RAGNAR: Exploring Volatile-Channel Vulnerabilities on RDMA NIC

Yunpeng Xu[*], Yuchen Fan[*], Teng Ma[†], Shuwen Deng[*‡§]
[*] *Department of Electronic Engineering, Tsinghua University,* [†] *Alibaba Group,* [‡] *Zhongguancun Laboratory*
[*] {xyp24, fanyc22}@mails.tsinghua.edu.cn, shuwend@tsinghua.edu.cn, [†] sima.mt@alibaba-inc.com

***Abstract*—With the surge in data computation, Remote Direct Memory Access (RDMA) becomes crucial to offering low-latency and high-throughput communication for data centers, but it faces new security threats. This paper presents RAGNAR, a comprehensive suite of hardware-contention-based volatile-channel attacks leveraging the under-explored security vulnerabilities in RDMA hardware. Through comprehensive microbenchmark reverse engineering, we analyze RDMA NICs at multiple granularity levels and then construct covert-channel attacks, achieving 3.2x the bandwidth of state-of-the-art RDMA-targeted attacks on CX-5. We apply side-channel attacks on real-world distributed databases and disaggregated memory, where we successfully fingerprint operations and recover sensitive address data with 95.6% accuracy.**

I. INTRODUCTION

With the boom of data computation, the data center has prevailingly used Remote Direct Memory Access (RDMA) [31] for low latency and high throughput [41]. RDMA allows kernel bypass and zero-copy, crucial for large-scale and real-time data processing [8], [20].

However, recent studies have shown that RDMA also opens up new attack vectors [13], [17], [34], [37]. These security threats can significantly compromise the data security of RDMA-related systems.

Given the extensive attack surface, we ❶ *provide a systematic abstraction and analysis of RDMA security*, categorizing them from the perspectives of HW/SW, RDMA-targeted/related, granularity, and attack types. While several studies [17], [18], [22], [33], [34], [37] cover various aspects of RDMA security, significant gaps remain. We propose RAGNAR[1], which addresses these uncharted security concerns by exploring the potential of RDMA-targeted contention-based volatile-channel[2] covert-channel and side-channel attacks.

To better understand RDMA hardware, we ❷ *comprehensively reverse engineer RDMA NICs (RNICs) with four different levels of traffic granularity*. We analyze hardware resource contention between flows across traffic classes, memory regions (MRs), and address offsets within the same MR, and spot new hardware vulnerabilities.

We then ❸ *build covert-channel attacks given the reverse-engineered results, achieving 3.2× bandwidth compared with state-of-the-art RDMA covert channels on the same setup*. Our attacks achieve a data rate of 84.3 Kbps on CX-6 and 63.6 Kbps on CX-5, outperforming PYTHIA's covert channel (20 Kbps on CX-5) [37] and not being mitigated by state-of-the-art RDMA isolation defenses [22].

Furthermore, we ❹ *perform side-channel attacks on distributed database [23] and disaggregated memory [39]* to demonstrate the real scenario impact of RAGNAR. We fingerprint different shuffle/join application patterns and target a key-value store application, obtaining information about the victim's access address with 95.6% accuracy.

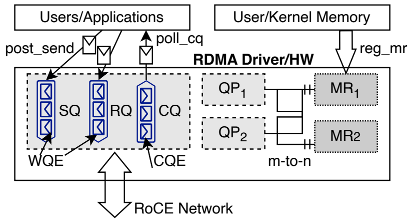


Fig. 1: Data Structures and Architecture of RDMA

Our main contributions are as follows:

- Propose granularity levels and analyze the stealthiness of RDMA-targeted HW attacks and mitigation with such metrics.
- Comprehensively reverse-engineer RDMA hardware from four different traffic granularity, especially reconstructing information on traffic priority and address offset.
- Construct 3 different granularity-level covert-channel attacks achieving 3.2× bandwidth compared with state-of-the-art attack.
- Build 2 side-channel attacks on real-world applications including distributed database and disaggregated memory, achieving up to accuracy of 95.6% on recovering victim address.

The code used in this paper will be released under an open-source license at https://github.com/THU-HAS/Ragnar.

II. BACKGROUND, RELATED WORK AND ATTACK ANALYSIS

A. Remote Direct Memory Access (RDMA)

RDMA is a network technology enabling direct memory access between physical servers, bypassing the OS [4], [7], [14], [30], [31], [35]. For a connection, each side generates queue pairs (QPs) for request queuing, registers memory region (MR) for access control, and exchanges MR addresses, as shown in Figure 1. Once connected, both sides post work queue entries (WQEs) and poll completion queue entries (CQEs) for message exchange.

B. Side-Channel and Covert-Channel Attacks

Side-channel attacks leverage indirect information of the architecture and hardware leakage to extract secrets [15], [16], [28]. *Covert-channel attacks* are where the sender and receiver establish covert communications through indirect information channels.

These channels can be categorized into *persistent channels* and *volatile channels* [42]. Persistent channels utilize state changes, e.g., the cache state attacks [1], [3], [5], [11], [28]; volatile channels utilize resource sharing on the fly, e.g., exploiting port contention in the execution engine [2], or contend network-on-chip (NoC) [9], [27].

C. Security Issues on RDMA

RDMA introduces variant security issues. We categorize these issues from two perspectives: SW/HW, RDMA-targeted/related.

- **RDMA-targeted/related SW issues.** Software implementation issues by REDMARK [33], and one-sided non-auditability [32].

[§] Shuwen Deng is the corresponding author.

[1] Ragnar Lodbrok is a legendary Norse hero known for his multiple conflicts and raids, symbolizing intense warfare and struggle. His many battles represent significant interference and opposition, mirroring the volatile contention that happened in RDMA attacks.

