Relocate-Vote: Using Sparsity Information to Exploit Ciphertext Side-Channels

Yuqin Yan[†] Wei Huang[†‡] Ilya Grishchenko[†]
Gururaj Saileshwar[†] Aastha Mehta[*] David Lie[†]

[†]University of Toronto
[‡]Seneca Polytechnic
[*]University of British Columbia

Confidential Computing

Threat model

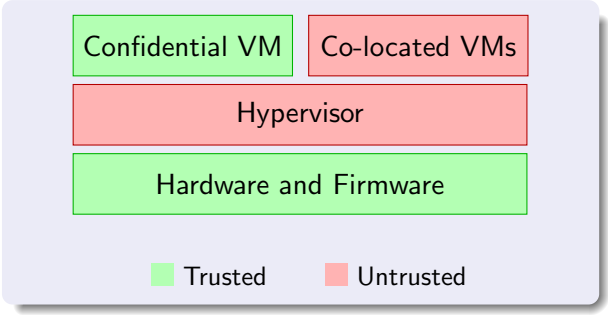
- Untrusted hypervisor
- Trusted hardware and firmware

Cloud service providers (CSPs)

aws Azure Google Cloud

Trusted hardware

AMD SEV-SNP



Confidential Computing

Threat model

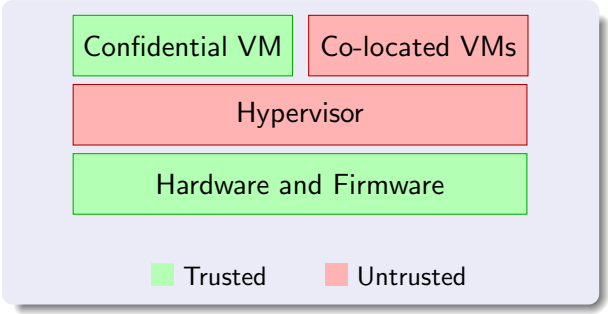
- Untrusted hypervisor
- Trusted hardware and firmware

Cloud service providers (CSPs)

aws Azure Google Cloud

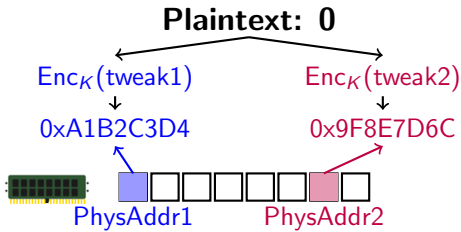
Trusted hardware

AMD SEV-SNP



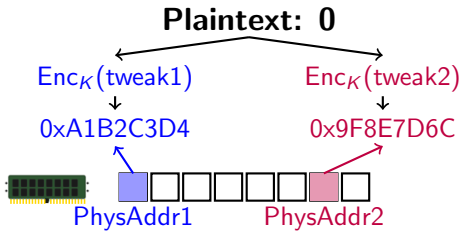
SEV-SNP: Address-dependent Deterministic Memory Encryption

- Visibility: Ciphertext is visible to the hypervisor
- Each memory location has unique encryption parameters (tweaks)
  - Same plaintext → different ciphertexts at different locations



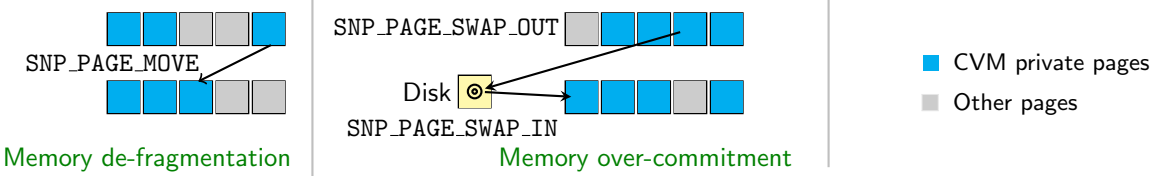
SEV-SNP: Address-dependent Deterministic Memory Encryption

