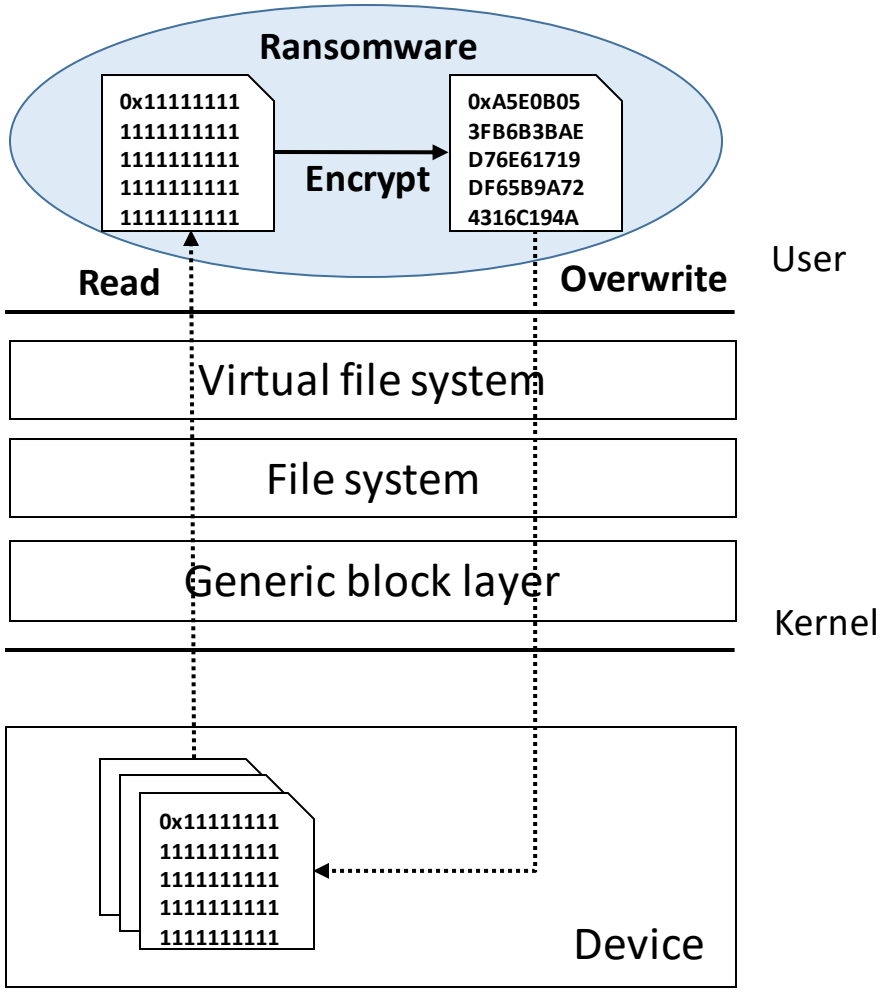
blkCtrace : Lightweight Block I/O Layer Content Tracing on Linux

Donggyu Park[1], Youngjae Kim[1], Sungyong Park[1], Junghee Lee[2]

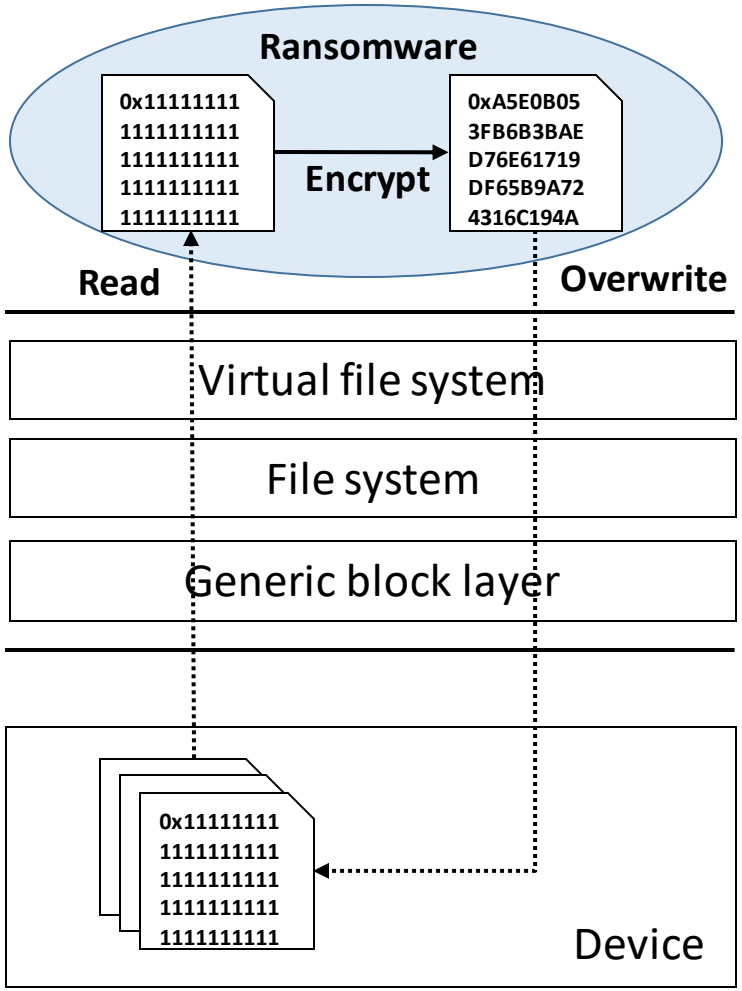
[1]Sogang University, Seoul, Republic of Korea
[2]University of Texas at San Antonio, TX USA