[2] Volatile channels refer to the cases that the sender and receiver share the resources on the fly, which maps to the hardware contention scenario previous RDMA attacks do not explore and we will introduce more in Section II.

TABLE I: Comparisons with prior works on RDMA-targeted HW attacks. Attack Types and Granularity Levels are discussed in II-D.

	Type	Grain	Defended	Channel	Steal.
Zhang [43]	P	II	🛡 [22]	-	Medium
Kong [18]	P	II	🛡 [22]	-	Medium
Husky [17]	P	II	🛡 [22]	-	Medium
Kim [13]	S	I	-	Volatile	Low
Pythia [37]	C+S	IV	🛡[3]	Persistent	High
Ragnar	C+S	I/II/III/IV	-	Volatile	High

- **RDMA-targeted HW issues.** Hardware vulnerabilities of RNICs on performance isolation [43], [18], [17], and persistent covert-channel issues with onboard cache by Pythia [37].
- **RDMA-related HW issues.** RNIC exacerbates vulnerabilities together with other components in RDMA-enabled systems, e.g., DRAM, DDIO, by ThrowHammer [36] and NetCat [19].

When we focus on RDMA-targeted vulnerabilities, we find limitations of the existing works on the hardware side, where we propose new hardware vulnerabilities and build new volatile-channel attacks.

D. Analysis and Comparison on RDMA-Targeted HW Vulnerabilities

We sort out works on RDMA-targeted HW security issues and compare them with our work, Ragnar, in Table I.

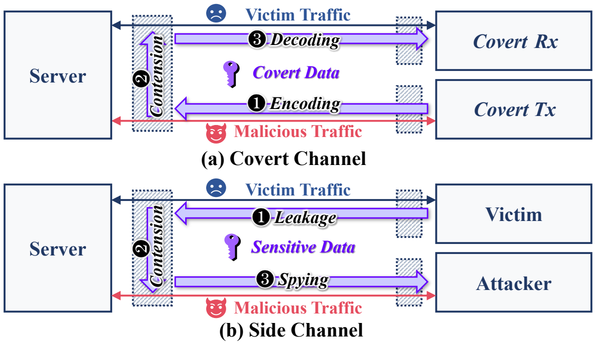
Attack Types. RDMA-Targeted HW Vulnerabilities are categorized into 3 types: 1) *Performance (P)* performs availability attacks on bandwidth or PFC; 2) *Covert (C)* performs covert-channel attacks; and 3) *Side (S)* performs side-channel attacks.

Granularity Levels. For attacks, the levels refer to the granularity of resources or parameters the attacker manipulates; for defenses, they refer to counters or metrics the defender monitors. We divide the granularity into the coarsest Grain-I to the finest Grain-IV.

- **Grain-I**. Traffic pressure. Attackers can manipulate traffic pressure to exert influence on bandwidth. For defenses, modern RNIC provides native Grain-I per-traffic-class counters and flow control to detect and defend Grain-I attacks easily.
- **Grain-II**. Main traffic patterns, e.g., traffic classes, opcodes, and message sizes. Several studies [17], [18], [43] manipulate Grain-II parameters to apply availability attacks. For defenses, Harmonic [22] provides Grain-II counters on RDMA opcodes for performance isolation to mitigate these Grain-II attacks.
- **Grain-III**. RDMA-specific resources engaged, e.g., PDs, QPs, MRs. For attacks, variant RDMA resources can be utilized to exert influence. For defenses, Harmonic also provides RDMA-specific resource utilization counters to detect Grain-III attacks.
- **Grain-IV**. Addresses and other detailed parameters of traffic patterns. If an attacker keeps accessing different addresses with such parameters, as done by Pythia [37], typical Grain-I-to-III defenses cannot detect the existence of such an attacker.

As discussed above, Modern RNIC provides native Grain-I counters for resource isolation and detection to mitigate Grain-I attacks. Several studies [17], [18], [43] exploits Grain-II performance issues that bypass native Grain-I PFC mitigation but are defended by the consequent work Harmonic [22] providing Grain-II/III counters.

Kim [13] implements side-channel attack [13] utilizing PCIe contention, but is not fine-grained enough[4]. Pythia [37] utilizes Grain-IV address manipulation to build side-channel attacks with onboard caches. Due to fine granularity, Pythia can bypass Grain-I-to-III counters and mitigations provided by RNIC and Harmonic. However, Pythia is a cache-based attack, so general cache-attack detection and mitigation [38] can be applied to mitigate Pythia. Also, the Pythia PTE attack can be mitigated by widely-used huge pages [37].



(a) Covert Channel

(b) Side Channel

Fig. 2: Threat models of (a) covert channel and (b) side channel. Light blue boxes are hosts, and dotted boxes are RNICs. Blue and red arrows show the victim and malicious traffic, respectively. Purple arrows show the covert/sensitive data flow in contention channels.

TABLE II: Specifications of Test Environment

Host	Processor	RNIC	OS	RAM
H1	AMD EPYC 9554	CX6	Ubuntu 20.04	755GB
H2	Intel Xeon S4314	CX4,5	Ubuntu 18.04	256GB
H3	Intel Xeon P8480+	CX4-6	Ubuntu 22.04	1TB

TABLE III: Parameter Sheet of CX-4 to CX-6 Network Adapters

Feature	ConnectX-4	ConnectX-5	ConnectX-6
Speed	25Gbps	100Gbps	200Gbps
PCIe Interface	PCIe 3.0 x8	PCIe 3.0 x8	PCIe 4.0 x16

Comparison with related work. Compared with prior works, Ragnar utilizes up to Gran-IV to achieve high stealthiness and bypass state-of-the-art defense Harmonic. Compared with Pythia, Ragnar is based on the processing path in RNICs. It does not rely on cache mechanism. Volatile channel brings our attack higher stealthiness and more general applicability.

