Interrupt Trace Fusion for Enhanced Website Fingerprinting Attacks under Defensive Mechanisms

Yefeng Lv
College of Information Engineering
Shanghai Maritime University
Shanghai, China
email: 202330310295@stu.shmtu.edu.cn

Jiajia Jiao
College of Information Engineering
Shanghai Maritime University
Shanghai, China
email: jiaojiajia@shmtu.edu.cn

Hong Yang
College of Information Engineering
Shanghai Maritime University
Shanghai, China
email: 202230310104@stu.shmtu.edu.cn

Ran Wen
College of Information Engineering
Shanghai Maritime University
Shanghai, China
email: 202230310052@stu.shmtu.edu.cn

***Abstract*—Machine learning has significantly enhanced the effectiveness of website fingerprinting attacks, increasing the success rate of privacy leakage to 93.7%. Website fingerprinting attacks analyze various side-channel signals to deduce the specific websites a user visits, thus posing a threat to user privacy. In response to this threat, advanced defense mechanisms have been developed, such as randomized timers. These defenses alter the pattern of interrupt events, reducing the success rate of interrupt-based side-channel attacks to 1.8%, thereby significantly disrupting attackers' ability to accurately infer information. To counter these effective defenses, we propose an interrupt trace fusion-based attack model. By combining interrupt traces from multiple attack sources, this model captures a more comprehensive set of interrupt features, thereby improving both the accuracy and stability of the attacks. Compared to single-source methods, this fusion model can still effectively extract useful information under strong defensive measures, enhancing overall attack performance. Experimental results show that this model raises the attack success rate from 93.7% to 94.9% under no defense. Under robust defenses, its accuracy increases from 1.8% to 45.3%.**

***Index Terms*—Website Fingerprinting, Interrupt Trace Fusion, Defensive Mechanism**

I. Introduction

Website fingerprinting attacks are a form of side-channel attack for inferring the websites a user visits. Attackers can collect system or network traces generated during a user's browsing activity and analyze them using machine learning models to infer the user's browsing behavior. Despite the use of encryption technologies, these attacks can still capture sensitive user data, such as personal information and browsing habits. Existing research has shown that microarchitectural side-channel techniques are highly effective in identifying different websites [1]–[4].

Depending on the source of website fingerprinting information, the related attack methods vary significantly. Attackers usually launch attacks through network traffic characteristics [5]–[10], cache behaviors [1], [11], or operating system interrupt traces [12]–[14]. Each method bypasses encryption protections at different levels to identify the websites a user visits. Traffic-based attacks infer user browsing behavior by analyzing patterns in encrypted communication, such as packet size, direction, and timing intervals. Shen et al. proposed Robust Fingerprinting (RF) [6], which uses the Traffic Aggregation Matrix (TAM) for resilient feature representation, effectively countering various defenses such as WTF-PAD [15], Front [16], and TrafficSliver [17], with an accuracy range of 97.51%-96.98%. On the other hand, cache side-channel attacks infer user activity by monitoring access patterns to the processor cache caused by different websites. These methods have been extensively applied in the field of website fingerprinting attacks. Unlike cache-based attacks, interrupt trace-based website fingerprinting attacks are more covert, capable of bypassing encrypted traffic defenses and predicting the websites a user visits directly through hardware and operating system behaviors.

In interrupt trace-based website fingerprinting attacks [12]–[14], attackers monitor the frequency and timing of system interrupts to identify the websites a user visits. When a user loads a website, the system generates a series of interrupts (such as network card or other device interrupts). By capturing and analyzing these interrupts, attackers can construct feature vectors and use machine learning models to make accurate predictions. This type of attack is highly covert and is not affected by encrypted traffic. Therefore, we focus on this interrupt-based fingerprinting attacks in the paper.

In order to defend against the novel website fingerprinting attacks, researchers have proposed various defense mechanisms aimed at reducing the attacker's reliance on different side-channel features, thereby lowering their accuracy. For network traffic based attacks, common defenses [10], [18], [19] include traffic padding, traffic obfuscation, and encrypted tunneling, which disrupt the attacker's ability to analyze packet patterns. For cache-based attacks, defense methods [20] primarily focus on cache isolation, cache flushing, and noise injection to interfere with the attacker's monitoring of cache access patterns. As for interrupt trace-based attacks, defense mechanisms [13] often introduce randomness or noise into the interrupts, disrupting the attacker's analysis of interrupt traces and reducing the effectiveness of the attack. These defense methods significantly decrease the attack success rate down to 1.8% by breaking the inherent characteristics of various side-channel traces at different levels.