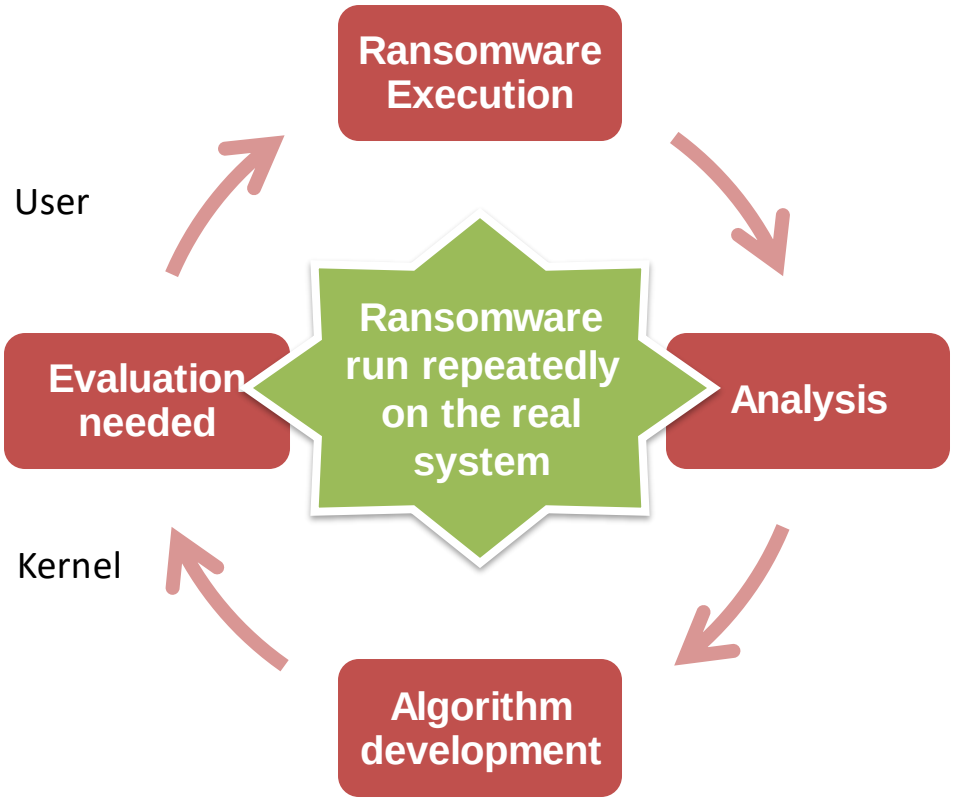


Motivation

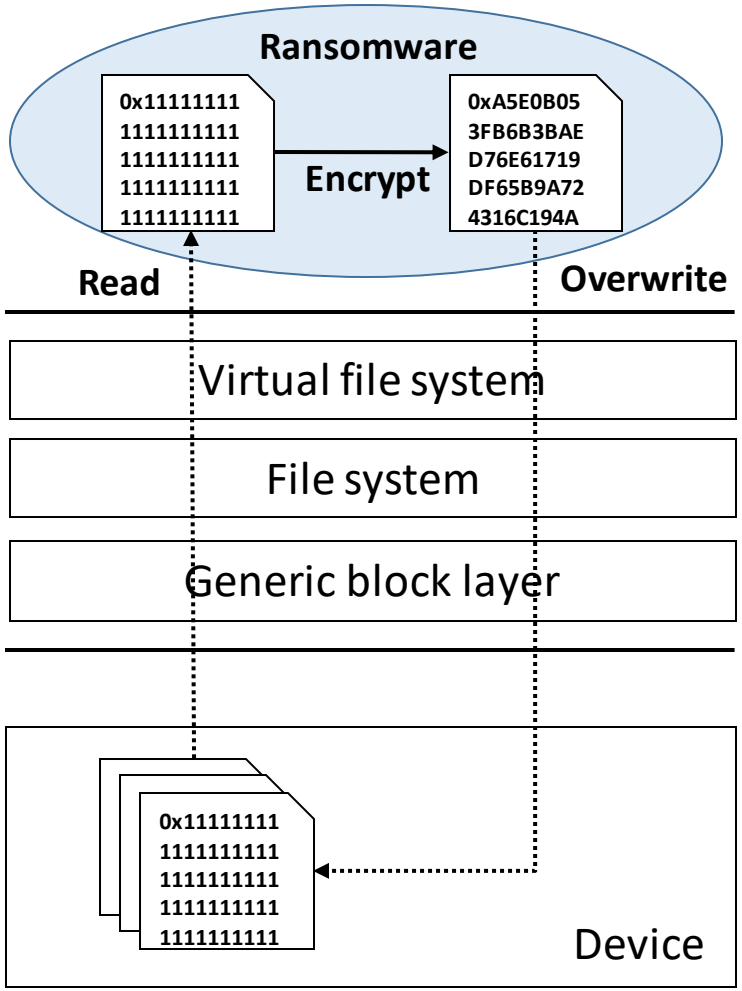

Ransomware
0x11111111
1111111111
1111111111
1111111111
1111111111
Encrypt
0xA5E0B05
3FB6B3BAE
D76E61719
DF65B9A72
4316C194A
Read
Overwrite
User
Virtual file system
File system
Generic block layer
Kernel
0x11111111
1111111111
1111111111