III. Threat Model and Experimental Setup

A. Threat Model

In our attack scenario, three parties are involved, as illustrated in Figure 2. The server hosts data, such as an in-memory database or key-value store. One client keeps regular traffic to access the server's in-memory data via RDMA, while the other client controls malicious traffic to perform the attack.

For (a) covert channels, the covert *Tx* ❶ sets up malicious traffic to encode covert data, and ❷ transfers the encoded data to observable effects in the victim traffic, which ❸ is decoded by covert *Rx*.

For (b) side channels, the victim ❶ leaks sensitive data in its traffic and ❷ influences the malicious traffic through the contention channel. ❸ The attacker sets up and observe variant malicious traffic at the same time to recover sensitive data and the access patterns of the victim.

B. Experimental Setup

We run all the tests and attacks across different hosts and environments, including various processors from Intel and AMD listed in Table II, as well as various network cards such as CX-4, CX-5, and CX-6 listed in Table III.

[3] Defenses are discussed in Tsai's work [37]. Using huge pages or physical addresses can mitigate PTE-based attacks, and sniffers can easily detect evicting operations.

[4] It can only steal coarse information if the GPU is running rather than reveal detailed data.

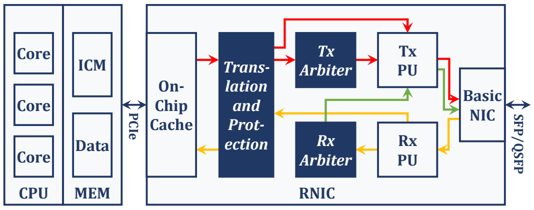


Fig. 3: Framework of RDMA System. Red arrows are the Tx flow, yellow arrows are the Rx flow, and green arrows are the replying flow. Lightboxes are microarchitectures whose existence and basic processing logic are known. Dark boxes are exponents that are black boxes to us, where we find new features by reverse engineering. [6]

IV. Reverse-Engineer RDMA NICs

The deployment of RDMA networks is contingent upon the support of RNICs. To develop RDMA-specific covert-channel and side-channel attacks, we provide some reverse-engineering results for a better understanding of RNICs in this section.

A. Architectural and Microarchitectural Analysis

Due to the black-box nature of detailed structure, we provide an architectural and microarchitectural picture of the RNIC, which are recovered from ETHTOOL [25] bps and pps counters and description of existing literature[5] and Nvidia manual [24], [26], as shown in Figure 3.

The RNIC is usually connected to the network with a Fiber Optic Interface and to the CPU/MEM with PCIe. MRs are pinned to physical memory to avoid page faults in the data path. The RNIC typically has the same functional parts as general network cards (Basic NIC) with typical L1 to L3 layer operations. Main RDMA-specific microarchitectures on the RNIC are shown in Figure 3.

In Figure 3, the red arrow indicates the Tx flow of outbound traffic. The RNIC reads the SQE from the host and arbitrates among all the SQEs. Requests are sent to the processing part according to RDMA opcodes and then handed over to the basic NIC for non-RDMA-specific procedures.

The yellow arrow indicates the Rx flow of inbound traffic. The basic NIC hands over the received and parsed message to the Rx processing pipeline.

RDMA Reads, as well as reliable service, need ACK packets, forming a reverse flow (green arrows). The reverse flowis arbitrated and processed as the outbound flow.

B. Grain-I/II Contention on Different-Priority Traffics

We identify *arbiters* that arbitrate packets of different flows within the RDMA hardware pathway. This arbitration impacts RDMA traffics in very different priorities in contention scenarios depending on parameters like opcodes, QP count, and message size.

Setup and Priority Indication. We systematically test the priority behavior of traffic during contention to evaluate the impact of different RDMA operations on competing flows. Using the `mlnx_qos` tool [26], we configure two traffic flows in ETS mode, each allocated 50% of the bandwidth. *However, we observe unbalanced bandwidth affected by priority mechanisms and hardware arbiters.* We configure traffic with varying QP numbers and message sizes and monitor changes in bits per second (bps) and packets per second (pps).

Comprehensive study on priority effects between traffics. We run a comprehensive benchmark of over 6000 parameter combinations to describe traffic performance in competitive scenarios. Analysis of the results yields a conceptual diagram (Figure 4) illustrating competition patterns under typical parameters. From Figure 4, we have the following key observations:

- The competition between RDMA Read and Write operations is complex and non-monotonic. As shown in the blue-outlined section ❶ of Figure 4, when message sizes are small, RDMA Write traffic loses over 50% of its bandwidth due to competition. However, once the write message size reaches around 512 Bytes, its bandwidth increases significantly, causing RDMA Read traffic bandwidth to drop by 30% to 80%.
- The RDMA Atomic operation exhibits a similar trend when competing with RDMA Read/Write operations, as illustrated in the orange-outlined section ❷ of Figure 4.
- In some cases, competition increases both traffic flows, resulting in total traffic exceeding 200% of the original single flow, as demonstrated in the green-outlined section ❸ of Figure 4.
- RDMA Write and reverse RDMA Read traffic with identical parameters show different competition dynamics when competing with RDMA Write traffic, highlighted in the yellow-outlined section ❹ of Figure 4.

From the analysis of the above observations, we can derive the following findings:

Key Finding 1: Non-monotonic bandwidth contention. When the competing write flow is small, only the medium read flow experiences a significant drop, while the bandwidth of both the small and large read flows remains unaffected. However, as the competing write flow increases, the situation is reversed.
Key Finding 2: Abnormal bandwidth increment. Contention of small RDMA Writes traffic can lead to abnormal BW increment in both traffic, which can be related to NoC activation.
Key Finding 3: Arbiter priority. The logical Tx arbiter has a higher priority than the logical Rx arbiter.