To enhance user privacy security, it is critical to develop new attacks for designing effective defenses. Therefore, we propose a novel approach that leverages interrupt trace fusion to enhance the effectiveness of these attacks in the paper. By integrating interrupt features from multiple attack sources, our model significantly improves the accuracy and robustness of website identification, even in the presence of powerful defensive mechanisms.

Our work mainly includes the following contributions:

- Using Interrupt Trace Fusion Technology to enhance attacks. To the best of our knowledge, we are the first to propose a model that enhances website fingerprinting attacks by fusing interrupt traces from different sources. Compared to previous methods that rely on a single interrupt trace, the fusion of multiple interrupt traces allows for a more comprehensive capture of the features generated when users visit websites, thereby improving the accuracy of the attack.
- Verify the effectiveness of Enhanced Attack Against Existing Defenses. In response to existing defense mechanisms, such as randomized timers and noise injection, this model proposes a novel attack method that effectively counters these defenses, significantly improving the attack success rate. This approach not only breaks through current defenses but also enhances the effectiveness of attacks when faced with such defensive strategies.
- Conduct the experiments to analyze the fusion impacts on attacks. Our work investigates the interaction between parallel attacks, assessing whether running two attacks simultaneously can enhance performance or inadvertently interfere with each other.

II. Related Work

A. Interrupt based website fingerprinting attacks

Some attackers exploit interrupt patterns generated during the graphics rendering process, extracting feature information and classifying it using machine learning models. Ma et al. [21] introduced a side-channel attack method based on graphics interrupts, using GPU and CPU graphics rendering interrupts to infer user behavior. This method encompasses both website fingerprinting and GUI application fingerprinting, achieving over 85% accuracy in website fingerprinting and over 90% accuracy in application recognition by extracting interrupt statistical features and classifying them through machine learning models. Additionally, they recorded the frequency and distribution patterns of GPU graphics processing interrupts through the operating system interface [22], extracting features for classification to identify specific website access behaviors, achieving over 85% accuracy in experiments. This demonstrates the feasibility of performing website fingerprinting without direct access to GPU resources.

Some attackers infer users' network behavior by exploiting the execution characteristics of CPU instructions. Although these instructions do not generate interrupts themselves, they provide potential fingerprint features for attackers by monitoring and responding to processor state changes and memory access behavior. Zhang et al. [14] proposed a side-channel website fingerprinting attack based on the CPU (U)MWAIT and UMONITOR instructions. This attack infers users' web browsing behavior by monitoring network interrupts within the system, without relying on high-precision timers. It utilizes interrupts triggered by network traffic to obtain fingerprinting information, achieving a high level of stealth. Experimental results show that this method achieved over 88% accuracy in website recognition without any defense mechanisms.

Another type of attack method identifies the websites a user visits by precisely analyzing the time intervals of interrupts handled by the operating system. Attackers exploit the timing characteristics during system interrupts, such as the sequence and intervals of interrupt triggers, to build a fingerprint database. Zhang et al.'s SegScope [12] leverages the x86 architecture's segment protection mechanism to probe interrupt behavior in a fine-grained manner. This attack identifies websites by capturing and analyzing the time intervals of timed interrupts. Attackers use hardware and software timers to record the timing patterns of the operating system handling interrupts. The distribution characteristics of these interrupt patterns on the timeline can be used by attackers to build a website fingerprinting database, enabling recognition when the user revisits the same website by matching the interrupt patterns. Cook et al.'s loop-counting attack [13] employs malicious code that runs loops over specific time intervals for counting, without involving any actual memory access. Since the occurrence of interrupts causes a decrease in loop count values, this method constructs website fingerprints by comparing the changes in these values. Experimental results show that even with various isolation mechanisms and noise injection, the loop-counting attack maintains high accuracy, demonstrating its strong threat to existing defense measures.

B. Defenses against interrupt-based website fingerprinting attacks

An effective defense method is to reduce the likelihood of successful attacks by disrupting or altering the regularity of system interrupts. Ma et al. [21] proposed several defenses against graphics-interrupt-based website fingerprinting attacks, including interrupt obfuscation, which disrupts interrupt patterns by adding pseudo-random interrupts; scheduling randomization, which destabilizes interrupt frequency by altering CPU and GPU task scheduling; and resource padding, which increases GPU background resource usage to mask interrupt features. In experiments, these defenses significantly reduced attack accuracy. For example, interrupt obfuscation and scheduling randomization caused a noticeable drop in website fingerprinting accuracy. However, the study also noted that these defenses incur system performance costs, potentially affecting application efficiency, and thus require a balance between security and performance in practical use.