1111111111
1111111111
Device

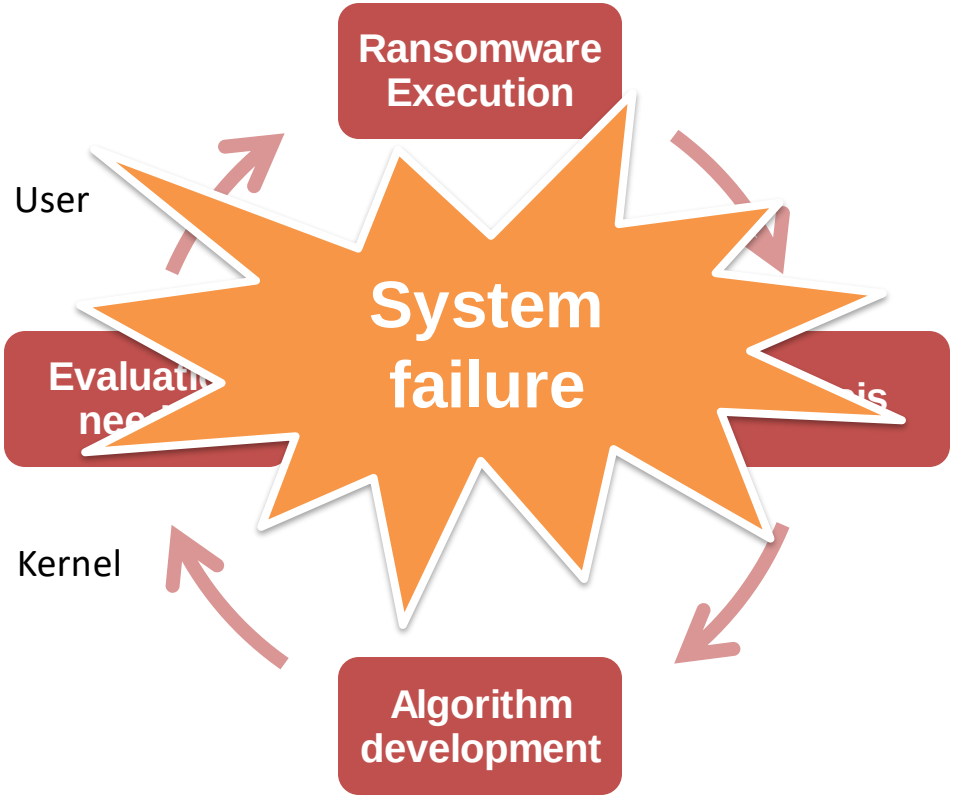
Motivation


Ransomware
0x11111111
1111111111
1111111111
1111111111
1111111111
Encrypt
0xA5E0B05
3FB6B3BAE
D76E61719
DF65B9A72
4316C194A
Read
Overwrite
Virtual file system
File system
Generic block layer
0x11111111
1111111111
1111111111
1111111111
1111111111
Device