C. Grain-III/IV Contention within Memory Region

We then post the fixed pattern of requests in a single-threaded manner and involve a new metric to explore finer-grained features.

Unit latency increase. The latency Lat_{total} from completing the `ibv_post_send` to polling a completion with `ibv_poll_cq` is a more precise and stable per-message observable than throughput statistics. Since Lat_{total} includes the queuing delay in the WQ before actual transmission, we assume that Lat_{total} should have a linear relationship with the queue length as $Lat_{total} = k \cdot (len_{sq} + 1) + C$. len_{sq} is the number of WQEs in front of the measured WQE. k stands for the latency increase caused by each queuing request, namely the *Unit Latency Increase (ULI)*. C is some len_{sq}-independent constant term caused by other overhead. Both theoretical analysis [7] and experiments [8] support the linearity and $C \approx 0$. So we use $ULI \approx Lat_{total}/(len_{sq} + 1)$ to characterize the contention.

[5] Only a few studies provide their original description. PYTHIA [37] proved the existence of multiple on-chip caches and reverse-engineered their parameters, Kalia et al. [12] illuminates the software-hardware interface data flows of various RDMA operations.

[6] The TxPU/RxPU and TxArbiter/RxArbiter in the figure are logically separated, but this does not mean that they are deterministically separated parts in hardware. For the convenience of discussion, we will still consider them separately later.

[7] If we consider an SQ reaching the maximum send queue size $len_{sq,max}$ in the stable traffic case, all these $len_{sq,max}$ work requests on the fly should be uniformly distributed in a Lat_{total} round trip time, where ULI should be $k = Lat_{total}/len_{sq,max} = Lat_{total}/(len_{sq} + 1)$.

[8] The linear relationship fits well with Pearson correlation coefficient = 0.9998, and C can be neglected.

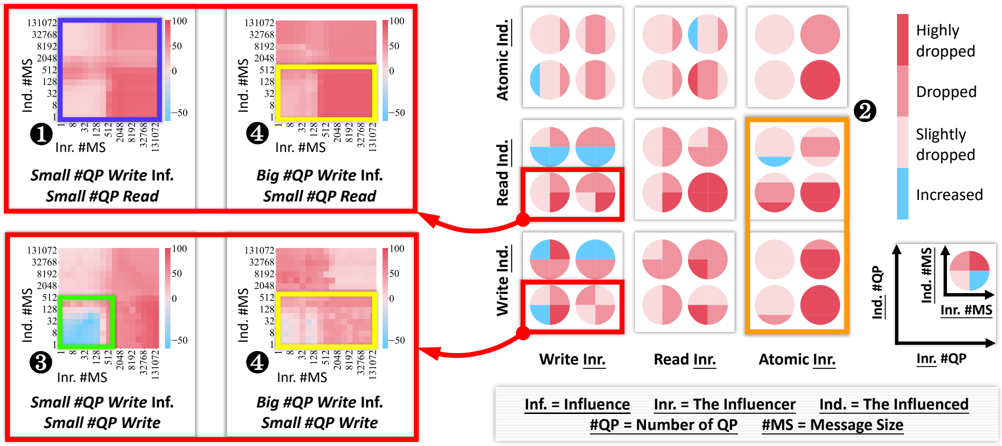


Fig. 4: Conceptual traffic priority diagram. This figure illustrates the competition-caused reduction in one party's traffic when competing with another under various conditions. For instance, the subplot of Write Inr. and Read Ind. shows a decrease in Read traffic when competing with Write traffic. The axes of each subplot represent qp number (qp_num), and the pie chart sections show results for different message sizes, as depicted in the legend. Dark red indicates no significant decrease, medium red a 50% decrease, light red a slight decrease, and blue an abnormal increase. These pie charts summarize measured data, with details expanded on the left side and key phenomena highlighted.

TABLE IV: Parameters of Grain-III/IV microbenchmark. a@MR#b means the address with offset a to the base address of the bth MR.

Operation	Msg Size	Remote Address 1	Remote Address 2
RDMA Read	1 - 1MB	0 @ MR#0	0 - 4KB @ MR#0
RDMA Read	1 - 1MB	0 @ MR#0	0 @ MR#1
RDMA Read	1 - 1MB	0 - 4KB @ MR#0	same as RA1

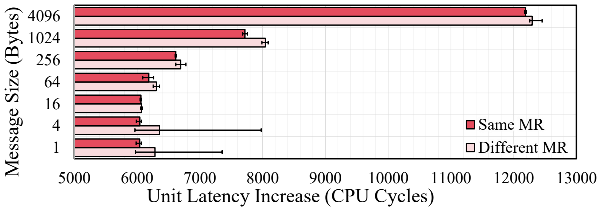


Fig. 5: ULI vs. same/different remote MRs vs. message size. Alternately accessing two addresses in the same/different remote MRs on CX-4 with RDMA Reads. Average and 10/90-percentile are shown.

Setup. We establish MRs on 2 MB huge pages, use 2 QPs, and set all resources within the same PD. We disable DDIO and bind the benchmark process to the same CPU core in the same NUMA node with the RNICs to eliminate the differences in address translation and context switching. Detailed configurations are shown in TABLE IV. [9] We perform the tests on CX-4, CX-5, and CX-6 RNICs.