By randomizing the order of requests or adding fake requests to the system, the interrupt signals become more chaotic, making it difficult for attackers to extract valid feature information. Zhang et al. [14] suggested that for single-instance attacks like website fingerprinting, introducing noise could increase difficulty. For example, randomizing request order or adding dummy requests creates randomness in interrupt traces. Although this approach may not entirely prevent attacks, it would require attackers to collect more traces to identify useful patterns.

To prevent attackers from using precise time features for identification, randomizing timers and introducing fake interrupt noise are common defense methods. Cook et al. proposed two main defense methods against interrupt-based side-channel attacks [13]. One defense t is Randomized Timer, which introduces randomized timers into the system to disrupt the timing characteristics of interrupt traces, making it difficult for attackers to predict and accurately analyze the timing features of interrupts. The other defense is Adding Noise with Spurious Interrupts, which involves introducing spurious interrupt traces and noise into the system to obfuscate and mask real interrupt traces. The system randomly generates some unrelated spurious interrupt traces and mixes them with real interrupt traces, making it challenging for attackers to distinguish and identify real interrupt patterns. Both defense methods aim to enhance system security by increasing the randomness and obfuscation of interrupt traces, effectively mitigating interrupt-based side-channel attacks.

III. Proposed method

A. Motivation

Among the various attacks discussed above, this paper focuses mainly on the loop count attack [13] and Segscope [12], which achieved attack accuracy of 93.7% and 92%, respectively. Regarding the two mentioned defense strategies: Randomized Timer and Adding Noise with Spurious Interrupts, the Randomized Timer introduces randomness, reducing the loop count attack accuracy to 1.8%. The addition of spurious interrupt noise disrupts the attacker's feature vector construction by inserting irrelevant interrupt information, lowering the Top-1 accuracy to 25.5% and the Top-5 accuracy to 55.9%. For Segscope, adding random noise results in a Top-1 accuracy of 1.4% and a Top-5 accuracy of 7.5%, as shown in Fig. 1.

Therefore, we try to enhance attack strategies to fight against these defense mechanisms using data fusion techniques.

B. Framework of proposed MultiInput Model enhanced attack

To enhance website fingerprinting accuracy under defensive mechanisms, we propose a multi-input attack model specifically designed for interrupt traces—the MultiInputModel in Fig. 2. We first run two different interrupt-based website fingerprinting attacks to obtain two types of website fingerprints. The data collected from the loop count attack is named *trace_fish*, a time series of length 15000, where each value represents the loop count within a fixed time. The data collected from segscope is named *trace_seg*, a time series of length 5000, where each value represents the time elapsed between consecutive interrupts. This model aims to extract features from these two distinct interrupt trace datasets to recognize fingerprint attacks. It consists of three main components: the feature extraction module, data fusion module, and classification module. Using multi-scale convolution, self-attention, and multi-head attention mechanisms, the model extracts and effectively fuses key features from interrupt traces of different lengths, generating highly distinctive feature representations. The final classifier further compresses and optimizes these features to enable accurate website recognition.

(1) The feature extraction module aims to capture unique patterns in interrupt traces. Due to the different lengths of two kinds of interrupt data, the model employs a multi-scale convolution module, which uses various kernel sizes to capture the key characteristics of each trace type. More importantly, the multi-scale convolution can process input data across different temporal scales, effectively identifying periodicity and short-term fluctuations within the interrupt traces, thereby reconstructing the system behavior patterns associated with accessing specific websites. This approach ensures that the model achieves higher temporal resolution in trace interpretation, allowing it to retain robust feature extraction capabilities. In order to further exploit the uneven characteristics of interrupted traces in time fragments and the integration of features during the data fusion phase, a self-attention layer is designed to dynamically adjust the weights of different parts of the time series, highlighting key traces of attack value while diminishing the impact of interference information. This design enables the model to focus on important features in complex environments, enhancing the accuracy of fingerprint attacks.