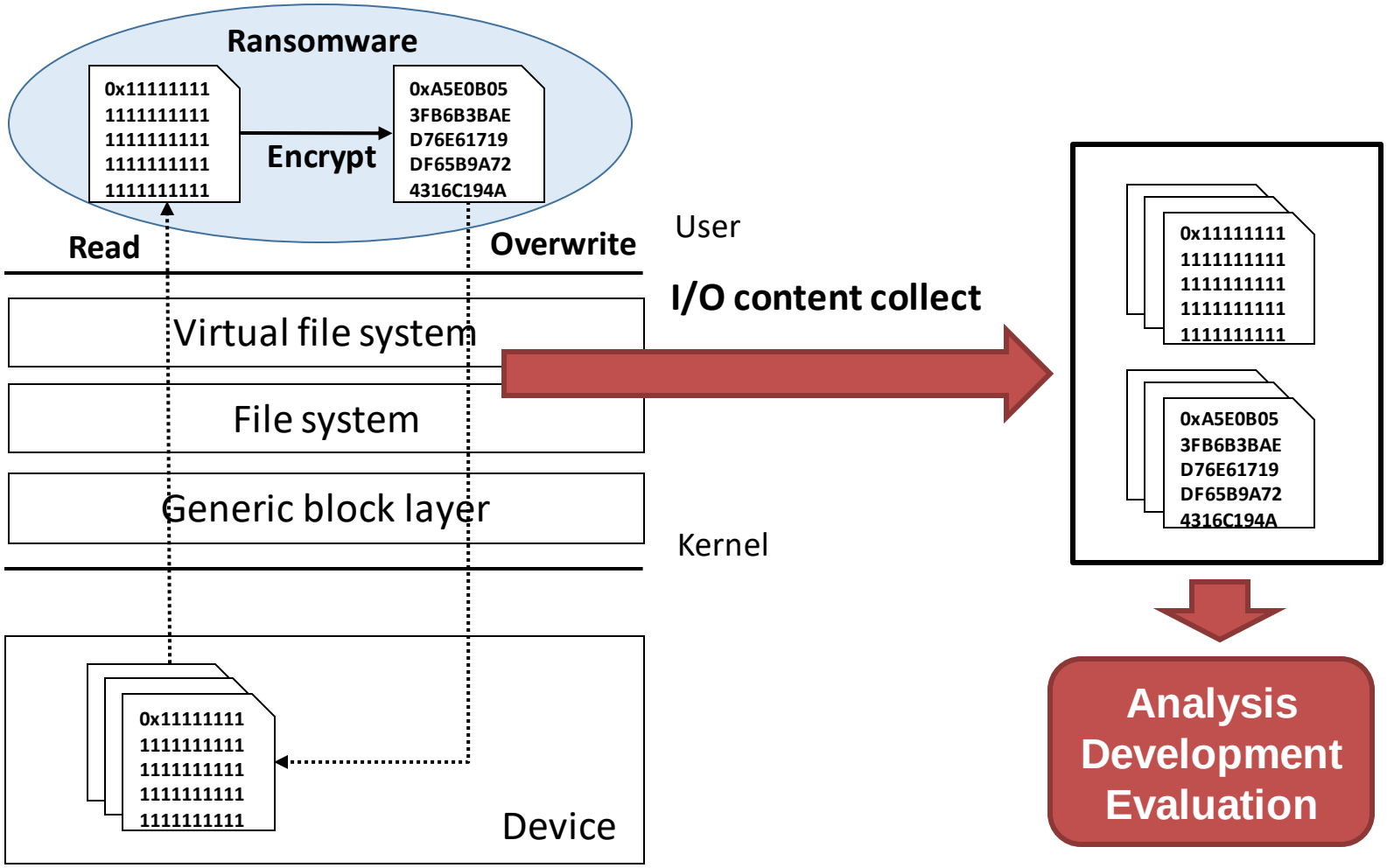

User
Kernel
Ransomware Execution
Analysis
Algorithm development
Evaluation needed
Ransomware run repeatedly on the real system