- Visibility: Ciphertext is visible to the hypervisor
- Each memory location has unique encryption parameters (tweaks)
  - Same plaintext → different ciphertexts at different locations
- Determinism: At a fixed location: Same plaintext → same ciphertext
  - Ciphertext side-channels: CIPHERLEAKS [SEC'21], Li et al. [Oakland'22], HyperTheft [CCS'24], CipherSteal [Oakland'25]
    - ⋆ Only attacks at fixed locations
    - ⋆ Learned ciphertexts at a location are not useful for other locations

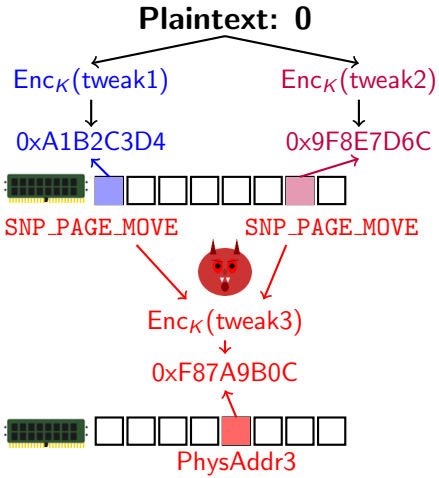


Relocate-Vote: SEV-SNP's Hardware-assisted Page Relocation

- Hardware assists in re-encrypting the relocated pages
- SEV-SNP commands support page relocation
  - `SNP_PAGE_MOVE`: Direct relocation for memory de-fragmentation
  - `SNP_PAGE_SWAP`: Moving page in/out of disk for memory over-commitment
- Controlled relocation: The hypervisor controls the destination page frame

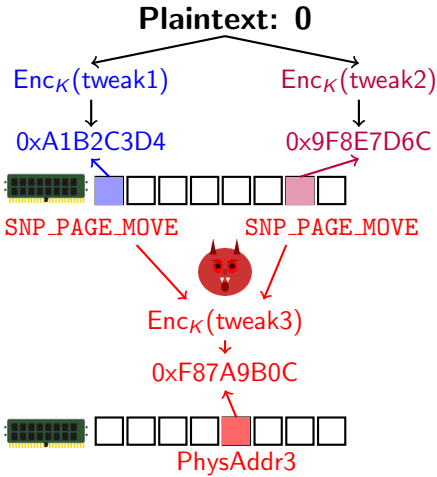


Relocate-Vote: Exploiting Relocation Mechanism



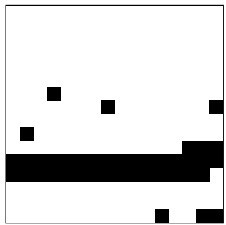
- Plaintext frequency preserved in ciphertext
- Exploit existing values across memory locations
  - Break tweaked encryption's spatial protection

Relocate-Vote: Exploiting Frequency under Tweaked Encryption



- Infer the ciphertexts of prevalent values (e.g., zero)
  - Relocate CVM's pages onto the same page frame
  - Collect the re-encrypted ciphertexts
    - ★ Vote for frequencies
  - Zero is a prevalent value in CVMs

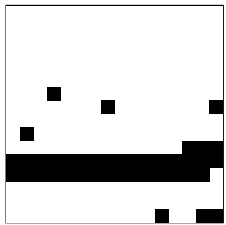
Relocate-Vote: Exploiting Frequency under Tweaked Encryption



Test a CVM's private page:
□ blocks: zero memory locations
■ blocks: non-zero memory locations

- Infer the ciphertexts of prevalent values (e.g., zero)
  - Relocate CVM's pages onto the same page frame
  - Collect the re-encrypted ciphertexts
    - ★ Vote for frequencies
  - Zero is a prevalent value in CVMs
- Test arbitrary memory locations
  - Relocate a page onto the above page frame
  - Examine if ciphertexts match the prevalent ones

