPC).

	Method	3D	Spatial	Temporal
CryptoNets [3], CryptoDL[5]	HE	✗	✓	✗
MiniONN [8], DeepSecure [12]	GC	✗	✓	✗
PSA [16]	SS	✗	✓	✗
MLCapsule[4]	TEE	✗	✓	✗
Visor [11]	TEE	✗	✓	✓
Gazelle [7], Delphi [9]	Mix	✗	✓	✗
GALA [15], PPVC [10]	Mix	✗	✓	✗
`Crypto3D` (Ours)	Mix	✓	✓	✓

1 INTRODUCTION

Recently deep neural networks (DNNs) have been increasingly deployed by the cloud to provide services for object detection, image and video classification, anomaly detection, etc. The client may send its data to the cloud for accurate classification and prediction using the pre-trained DNN models. However, severe privacy concerns may occur between the client and cloud. In video inferences, the users' videos involve considerable amounts of sensitive information (e.g., human face, identities, activities, and workspace). Directly disclosing them to the cloud would compromise the privacy of users. Indeed, the pre-trained DNN model should also be considered as the proprietary information for the cloud, which cannot be shared.

To eliminate such privacy risks, cryptographic protocols [1, 8] are designed for *secure inferences* (as summarized in Table 1). A secure inference protocol allows the client to send its private input data (encrypted), and privately obtain the learning result from the cloud. Neither party can learn anything regarding the model weights and private inputs from each other. Many existing works [8] use one or more cryptographic techniques such as Fully Homomorphic Encryption (FHE) [1], Garbled Circuits (GC) [14] and Secret Sharing (SS) [8] to compose the protocols. FHE can provide higher privacy guarantees, but it brings expensive computational overheads. Moreover, some non-polynomial functionalities (e.g., Non-linear Activation Functions ReLU) cannot be supported. Garbled circuits support arbitrary functionality, but it results in significant computation and communication overheads. Trusted Execution Environment (TEE) [4] provides secure *enclave* for the isolated sensitive computation with attestation. It ensures data privacy and integrity without provable guarantees. Moreover, current TEEs are not scalable enough for processing large amounts of data. Thus, directly using such systems are not ideal for secure DNN inferences. The Delphi system [9] was recently proposed as one

of the state-of-the-art efficient cryptographic inference systems. It outperforms other protocols in both latency and communication cost for image DNN with a hybrid cryptographic protocol. Unfortunately, *securely inferring images based on 2D features* by Delphi (the state-of-the-art) is far from enough for video-based applications. Compared with the 2D, most 3D ConvNets have to infuse the temporal information of the videos after each convolution/pooling operations. Performing 3D convolution and pooling operations are supposed to deliver temporal information across all the neural network layers to the end. Integrated with both spatial and temporal information in each feature, 3D ConvNets have proven to be more accurate on video inferences than 2D ConvNets [2, 13]. However, to our best knowledge, cryptographic inferences on 3D features for video DNNs have not been studied yet in literature.[1]

To fill this gap, we design and implement the first cryptographic inference system (namely "Crypto3D") that privately infers videos based on 3D spatial-temporal features (both C3D [13] and I3D [2]). Also, we further boost the system efficiency with optimized matrix operations and ciphertext packing technique.

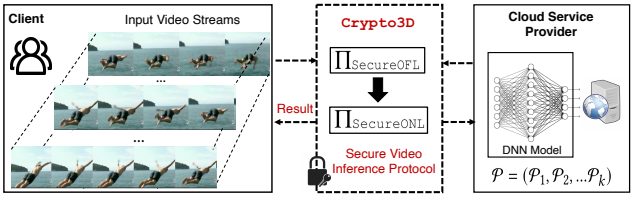

Figure 1: Crypto3D Framework

2 CRYPTOGRAPHIC PROTOCOL

Threat Model. In Crypto3D, each client holds its video streams and it expects not to disclose the content of video to the cloud or other video analytics services. We assume that computing the 3D and the DNN architecture are known to the public (i.e., dimensions and type of each layer in the neural networks), except the parameter of model weights. Based on the proposed cryptographic protocols, the privacy of input video and model weights are guaranteed.