The input features of the model are $x_1 \in \mathbb{R}^{d_1}$ and $x_2 \in \mathbb{R}^{d_2}$, with the output $y \in \mathbb{R}^c$, where d_1 and d_2 represent the feature dimensions of the two types of input interrupt data, and c is the number of classification categories. The model processes the two inputs independently at first, using separate feature extractors to generate intermediate feature representations h_1 and h_2. Each feature extractor consists of multi-scale convolution, self-attention, and residual modules, focusing on extracting high-dimensional features that are most valuable for the attack. These modules efficiently capture features that have a critical impact on the attack from both long and short time-series data, ensuring the model can accurately identify the target website's fingerprint.

$$h_1 = \sum_{i=1}^{N} W_i^{(1)} \cdot \mathrm{ReLU}(x_1), \quad h_2 = \sum_{j=1}^{M} W_j^{(2)} \cdot \mathrm{ReLU}(x_2) \quad (1)$$

Here, $W_i^{(1)}$ and $W_j^{(2)}$ are the weight matrices of the multi-

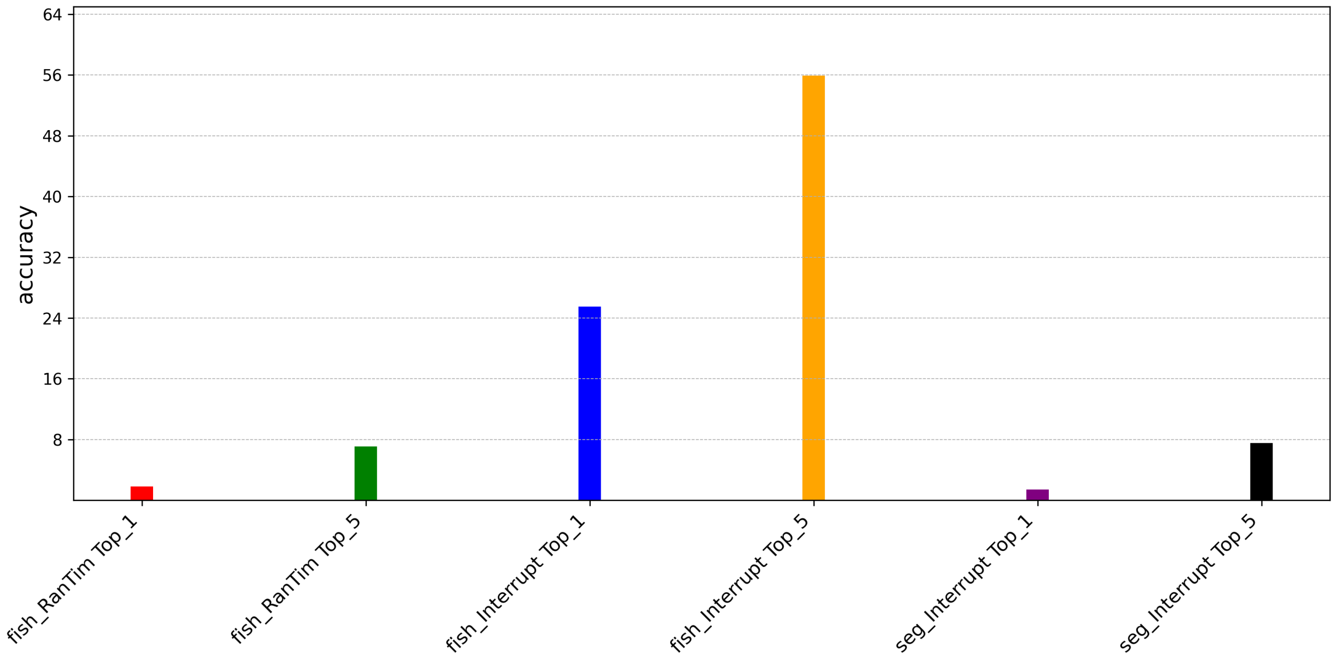

Fig. 1. Attack accuracy under defense

scale convolutional layers, with $*$ representing the convolution operation, and N and M denoting the number of convolution kernels at specific scales. $\text{ReLU}(x)$ is the nonlinear activation function. Each feature extraction stage also includes residual blocks to enhance the model's ability to express deep-level features, thereby better capturing the features that are critical to the attack.