Relocate-Vote: Exploiting Frequency under Tweaked Encryption

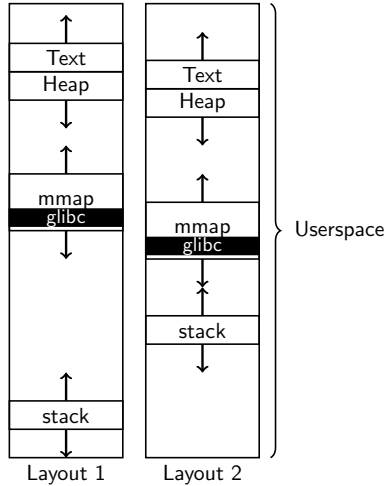


Test a CVM's private page:
- □ blocks: zero memory locations
- ■ blocks: non-zero memory locations

- Infer the ciphertexts of prevalent values (e.g., zero)
  - Relocate CVM's pages onto the same page frame
  - Collect the re-encrypted ciphertexts
    - ★ Vote for frequencies
  - Zero is a prevalent value in CVMs
- Test arbitrary memory locations
  - Relocate a page onto the above page frame
  - Examine if ciphertexts match the prevalent ones
- Leverage patterns of prevalent and non-prevalent values

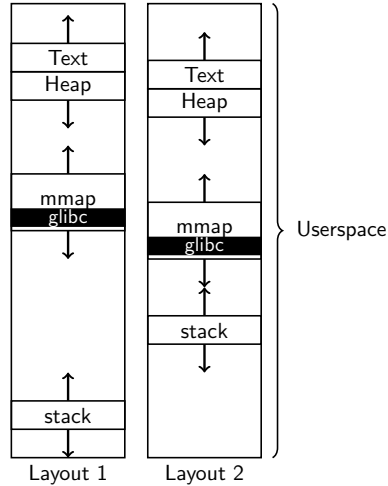
De-randomize ASLR with Relocate-Vote

- ASLR: Address Space Layout Randomization
  - Randomizes memory layout
  - Prevents predictable addresses for exploitation, such as `glibc`'s addresses for ROP.

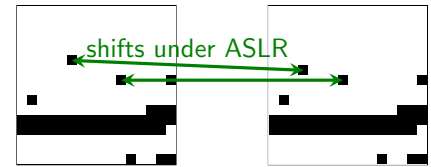


De-randomize ASLR with Relocate-Vote

- ASLR: Address Space Layout Randomization
  - Randomizes memory layout
  - Prevents predictable addresses for exploitation, such as `glibc`'s addresses for ROP.
- Our attack infers `glibc`'s location:
  - Base address (guest virtual address, GVA)
  - Queries services provided by the victim application
  - No direct code execution inside the CVM

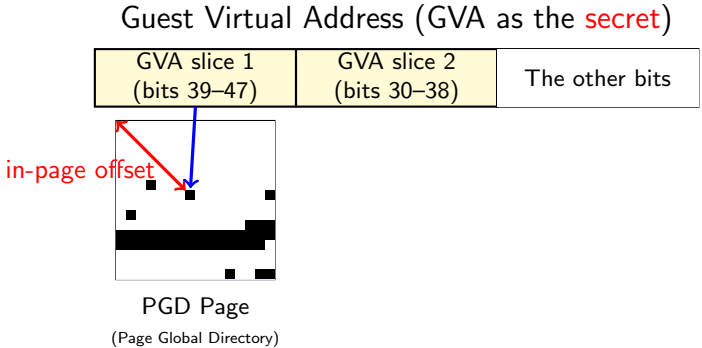


De-randomize ASLR: Sparsity in Page Tables



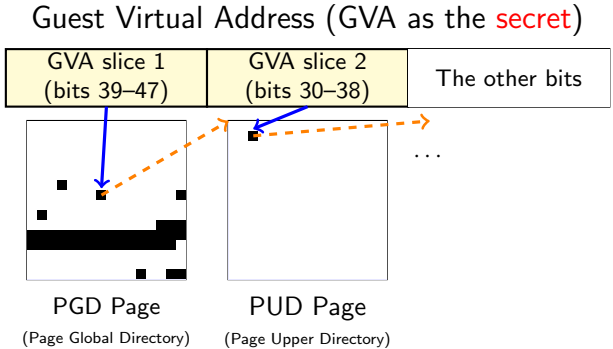
- Mapped regions (non-zeroed entries) alternate with unmapped regions (zeroed entries)
  - ASLR shifts the mapped regions $\rightarrow$ shifts the non-zero blocks