Ciphertext Packing. Our Crypto3D contains offline $\prod_{\text{SecureOFL}}$ and online inference/predication $\prod_{\text{SecureONL}}$ phase. Assume that the pre-trained DNN model from the server will not be changed and updated. The offline phase is supposed to be independent of the input data from the client. Once the offline $\prod_{\text{SecureOFL}}$ is completed, the input data given by the client will be sent to the cryptographic protocol for executing the online phase. However, the arithmetic operations of the encrypted matrices are involved and it leads to the inefficiency for the high-dimensional data tensors computation.

To mitigate this issue, Crypto3D utilized the optimized matrix permutation [6] to efficiently perform the operation of matrix computation with the ciphertext packing and parallelism. The operation of the matrix multiplication can be considered as the sum of component-wise products with the specific permutations of the matrices themselves. Assume that there are two square matrices with size $n \times n$, the n permutations of the matrix A via the followings symmetric permutations: $\sigma(A)_{i,j} = A_{i,i+j}$, $\tau(A) = A_{i+j,j}$ and $\phi(A)_{j,j} = A_{i,j+1}$, $\psi(A) = A_{i+1,j}$, where ϕ and ψ are denoted as the shifting functions for column and row, respectively. Then, the multiplication of two matrices (we denote A and B) with the order d can be computed as: $A \cdot B = \sum_{k=1}^{d-1}(\phi^k \odot \sigma(A)) \times (\psi^k \odot \tau(B))$ where $\odot$ refers to the component-wise product and k is used to represent the number of times for perturbation. As such, we can efficiently compute the two matrix multiplications. In Crypto3D, we utilize the function $\text{Permu}(\cdot)$ to represent the computation of the n permutation operations. To boost the efficiency, we also utilize the vectorable homomorphic encryption "Ciphetext packing". We use the $\text{Encode}(\cdot)$ to refer to the matrix transformations, which transforms a matrix into a plaintext vector with encoding map functions. Our Crypto3D uses the optimized matrix multiplication and ciphertext packing [6] for the efficiency improvement. Since we can pack all the inputs into a single ciphertext and perform layer computation (e.g., convolutions) in parallel, we can enable the SIMD parallelism with the ciphertext packing.

2.1 Protocol Design

As shown in Figure 1, Crypto3D secures the *two-party inference* between the client and the cloud service provider. The Crypto3D by extending the design in DELPHI [9]: the neural network is processed with linear and non-linear layer one after the other, and the output will be delivered as input for the next layer.

Offline Phase ($\prod_{\text{SecureOFL}}$). Our Crypto3D provides the offline phase execution, which can be executed before the input is known. First, (pk, sk) can be fetched via the KGen algorithm. The input value $\mathbf{x}$ is independent of the *offlinePhase()* execution. We denote $[\![r_i]\!] \leftarrow \mathbb{R}^n, i \in [1, .., l]$ and $[\![s_i]\!] \leftarrow \mathbb{R}^n, i \in [1, .., l]$ as the random masking vectors for the i-th layer. In the linear layer,the $\text{Enc}(\text{pk}, [\![r_i]\!])$ is sent to the server by the client. With the Eval procedure, the server computes the $\text{Enc}(pk, (\mathcal{P}_i \cdot [\![r_i]\!] - [\![s_i]\!]))$ and send its back to the client. Then, the client decrypts and obtains decrypted value for all layers. Thus, the additive secret sharing of $\mathcal{P}_i \cdot [\![r_i]\!]$ is held by both the client and the server before the online phase execution. Regarding the non-linear layer execution, the execution of activation function depends on what type of function. The garbled circuit is constructed via GC schemes. It helps to solve the ReLu function by exchanging the labels for input wires with $[\![r_{i+1}]\!]$ and $\mathcal{P}_i \cdot [\![r_i]\!] - [\![s_i]\!]$. On the other hand, the Beaver's triples protocol is used for the polynomial approximation functions.