With the Grain-II parameters unchanged, we focus on a more covert parameter, *remote address*. We find that the remote address of RDMA Reads can have a significant impact on the datapath contention. This impact can be reflected in ULI as the following:

- Latency distinction does not only occur when accessing different/same MR for RDMA Reads (Figure 5), but also when accessing variant addresses, with huge pages enabled, DDIO disabled, and affects of NUMA and cache excluded (Figure 6).
- The latency pattern caused by accessing different remote address offsets varies with message sizes but shows similar 2's power periodicity. As shown in Figure 6, 7 and 8, stable latency drops occur at addresses aligned with 8 Bytes under RDMA Reads. More significant drops appear at addresses multiples of 64 Bytes. An apparent periodicity at 2048 Byte intervals occurs as well.
- The absolute address offset (to the base address of the MR) and the relative address offset (between consequent RDMA Reads) have different impacts on latency, shown in Figure 6 and 8. The relative address offset effect indicates the mutual interaction among different packets due to complex mechanisms in the Translation and Protection module in Figure 3.

Key Finding 4: Offset effect. Remote address offsets affect ULI in variant 2's power periodic manners for RDMA Reads.

V. Covert-Channel Ragnar Attacks

A. Covert-Channel Attack Setup

In this section, we present three covert-channel attacks. In our threat model (Figure 2(a)), the covert Tx/Rx cannot communicate mutually but share RDMA-based service with the same server. In V-B, the clients do not need to share a common address space, while in V-C and V-D, the clients are only permitted to perform certain RDMA Reads. In both cases, they cannot communicate directly.

The approach for these covert channel attacks is illustrated as follows: The sender modifies the mode of resource X based on the covert bits, which influences the receiver's observed metric of resource Y, allowing the receiver to infer the transmitted information.

B. Inter-Traffic-Class Priority-Based Channel

In this attack, the covert Rx maintains a flow with a small bandwidth, continuously monitoring its bandwidth (resource Y). Meanwhile, the covert Tx encodes bits 1/0 using the bandwidth of another flow (resource X). Covert Rx can infer the transmitted bits by observing Y.

Figure 9 shows our results on three RNICs, using the covert channel to transmit the bitstream `11011111101010010`. The TX performs RDMA Writes with 128 Bytes (bit `1`) and 2048 Bytes bit `0`). The results clearly show distinct bandwidth for bits `1` and `0`. This attack has a meager error rate because priority-based covert channel attacks rely on traffic bandwidth.

[9] Besides, we have tried RDMA Write, different MR sizes, and different local addresses. Yet, these variations do not bring up stable and observable effects.

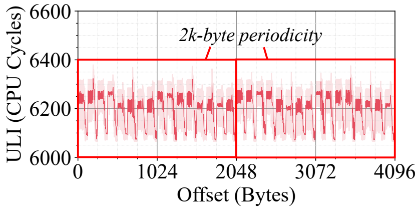


Fig. 6: ULI vs. *absolute* address offset. Alternately accessing two addresses with *64 B* RDMA Reads with the same remote MR on CX-4. Red lines and zone show the average and 10/90-percentile.

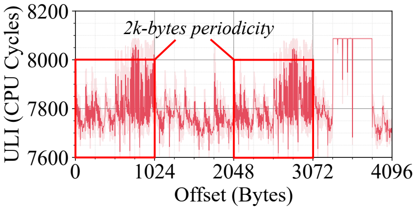


Fig. 7: ULI vs. *absolute* address offset. Alternately accessing two addresses with *1024 B* RDMA Reads with the same remote MR on CX-4. Red lines and zone show the average and 10/90-percentile.

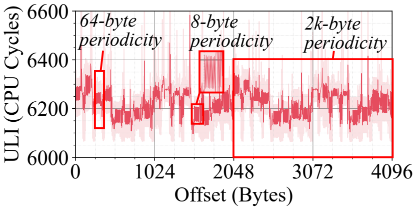


Fig. 8: ULI vs. *relative* address offset. Alternately accessing two addresses with *64 B* RDMA Reads with the same remote MR on CX-4.Red lines and zone show the average and 10/90-percentile.

TABLE V: Design features and evaluations of different granularity level covert-channels attacks on CX-4, CX-5, and CX-6.

Covert Channel	**Inter Traffic-Class**			**Intra Traffic-Class**					
				Inter MR			**Intra MR**		
Granularity Level	I+II			III			IV		
Base	Priority			RDMA resources			Offset effect		
RNIC (ConnectX)	CX-4	CX-5	CX-6	CX-4	CX-5	CX-6	CX-4	CX-5	CX-6
Bandwidth	1.0 bps	1.1 bps	1.1 bps	31.8 Kbps	63.6 Kbps	84.3 Kbps	32.2 Kbps	31.5 Kbps	81.3 Kbps
Error Rate	0.00%	0.00%	0.00%	5.92 %	3.98 %	7.59 %	6.95%	4.84%	4.08%
Effective Bandwidth	1.0 bps	1.1 bps	1.1 bps	21.5 Kbps	48.3 Kbps	51.6 Kbps	20.5 Kbps	22.7 Kbps	61.3 Kbps

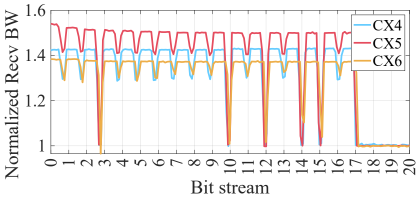


Fig. 9: Priority-based covert-channel attacks on CX-4/5/6. The significant drop means bit 0, and the slight drop means bit 1.

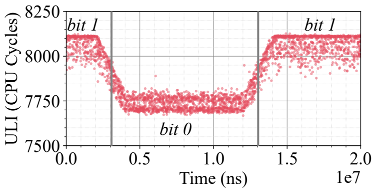


Fig. 10: Covert bits decoded from different unit latency increase, 1024 B RDMA Read, Max Send Queue Length = 256, on CX-4.

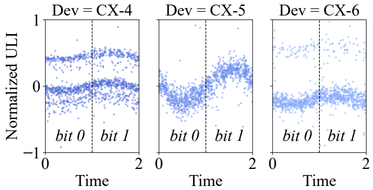


Fig. 11: Inter-MR resources-based channel on CX-4/5/6. X-axes indicate a folded period of two covert bits; Y-axes are normalized ULI.