To fuse these two sets of features, we first use the self-attention mechanism to calculate the weight matrices A_1 and A_2 for each feature. The formula for calculating the weights is as follows:

$$A_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right), \quad i = 1, 2 \tag{2}$$

Here, $Q_i = W_q h_i$, $K_i = W_k h_i$, $V_i = W_v h_i$ represent the query, key, and value matrices, respectively. d_k is the scaling factor used to stabilize the attention computation process. Next, the self-attention feature representation is obtained through a weighted sum. After applying the self-attention mechanism, these feature representations can more precisely capture the information that has a significant impact on the attack outcome, ensuring that the model effectively focuses on the features most valuable for the attack. This, in turn, further enhances the accuracy and success rate of the attack.

$$h_i^{\text{att}} = A_i V_i, \quad i = 1, 2 \tag{3}$$

(2)In the data fusion stage, the multi-head attention mechanism efficiently fuses the features of different input traces, distributing the feature weights of the fused long sequence traces through multiple attention heads to avoid losing any specific type of feature. Additionally, the model introduces a learnable balance parameter α, which dynamically adjusts the contribution ratio of the two types of traces, ensuring that important information is effectively retained under strong interference conditions. This interrupt data fusion method significantly improves the model's stability against noise and interference.

During the fusion phase, the model introduces a learnable parameter α to adjust the weights of the two feature representations, resulting in the fused feature:

$$h_{\text{fused}} = \alpha \cdot h_1^{\text{att}} + (1 - \alpha) \cdot h_2^{\text{att}} \tag{4}$$

Here, α dynamically adjusts the contribution ratio of the input features, ensuring that the model can adaptively focus on the most distinguishing features.The balance parameter α is automatically trained via backpropagation with gradient descent, initialized at 0.5 and bounded in [0,1] through sigmoid activation. The optimization process follows Adam's update rule as specified in Table III. Finally, the fused features are further processed through the multi-head attention mechanism. The multi-head attention mechanism allows the model to focus on different feature subspaces from multiple ”heads,” thereby enhancing feature representation and improving the attack accuracy.

(3)In the classification stage, the model further compresses and refines the features before utilizing a classifier composed of fully connected layers, ReLU activation, and dropout for identification. This design fully leverages the interactive features of interrupted traces and deeply mines the fused features, significantly improving the success rate of attacks

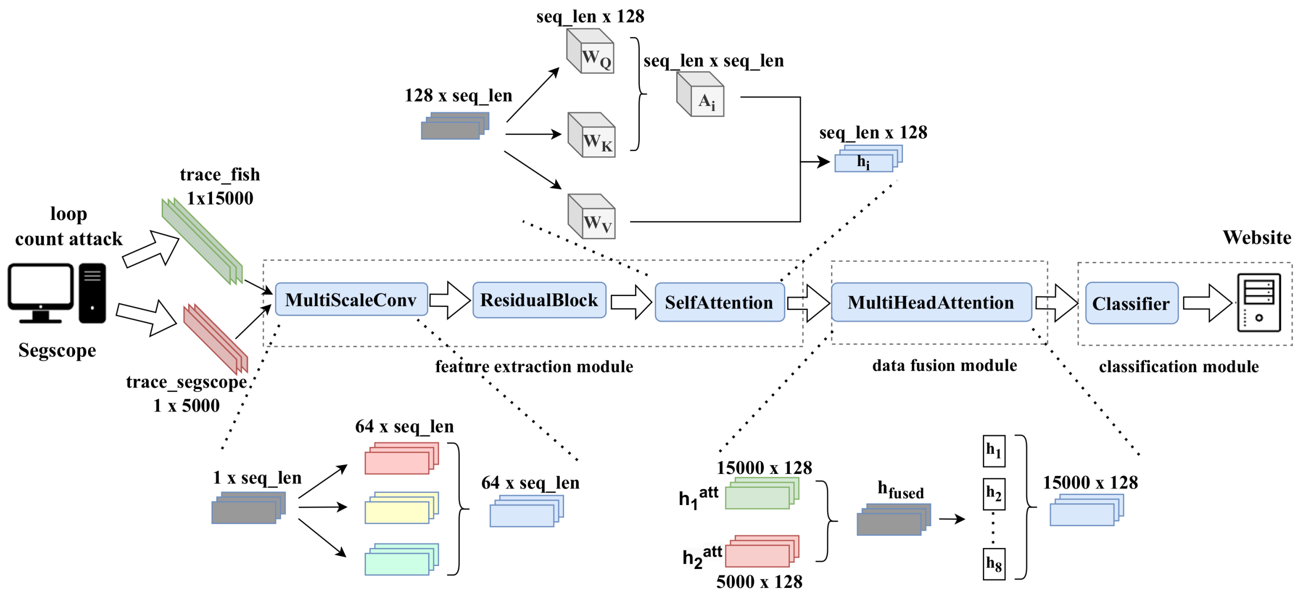

Fig. 2. MultiInputModel: trace fusion and classification flowchart

under defensive conditions. Overall, the model structure is designed to achieve effective fingerprint identification even in the absence of defensive mechanisms, thereby improving the robustness and reliability of interrupted trace fingerprint attacks.