De-randomize ASLR: Sparsity in Page Tables



- Mapped regions (non-zeroed entries) alternate with unmapped regions (zeroed entries)
  - ASLR shifts the mapped regions $\rightarrow$ shifts the non-zero blocks
- GVA slices index page table entries $\rightarrow$ Offsets of entries reveal GVA slices

De-randomize ASLR: Sparsity in Page Tables



- Mapped regions (non-zeroed entries) alternate with unmapped regions (zeroed entries)
  - ASLR shifts the mapped regions → shifts the non-zero blocks
- GVA slices index page table entries → Offsets of entries reveal GVA slices
- Address translation walks through page table levels: PGD → PUD → PMD → PTE

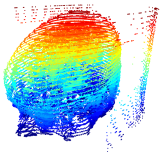
De-randomize ASLR: End-to-end Attack and Results

- Offline phase
  - Profiles on attacker's own CVM
  - Prepares the classifiers for identifying page table pages and producing offsets
- Online phase
  - Triggers the service accessing `glibc` symbols provided by the victim CVM
  - Applies classifiers to the accessed pages

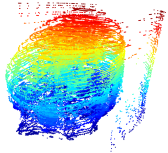
De-randomize ASLR: End-to-end Attack and Results

- Offline phase
  - Profiles on attacker's own CVM
  - Prepares the classifiers for identifying page table pages and producing offsets
- Online phase
  - Triggers the service accessing `glibc` symbols provided by the victim CVM
  - Applies classifiers to the accessed pages
- **Results:** 8,388,608 possibilities reduced to 35-104 on average
  - 5 server applications (nginx, apache, memcached, redis, and mysql)
  - 10 different memory layouts per application

Other Attack Scenarios with Relocate-Vote



Original

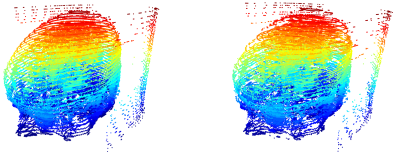


Recovered

Extract 3D object constructed from CT scanning

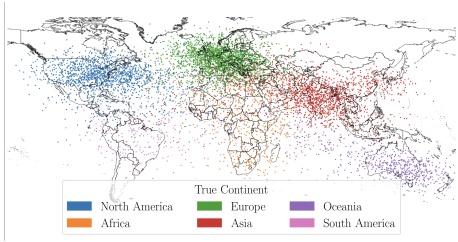
- OpenVDB: Library for representing and processing sparse 3D voxel grids
  - Extracts voxel distributions from the victim's construction and read-only traversal operations

Other Attack Scenarios with Relocate-Vote



Original Recovered

Extract 3D object constructed from CT scanning



Predicted coordinates of the places from activation patterns

- OpenVDB: Library for representing and processing sparse 3D voxel grids
  - Extracts voxel distributions from the victim's construction and read-only traversal operations
- Sparse LLM: LLM with ReLU activations
  - Decodes geographical information about the processed prompts from activation patterns
- More details in our paper

Mitigation

- Mitigating sparsity leakage at the software level
  - ↓ Compatibility and performance
- Restricting the hypervisor's relocation ability
  - **A new guest policy in SEV-SNP**: `PAGE_SWAP_DISABLE`
  - ↓ Hypervisor's ability of memory resources management
- Enforcing ciphertext-hiding in SEV-SNP instances
  - ↓ Specific hardware required
  - ↓ Limited availability of supported SEV-SNP instances in major CSPs

Conclusion

- Relocate (page relocation) and Vote (frequency analysis)
  - Leaks ciphertexts of prevalent values
  - Tests arbitrary memory locations
- Recovered sparsity information in CVMs
  - ASLR, OpenVDB, sparse LLM
  - Various operations: lookup, construction, traversal
- Exacerbated implications for ciphertext side channels
  - Broadens attack scenarios—no ciphertext updates or collisions
  - Cross-location ciphertext knowledge transfer
- Mitigation is required
  - Security brief **AMD-SB-3021** by AMD