C. Inter-MR Resource-Based Channel

We further employ Grain-III parameters, accessing the same or the different MRs (resource X) to encode the covert bits and measure the ULI (resource Y) of the background traffic.

Figure 10 demonstrates the folding ULI pattern with periodically switching covert bitstream. This ULI distinction remains stable over tens of seconds. More tests are done on CX-4, CX-5, and CX-6, utilizing 2 MB MR and 2 QPs. Under the best parameter combinations[10], the results are shown in Figure 11. As shown in TABLE V, RAGNAR's covert-channel attack on CX-6 reaches 84.3 Kbps bandwidth with 51.6 Kbps effective bandwidth.

D. Intra-MR Address-Based Channel

For better stealthiness, we manipulate Grain-IV parameters to construct an intra-MR channel. We switch address offsets (resource X) and observe ULI (resource Y) with 512 B RDMA Reads.

Under the best parameter combinations[11], the attack achieves 78.0 Kbps bandwidth on CX-6 with an error rate of 4.08 %, shown in TABLE V. Compared to the inter-MR channel, it offers higher stealthiness. This is because that encoding the covert bits brings nothing more than normal variation of access address offsets.

VI. SIDE-CHANNEL RAGNAR ATTACKS

In this section, we establish side-channel RAGNAR attacks on real-world applications.

We illustrate the threat model in Figure 2(b), where both the attacker and the victim are clients sharing an RDMA-based real-world service with the same server. We focus on privacy leakage of the access pattern rather than the data. In VI-A, we snoop database operation workloads in a distributed database. In VI-B, we reconstruct which data the victim accesses. In this model, the attacker and victim can read from a shared memory, like an open library or key-value store. Workload identification and access pattern leakage cause privacy breaches. An attacker can infer individual usage habits and expose system access hotspots in key-value stores. This leakage exacerbates the system's vulnerability. For instance, NVLEAK [40] compromise the privacy of a SQL database or key-value storage by maliciously spying on the victim's queries. Researches on Oblivious RAM indicate that curious attackers can concentrate resources on decrypting only hotspot data, analyze inter-query relations, and eventually break data confidentiality through access pattern snooping [29].

A. Fingerprint Distributed Database with Grain-II Attack

We exploit our reverse-engineering findings to fingerprint and extract operation workloads from the victim.

Specifically, we target an RDMA-based database shuffle/join application [23]. The shuffle/join method is a network-intensive operation and is widely used in distributed databases, especially suitable for

[10] Operations are 512 B Read, 64 B Read, 512 B Read respectively; max send queue sizes are 10, 6, 6 respectively.

[11] Max send queue size is set to 8. Covert bits are encoded to 0 B/255 B address offsets for CX-4 and CX-5, and 0 B/257 B address offsets for CX-6.

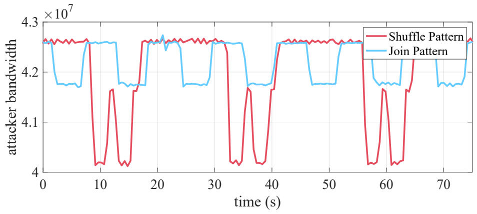


Fig. 12: Fingerprint patterns of shuffle/join.

Algorithm 1 Side-channel attack on SHUFFLE-JOIN

```
Input: $P_{Shuf/Join}$. Pattern of the Shuffle/Join operation
Output: Detected pattern P
$BW_{History} \leftarrow [\,]$
Establish RDMA connection and initialize monitoring traffic
while RDMA connection is active do
    Current time $t \leftarrow GetCurrentTime()$
    Current traffic bandwidth $BW \leftarrow GetTrafficBW()$
    Append $(t, BW)$ to $BW_{History}$
    Maintain Time Window:
    for each entry $(time, \_)$ in $BW_{History}$ do
        if $time < t - T_{window}$ then
            Remove $(time, \_)$ from $BW_{History}$
        end if
    end for
    Traffic Pattern Detection:
    P = CorrelationDetect $(BW_{History}, P_{Shuf}, P_{Join})$
    Pattern P detected if P != $P_{Null}$
    Wait for the next monitoring cycle
end while
return P
```

large datasets. We establish a side channel based on priority-based contention, and perform fingerprinting on each shuffle/join operation.

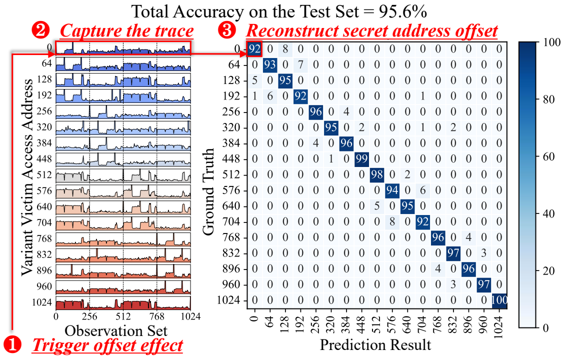
In this experiment, we monitor the behavior of a small flow maintained by the attacker (as shown in Algorithm 1). As illustrated in Figure 12, the attacker's bandwidth decreases plateau-like during the shuffle and tooth-like during the join. The observed pattern slightly deviates from the baseline under different round times and configurations, demonstrating that this side-channel attack can extract clear information about shuffle/join operations.

B. Snoop on Disaggregated Memory with Grain-IV Attack

Disaggregated memory decouples CPU and memory into independent and network-attached components, computing, and memory servers (CS/MS) [6], [10], [21]. We choose SHERMAN [39], a disaggregated memory setup with write-optimized distributed B$^+$ Tree index, to perform our side-channel attack. SHERMAN is currently implemented as a 64 B KV store. We regard it as a file index in the MS cluster. We set the shared file to be 1 KB in the remote memory, assuming the ratio of numbers of file index access to file access to be 0.01 and the file access size to be 64 B.