Online Phase ($\prod_{\text{SecureONL}}$). Given the input $\mathbf{x}$, the server receives $\mathbf{x} - [\![r_1]\!]$. At this time, the additive secret shares of $\mathbf{x}$ are held by the client and server, respectively. At the beginning of the i-th layer evaluation, $\mathbf{x}_i$ can be fetched from the first $(i-1)$ layers of the neural network. The client holds $[\![r_i]\!]$ while server holds $x_i - [\![r_i]\!]$. For the evaluation of the linear layer(s), the server computes $\mathcal{P}_i \cdot (\mathbf{x}_i - [\![r_i]\!])$, which ensures that the additive shared secrets of $\mathcal{P}_i \cdot \mathbf{x}_i$ are held by the client and server, respectively. Once the linear layer is completed, $\mathcal{P}_i \cdot (\mathbf{x}_i - [\![r_i]\!]) + [\![s_i]\!]$ and $\mathcal{P}_i \cdot [\![r_i]\!] - [\![s_i]\!]$ are held by the server and client, respectively. Similarly, we use the garbled circuits and Beaver's multiplication for evaluating the non-linear layers. For the Garbled Circuits evaluation, the client receives the garbled labels from the server, which is corresponding to the $\mathcal{P}_i \cdot$

[1]Visor [11] provides confidentiality for analyzing video streams via a hybrid TEE system. However, it still privately infers data (e.g., object detection and tracking) based on 2D features. PPVC [10] preserves privacy in video classification based on MPC, but it still utilizes the 2D ConvNets without fully preserving temporal information.

Table 2: Comparison with the state-of-the-art systems (significantly extended from 2D to 3D) on the UCF101 dataset with C3D model. The execution time of Crypto3D is over 186.89×, 63.75×, 61.52×, 45× 3.74× and 3× faster than CrytoDL (3D), HEANN (3D), MP-SPDZ (3D), E2DM (3D), Intel SGX (3D) and Gazelle (3D), respectively. PPVC [10] uses 2D CNN Network.

System	Method	Library	Network	Runtime w. GPU (Sec)	Speedup (×)	Amortized (Sec)	Accuracy
Gazelle (3D)	HE, GC, SS	PALISADE	C3D	1916.48	3.00×	2.48	> 49.4%
Intel SGX (3D)	TEE	-	C3D	2387.77	3.74×	3.08	49.4%
PPVC [10]	MPC, SS	MP-SPDZ	2D CNN	511.64 (from [10])	-	-	56%
MP-SPDZ (3D)	MPC, SS	MP-SPDZ	C3D	39303.72	61.52×	50.78	> 56%
CryptoDL (3D)	HE	HELIB	C3D	119388.28	186.89×	154.25	> 62%
HEANN (3D)	HE	HEANN	C3D	40725.29	63.75×	52.62	> 62%
E2DM (3D)	HE	HEANN	C3D	28747.26	45.00×	37.14	> 62%
Crypto3D (Ours)	HE, GC, SS	SEAL	C3D	638.83	-	0.83	82.3%

$(\mathbf{x}_i - [\![r_i]\!]) + [\![s_i]\!]$. With these labels, the garbled circuit is evaluated to return the output of one-time pad (OTP $(x_{i+1} - [\![r_{i+1}]\!])$) to the server. The $x_{i+1} - [\![r_{i+1}]\!]$ is obtained by the server with one-time pad key. On the other hand, the Beaver's multiplication procedure is executed for the polynomial approximation evaluation. The client and sever will hold the $[x_{i+1}]_1$ and $[x_{i+1}]_2$, separately after the Beaver's multiplication procedure. At this time, the client sends the results of the $[x_{i+1}]_1 - [\![r_{i+1}]\!]$ to the server. The $x_{i+1} - [\![r_{i+1}]\!]$ will be obtained by adding the $[x_{i+1}]_2$. Finally, the client learns the x_l.

3 EVALUATION

Setting and Datasets. Our Crypto3D is implemented with Rust, Python and C++. All the experiments are evaluated on a Ubuntu 20.04.2 LTS server with the NVIDIA-SMI 460.80 GPU. We evaluate C3D and I3D features on the UCF-101 and HMDB-51 datasets.