IV. Results and analysis

A. Experiments configuration

In our experiments, for the purpose of evaluating the model's performance, we assume that the attacker has prior knowledge of the complete set of websites the victim might visit. This controlled scenario allows us to assess the model's capability in distinguishing visits to 100 different websites under ideal conditions. Some software and hardware configurations used in the experiment are shown in Table I.

Since data fusion is required, we use different methods to collecting different interrupt traces (shown in Table II). For each method, we collect 100 traces from each of the often-used 100 websites, building two datasets with 10,000 traces each. The browser's cache is not cleared before visiting each website, reflecting typical user behavior.

Table III lists the training parameters and their values, including batch size, learning rate, optimizer, epochs, loss function, and regularization. These parameters were determined experimentally to achieve the best performance in our controlled test environment.

B. Model Performance Analysis

We collect interrupt trace data from different sources (trace_count, trace_seg in Table III) and then use data fusion techniques to combine these data, aiming to enhance the accuracy and robustness of the attack. The results are shown in the Fig. 3; after applying the fusion mechanism, there is a noticeable increase in accuracy, as shown in Figure 3. The attack accuracy has increased from 93.7% to 94.9%, and the Top-5 accuracy has also risen to 99.1%.

Experimental results show that by merging interrupt traces from different sources, the classifier can capture a more comprehensive set of features related to the target website visits, thereby improving the overall attack accuracy. When using a single type of interrupt trace alone, there may be some loss or incompleteness of feature information. The data fusion model compensates for these shortcomings, allowing the classifier to achieve higher precision and reliability in website identification.

When we apply the defense methods proposed in Section 3.1, the attack still achieves favorable results, as shown in

TABLE I
Browser and Hardware Configuration Information

Component	Configuration
OS	Ubuntu 22.04.1
Browser	Chrome Version 126.0.6478.126 (Official Build) (64-bit)
CPU	Intel(R) Core (TM) i7-9750H
HZ	2.60GHz

TABLE II
Parameter Configuration for Two Different Interrupt Data for Website Fingerprint

Parameter	Configuration
trace_count	Loop count attack
trace_count_test	Loop count attack
trace_seg	Segscope
trace_random_time	Using Randomized Timer defense in loop count attacks
trace_interrupt	Using Adding Noise with Spurious Interrupts defense in loop count attacks
trace_count_seg	Data collected during the parallel execution of loop count attack and Segscope

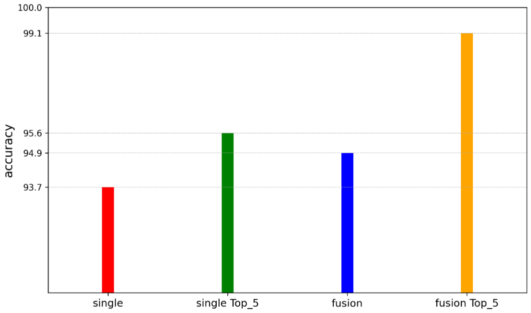

Fig. 3. The classification accuracy of the classifier after data fusion.

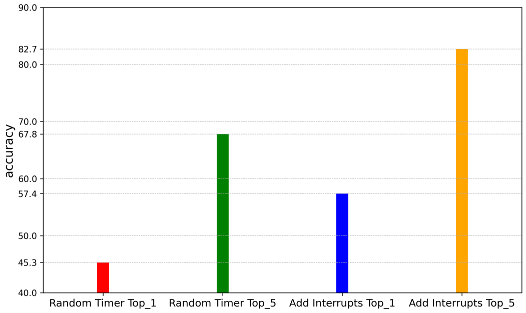

Fig. 4. Accuracy of fused data after different defenses

Fig. 4. Specifically, when using the Randomized Timer defense method, the Top-1 accuracy is 45.3% and the Top-5 accuracy is 67.8%. When using the Adding Noise with Spurious Interrupts method, the Top-1 accuracy reaches 57.4% and the Top-5 accuracy reaches 82.7%.

C. Feasibility Validation of Parallel Attacks

To ensure that the collected data would not be significantly affected in the event of parallel execution, we implement the feasibility validation of parallel attacks. Through this verification, the accuracy and consistency of data can be guaranteed in the process of attack data fusion.