In this setting, the attacker and the victim are procedures on CSs to read data from shared memory in an MS. Typically, one client cannot know which address the other client accesses, while the attacker aims to do so by posting certain RDMA Reads to trigger offset effect.

Figure 13 illustrates the attack's three steps. The victim repeatedly accesses an address from the *Candidate Set* (17 candidates, 0 B to 1024 B address offset) using 64 B RDMA Reads. During ❶, the attacker performs 64 B RDMA Reads on each address in the *Observation Set* (257 samples, 0 B to 1024 B address offset) N times



(a) Capture the Trace **(b) Predict the Victim Access Address**

Fig. 13: Evaluation on the RAGNAR side-channel attack. (a) Differentiated traces were captured by the attacker under 17 variant victim access addresses. (b) ResNet18-based 17-Classifier achieves an overall 95.6% accuracy on recovering the address from a trace.

to measure ULI. In ❷, the average ULIs form a trace revealing the victim's access. For instance, if the victim accesses 0 B address offset, the trace will match the pattern in the red box shown in Figure 13(a).

Furthermore, ❸ we involve a ResNet18 classifier to recover the victim's access address from the captured trace. We train the classifier with 6720x 257-dimensional traces. It performs 17-classification on the traces and achieves an overall 95.6% accuracy on the test set, shown in Figure 13 (b). It shows that RAGNAR can easily steal the address accessed by the victim with high accuracy.

VII. DISCUSSION ON POTENTIAL MITIGATION

In this section, we discuss potential defenses.

Existing defenses. As discussed in II-D, the state-of-the-art performance isolation implementation HARMONIC [22] is not sufficient to mitigate RAGNAR's attack, since it does not take Grain-IV metrics into account and cannot eliminate the bandwidth differences completely. To the best of our knowledge, there is no direct existing mitigation to our attacks.

Hardware partitioning or adding noise. Direct mitigation involves fixing hardware features like eliminating priority races and mitigating offset effects by partitioning traffic workloads fairly according to counters from Grain-I-to-IV, which is costly and degrades performance. On the other hand, introducing sub-microsecond noise into packet latency can obscure ULI but may still leave detectable traces. Adding full noise for complete masking results in significant performance degradation.

VIII. CONCLUSION

This paper introduces RAGNAR, a comprehensive suite of RDMA-targeted volatile-channel attacks leveraging RNIC contention-based vulnerabilities at different granularity levels. Covert-channel RAGNAR attacks offer higher bandwidth and stealthiness than existing RDMA-targeted hardware attacks. Side-channel RAGNAR attacks can successfully reconstruct operations and achieve up to 95.6% accuracy in stealing secrets on real-world applications.

ACKNOWLEDGMENT

This work was generously supported by NSFC (U24A6009), National Key Research and Development Program of China under Grant (2024YFB4405402), Beijing Municipal Science and Technology Project (Nos.Z241100004224028), Beijing Natural Science Foundation (L247013), BNRist, the Disruptive Innovation Talent Cultivation Program of Tsinghua University.

References

[1] O. Acıiçmez and Ç. K. Koç, "Trace-driven cache attacks on aes (short paper)," in *Information and Communications Security: 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006. Proceedings 8*. Springer, 2006, pp. 112–121.

[2] A. C. Aldaya, B. B. Brumley, S. ul Hassan, C. P. García, and N. Tuveri, "Port contention for fun and profit," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 870–887.

[3] D. J. Bernstein, "Cache-timing attacks on aes," 2005.

[4] C. Bestler, B. Metzler, J. Garcia, and A. Romanow, "Direct Data Placement over Reliable Transports," RFC 5041, October 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5041

[5] J. Bonneau and I. Mironov, "Cache-collision timing attacks against aes," in *Cryptographic Hardware and Embedded Systems-CHES 2006: 8th International Workshop, Yokohama, Japan, October 10-13, 2006. Proceedings 8*. Springer, 2006, pp. 201–215.

[6] I. Calciu, M. T. Imran, I. Puddu, S. Kashyap, H. A. Maruf, O. Mutlu, and A. Kolli, "Rethinking software runtimes for disaggregated memory," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2021, pp. 79–92.

[7] J. Carrier and B. Aboba, "Marker PDU Aligned Framing for TCP Specification," RFC 5044, October 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5044

[8] I. Corporation, "Remote direct memory access over the converged enhanced ethernet fabric: Ibm's perspective on iwarp," *IBM Systems Journal*, vol. 41, no. 4, pp. 726–745, 2002.

[9] M. Dai, R. Paccagnella, M. Gomez-Garcia, J. McCalpin, and M. Yan, "Don't mesh around:{Side-Channel} attacks and mitigations on mesh interconnects," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2857–2874.

[10] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker, "Network requirements for resource disaggregation," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 249–264.

[11] D. Gullasch, E. Bangerter, and S. Krenn, "Cache games–bringing access-based cache attacks on aes to practice," in *2011 IEEE Symposium on Security and Privacy*. IEEE, 2011, pp. 490–505.

[12] A. Kalia, M. Kaminsky, and D. G. Andersen, "Design guidelines for high performance rdma systems," in *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, 2016, pp. 437–450.

[13] H. Kim and J. Hur, "Pcie side-channel attack on i/o device via rdma-enabled network card," in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2022, pp. 1468–1470.

[14] M. Ko, J. Hufferd, J. Chu, and S. Shah, "Transport Mappings for Remote Direct Memory Access over IP Fabrics," RFC 5045, October 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5045

[15] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*. Springer, 1999, pp. 388–397.