Comparison with Existing Systems. We provide the performance comparison of Crypto3D and other privacy-preserving frameworks with 3D structure. As discussed in Section 1, all the benchmark systems cannot be directly applied to for video inferences based on the C3D model. We significantly extend them by modifying the 2D CNN network to embed with 3D architecture. With the 3D filters, the spatio-temporal features are able to be extracted. We re-implement the following systems on the C3D model: Gazelle (3D), Intel SGX (3D), MP-SPDZ (3D), CryptoDL (3D), HEANN (3D) and E2DM (3D). However, Delphi and GALA cannot be extended due to the 2D structure or lack of source codes. Table 2 summarizes the cryptographic method, library, total execution time, speedup and amortized time. Crypto3D significantly outperforms all other benchmarks. The execution time of Crypto3D is over 186.89×, 63.75×, 61.52×, 45× 3.74× and 3× faster than CrytoDL (3D), HEANN (3D), MP-SPDZ (3D), E2DM (3D), Intel SGX (3D) and Gazelle (3D), respectively. These results show that Crypto3D is much more efficient in 3D privacy-preserving video input inference. Additionally, Crypto3D only takes 0.83 sec on average to process the secure inference for each frame, while other HE-based frameworks take much longer time because of the computational overhead. Note that the accuracy of the all other benchmarks is only less than 70% while Crypto3D can achieve the accuracy of 82.3%.

4 CONCLUSION

Many existing techniques are proposed to perform the *secure two-party inferences* with the cryptographic schemes for the DNNs. However, they cannot be directly applied to video inferences which extracts spatio-temporal (3D) features for more accurate video recognition. In this paper, we propose crypto3D, the first cryptographic neural network inference based on 3D features, which achieves significant performance by (i) privately inferring videos on 3D spatial-temporal features; (ii) involving an optimized matrix operations and ciphertext packing technique in Crypto3D for efficiency boosting. In addition, we significantly modify most of the state-of-the-art secure DNNs protocols (CryptoDL, HEANN, MP-SPDZ, E2DM, Intel SGX, and Gazelle) to privately infer videos with 3D features as the benchmarks. Finally, it can also guarantee 82.3% accuracy on inferring videos with 3D features, which is significantly more accurate than all of other benchmarks.

Acknowledgments

This work is partially supported by the National Science Foundation (NSF) under the Grants No. CNS-2046335, CNS-2034870, as well as the Cisco Research Award. Also, the authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Trans. Comput. Theory* 6, 3 (2014).
[2] J. Carreira and A. Zisserman . Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE CVPR 2017*.
[3] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In *ICML 2016*.
[4] Lucjan Hanzlik, Yang Zhang, Kathrin Grosse, Ahmed Salem, Maximilian Augustin, Michael Backes, and Mario Fritz. . MLCapsule: Guarded Offline Deployment of Machine Learning as a Service. In *IEEE CVPR Workshops 2021* .
[5] Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Rebecca N. Wright. Privacy-preserving Machine Learning as a Service. In *PETS,* 2018, 123–142.
[6] X. Jiang, M. Kim, K. E. Lauter, and Y. Song. Secure Outsourced Matrix Computation and Application to Neural Networks. In *ACM CCS* 2018.
[7] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. GAZELLE: A Low Latency Framework for Secure Neural Network Inference. In *USENIX Security 2018*.
[8] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious Neural Network Predictions via MiniONN Transformations. In *ACM CCS 2017*.
[9] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. Popa. 2020. Delphi: A Cryptographic Inference Service for Neural Networks. In *USENIX Security 2020*
[10] Sikha Pentyala, Rafael Dowsley, and Martine De Cock. 2021. Privacy-Preserving Video Classification with Convolutional Neural Networks. In *ICML 2021*.
[11] Rishabh., G. Ananthanarayanan, S. Setty, S. Volos, and R. Popa. 2020. Visor: Privacy-Preserving Video Analytics as a Cloud Service. In *USENIX Security 2020*
[12] B. D. Rouhani, M. S. Riazi, and F. Koushanfar. Deepsecure: scalable provably-secure deep learning. In *DAC 2018*.
[13] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE ICCV 2015*
[14] A. Chi-Chih Yao. 1986. How to Generate and Exchange Secrets. In *IEEE FOCS*.
[15] Q. Zhang, C. Xin, and H. Wu. GALA: Greedy ComputAtion for Linear Algebra in Privacy-Preserved Neural Networks. In *NDSS 2021*.
[16] K. A. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *ACM CCS 2017*.