First, we run the loop count attack alone and record the interrupt traces from two runs, labeled *trace_count* and *trace_count_test*. The purpose of comparing these two traces is to establish a baseline for subsequent data comparison.

TABLE III
HYPERPARAMETERS AND TRAINING CONFIGURATION OF MULTIINPUTMODEL

Parameter	Value
Batch Size	64
Learning Rate	Initial learning rate: 0.001, decayed by 0.1 every 10 epochs
Optimizer	Adam, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$
Epochs	500
Loss Function	Cross-Entropy Loss
Regularization Method	L2 regularization (0.0001) and Dropout (0.5) to reduce overfitting

TABLE IV
DIFFERENCES BETWEEN *trace_count* AND DIFFERENT DATA

Data	MSE	MAE	Max Difference	Euclidean Distance
trace_count	0.071	0.20	0.81	18.77
trace_count_test	0.073	0.20	0.84	19.01
count_with_seg	0.074	0.21	0.89	19.11
trace_interrupt	0.082	0.21	0.96	20.17
trace_random_time	0.177	0.37	0.99	29.71

Next, we run the loop count attack in parallel with another attack (Segscope) and collect the interrupt trace of the loop count attack, labeled *count_with_seg*. Although we also collected Segscope's trace, it is not used in this verification experiment. This step is designed to verify whether parallel execution affects the data collection of the loop count attack.

To further validate the impact of parallel attacks on data collection, we introduce two defense mechanisms that interfere with the loop count attack's behavior. In the first defense mechanism, we introduce random time variations to disrupt the attack execution, recording the corresponding interrupt trace as *trace_random_time*. In the second defense mechanism, we modify the interrupt handling process and record the resulting trace as *trace_interrupt*. These defense mechanisms aim to disrupt the attack process, providing a comparison to confirm that parallel execution does not introduce similarly significant variations.

To quantitatively assess the impact of parallel execution on the loop count attack, we use numerical metrics such as mean square error (MSE), mean absolute error (MAE), maximum difference, and Euclidean distance to calculate the differences between *trace_count* (baseline) and other traces (*trace_count_test*, *count_with_seg*, *trace_random_time*, and *trace_interrupt*). These metrics give us a comprehensive view of the similarity or divergence between the traces under different conditions.

The numerical analysis results are summarized in Table IV, showing the differences between *trace_count* and other traces across various metrics:

From these data, we can make the following observations:

(1)The MSE between *trace_count* and *trace_count_test* is 0.073, almost identical to the self-comparison value of 0.071. This indicates that the loop count attack produces very similar trace data across different runs under the same conditions, validating the baseline consistency of the experiment.

(2)The MSE between *trace_count* and *count_with_seg* is 0.074, only slightly higher than the difference between separate runs (0.073). This minimal increase shows that even when the loop count attack is run in parallel with another attack, there is little interference or noise introduced into the data collection, and parallel execution does not significantly affect data consistency.

(3)The defense mechanisms were introduced to further verify that the impact of parallel execution is minor. As expected, the differences for *trace_interrupt* and *trace_random_time* are significantly larger than those for the parallel attack, particularly *trace_random_time* with an MSE of 0.177. This confirms that the defense mechanisms effectively disrupt the loop count attack's data collection process, and the stronger the defense effect, the greater the difference between the traces. In contrast, the small difference in *count_with_seg* indicates that parallel execution has limited impact on data consistency. By comparing the larger variations caused by the defense mechanisms to the smaller differences from parallel execution, we can further verify that parallel attacks do not introduce significant discrepancies in the data collection of the loop count attack.

V. Conclusion

This study enhances interrupt-based attack models by introducing data fusion techniques, improving both accuracy and stability. Unlike previous studies focused on single-interrupt modeling, we propose the first quantitative model for multi-source interrupt feature fusion, expanding the vulnerability space of time-series-based side-channel attacks. Experiments show that this fusion mechanism captures more comprehensive features, overcoming existing defense countermeasures. However, two main constraints remain: first, the model assumes synchronized acquisition of interrupt traces, which may introduce synchronization overhead; second, the current implementation processes 14999-length sequences at 37fps, limiting real-time deployment on low-power devices. Future research will focus on developing more effective defense mechanisms against increasingly complex attacks.

References

[1] A. Shusterman, A. Agarwal, S. O'Connell, D. Genkin, Y. Oren, and Y. Yarom, "{Prime+ Probe} 1,{JavaScript} 0: Overcoming browser-based {Side-Channel} defenses," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2863–2880, 2021.