[16] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104–113.

[17] X. Kong, J. Chen, W. Bai, Y. Xu, M. Elhaddad, S. Raindel, J. Padhye, A. R. Lebeck, and D. Zhuo, "Understanding {RDMA} microarchitecture resources for performance isolation," in *20th USENIX Symposium on Networked Systems Design and Implementation*, 2023, pp. 31–48.

[18] X. Kong, Y. Zhu, H. Zhou, Z. Jiang, J. Ye, C. Guo, and D. Zhuo, "Collie: Finding performance anomalies in {RDMA} subsystems," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 287–305.

[19] M. Kurth, B. Gras, D. Andriesse, C. Giuffrida, H. Bos, and K. Razavi, "Netcat: Practical cache attacks from the network," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 20–38.

[20] Y. Li, Z. Wang, B. Han, P. Li, and H. Zhang, "Rdma over commodity ethernet at scale," in *Proceedings of the 2017 ACM SIGCOMM Conference*, 2017, pp. 385–398.

[21] L. Liu, W. Cao, S. Sahin, Q. Zhang, J. Bae, and Y. Wu, "Memory disaggregation: Research problems and opportunities," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1664–1673.

[22] J. Lou, X. Kong, J. Huang, W. Bai, N. S. Kim, and D. Zhuo, "Harmonic: Hardware-assisted {RDMA} performance isolation for public clouds," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 1479–1496.

[23] S. Ma, T. Ma, K. Chen, and Y. Wu, "A survey of storage systems in the rdma era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4395–4409, 2022.

[24] Mellanox Technologies, *Mellanox Adapters Programmer's Reference Manual (PRM)*, https://network.nvidia.com/files/doc-2020/ethernet-adapters-programming-manual.pdf, 2020, accessed: 2024-08-02.

[25] NVIDIA Corporation, *Ethtool: A Standard Linux Utility for Network Drivers and Hardware*, NVIDIA, Santa Clara, CA, 2024, Accessed: 2024-11-19. [Online]. Available: https://docs.nvidia.com/networking/display/mlnxofedv461000/ethtool

[26] NVIDIA Corporation, *NVIDIA MLNX_OFED Documentation v24.04-0.6.6.0*, 24th ed., NVIDIA, Santa Clara, California, USA, May 2024, available at https://docs.nvidia.com/networking/display/mlnxofedv24040660.

[27] R. Paccagnella, L. Luo, and C. W. Fletcher, "Lord of the ring (s): Side channel attacks on the {CPU}{On-Chip} ring interconnect are practical," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 645–662.

[28] C. Percival, "Cache missing for fun and profit," *BSDCan*, vol. 2005, 2005.

[29] B. Pinkas and T. Reinman, "Oblivious ram revisited," in *Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings 30*. Springer, 2010, pp. 502–519.

[30] J. Pinkerton and E. Deleganes, "DDP/RDMAP Security," RFC 5042, October 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5042

[31] R. Recio, B. Metzler, P. Culley, J. Hilland, and D. Garcia, "Remote Direct Memory Access Protocol (RDMAP)," RFC 5040, October 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5040

[32] L. Rosa, L. Foschini, and A. Corradi, "Empowering cloud computing with network acceleration: A survey," *IEEE Communications Surveys & Tutorials*, 2024.

[33] B. Rothenberger, K. Taranov, A. Perrig, and T. Hoefler, "{ReDMArk}: Bypassing {RDMA} security mechanisms," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 4277–4292.

[34] A. K. Simpson, A. Szekeres, J. Nelson, and I. Zhang, "Securing {RDMA} for {High-Performance} datacenter storage systems," in *12th USENIX Workshop on Hot Topics in Cloud Computing*, 2020.

[35] R. Stewart and M. Kojo, "Stream Control Transmission Protocol (SCTP) Direct Data Placement (DDP) Adaptation," RFC 5043, October 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5043

[36] A. Tatar, R. K. Konoth, E. Athanasopoulos, C. Giuffrida, H. Bos, and K. Razavi, "Throwhammer: Rowhammer attacks over the network and defenses," in *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, 2018, pp. 213–226.

[37] S.-Y. Tsai, M. Payer, and Y. Zhang, "Pythia: remote oracles for the masses," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 693–710.

[38] H. Wang, H. Sayadi, T. Mohsenin, L. Zhao, A. Sasan, S. Rafatirad, and H. Homayoun, "Mitigating cache-based side-channel attacks through randomization: A comprehensive system and architecture level analysis," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020, pp. 1414–1419.

[39] Q. Wang, Y. Lu, and J. Shu, "Sherman: A write-optimized distributed b+ tree index on disaggregated memory," in *Proceedings of the 2022 international conference on management of data*, 2022, pp. 1033–1048.

[40] Z. Wang, M. Taram, D. Moghimi, S. Swanson, D. Tullsen, and J. Zhao, "{NVLeak}:{Off-Chip}{Side-Channel} attacks via {Non-Volatile} memory systems," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 6771–6788.

[41] T. S. Woodall, G. M. Shipman, G. Bosilca, R. L. Graham, and A. B. Maccabe, "High performance rdma protocols in hpc," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface: 13th European PVM/MPI User's Group Meeting Bonn, Germany, September 17-20, 2006 Proceedings 13*. Springer, 2006, pp. 76–85.

[42] W. Xiong and J. Szefer, "Survey of transient execution attacks and their mitigations," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–36, 2021.

[43] Y. Zhang, J. Gu, Y. Lee, M. Chowdhury, and K. G. Shin, "Performance isolation anomalies in rdma," in *Proceedings of the Workshop on Kernel-Bypass Networks*, 2017, pp. 43–48.