[2] A. Shusterman, Z. Avraham, E. Croitoru, Y. Haskal, L. Kang, D. Levi, Y. Meltser, P. Mittal, Y. Oren, and Y. Yarom, "Website fingerprinting through the cache occupancy channel and its real world practicality," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2042–2060, 2020.

[3] D. Genkin, L. Pachmanov, E. Tromer, and Y. Yarom, "Drive-by key-extraction cache attacks from portable code," in *Applied Cryptography and Network Security: 16th International Conference, ACNS 2018, Leuven, Belgium, July 2-4, 2018, Proceedings 16*, pp. 83–102, Springer, 2018.

[4] H. Naghibijouybari, A. Neupane, Z. Qian, and N. Abu-Ghazaleh, "Rendered insecure: Gpu side channel attacks are practical," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 2139–2153, 2018.

[5] Y. Wang, H. Xu, Z. Guo, Z. Qin, and K. Ren, "Snwf: website fingerprinting attack by ensembling the snapshot of deep learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1214–1226, 2022.

[6] M. Shen, K. Ji, Z. Gao, Q. Li, L. Zhu, and K. Xu, "Subverting website fingerprinting defenses with robust traffic representation," in *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 607–624, 2023.

[7] M. Sinha and M. Dave, "Enhanced non-defended website fingerprinting attack model using deep learning," in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, pp. 01–06, IEEE, 2024.

[8] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 1928–1943, 2018.

[9] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *23rd USENIX Security Symposium (USENIX Security 14)*, pp. 143–157, 2014.

[10] P. Liu, L. He, and Z. Li, "A survey on deep learning for website fingerprinting attacks and defenses," *IEEE Access*, vol. 11, pp. 26033–26047, 2023.

[11] A. Shusterman, L. Kang, Y. Haskal, Y. Meltser, P. Mittal, Y. Oren, and Y. Yarom, "Robust website fingerprinting through the cache occupancy channel. corr abs/1811.07153 (2018)," *arXiv preprint arXiv:1811.07153*, 2018.

[12] X. Zhang, Z. Zhang, Q. Shen, W. Wang, Y. Gao, Z. Yang, and J. Zhang, "Segscope: Probing fine-grained interrupts via architectural footprints," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 424–438, IEEE, 2024.

[13] J. Cook, J. Drean, J. Behrens, and M. Yan, "There's always a bigger fish: a clarifying analysis of a machine-learning-assisted side-channel attack," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, pp. 204–217, 2022.

[14] R. Zhang, T. Kim, D. Weber, and M. Schwarz, "({M) WAIT} for it: Bridging the gap between microarchitectural and architectural side channels," in *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 7267–7284, 2023.

[15] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," in *Computer Security–ESORICS 2016: 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part I 21*, pp. 27–46, Springer, 2016.

[16] J. Gong and T. Wang, "Zero-delay lightweight defenses against website fingerprinting," in *29th USENIX Security Symposium (USENIX Security 20)*, pp. 717–734, 2020.

[17] W. De la Cadena, A. Mitseva, J. Hiller, J. Pennekamp, S. Reuter, J. Filter, T. Engel, K. Wehrle, and A. Panchenko, "Trafficsliver: Fighting website fingerprinting attacks with traffic splitting," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1971–1985, 2020.

[18] T. Wang and I. Goldberg, "{Walkie-Talkie}: An efficient defense against passive website fingerprinting attacks," in *26th USENIX Security Symposium (USENIX Security 17)*, pp. 1375–1390, 2017.

[19] N. Mathews, J. K. Holland, S. E. Oh, M. S. Rahman, N. Hopper, and M. Wright, "Sok: A critical evaluation of efficient website fingerprinting defenses," in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 969–986, IEEE, 2023.

[20] H. Li, N. Niu, and B. Wang, "Cache shaping: An effective defense against cache-based website fingerprinting," in *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, pp. 252–263, 2022.

[21] H. Ma, J. Tian, D. Gao, and C. Jia, "Walls have ears: Eavesdropping user behaviors via graphics-interrupt-based side channel," in *Information Security: 23rd International Conference, ISC 2020, Bali, Indonesia, December 16–18, 2020, Proceedings 23*, pp. 178–195, Springer, 2020.

[22] H. Ma, J. Tian, D. Gao, and C. Jia, "On the effectiveness of using graphics interrupt as a side channel for user behavior snooping," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3257–3270, 2021.