Motivation

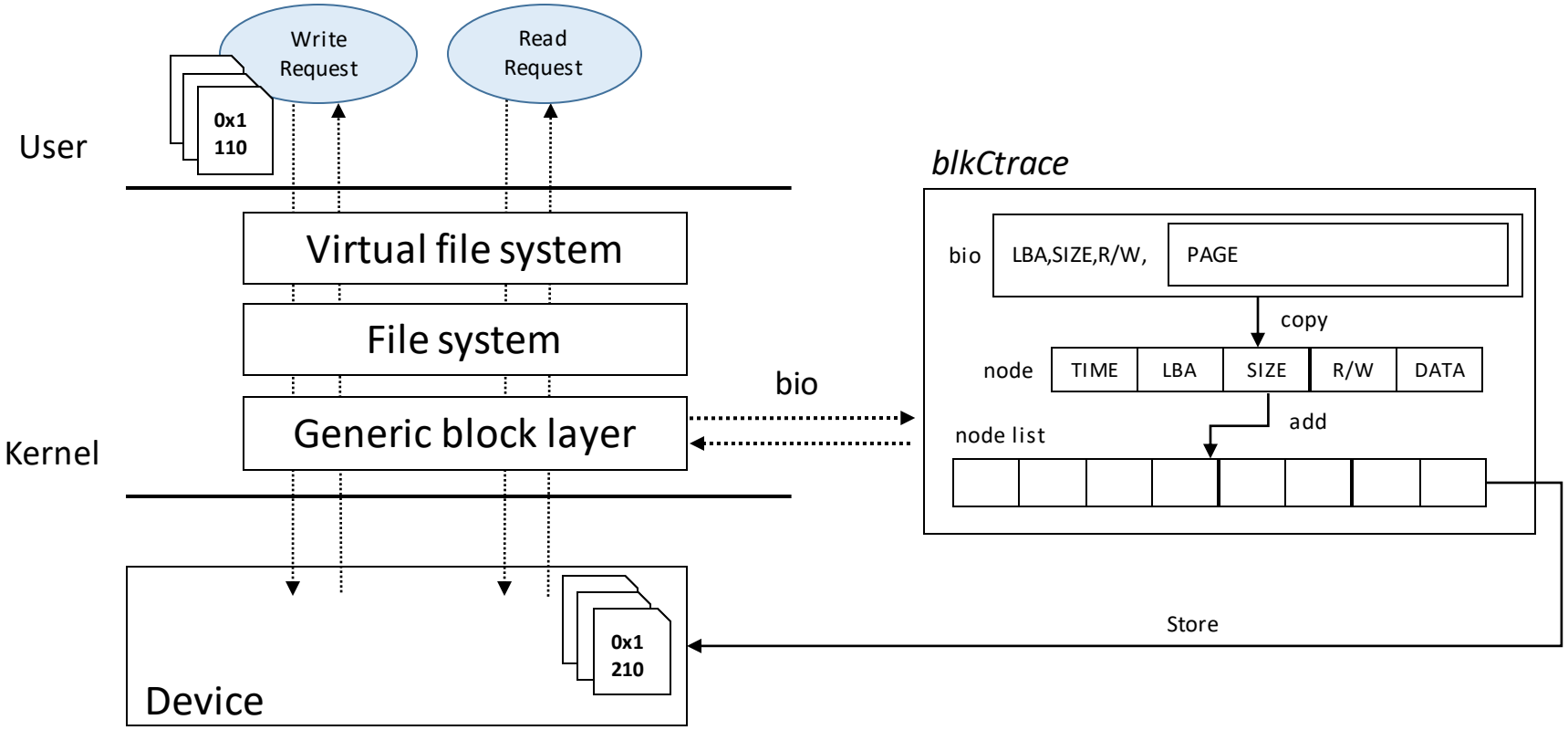

Ransomware
0x11111111
1111111111
1111111111
1111111111
1111111111
Encrypt
0xA5E0B05
3FB6B3BAE
D76E61719
DF65B9A72
4316C194A
Read
Overwrite
Virtual file system
File system
Generic block layer
0x11111111
1111111111
1111111111
1111111111
1111111111
Device


Ransomware Execution
User
System failure
Kernel
Algorithm development

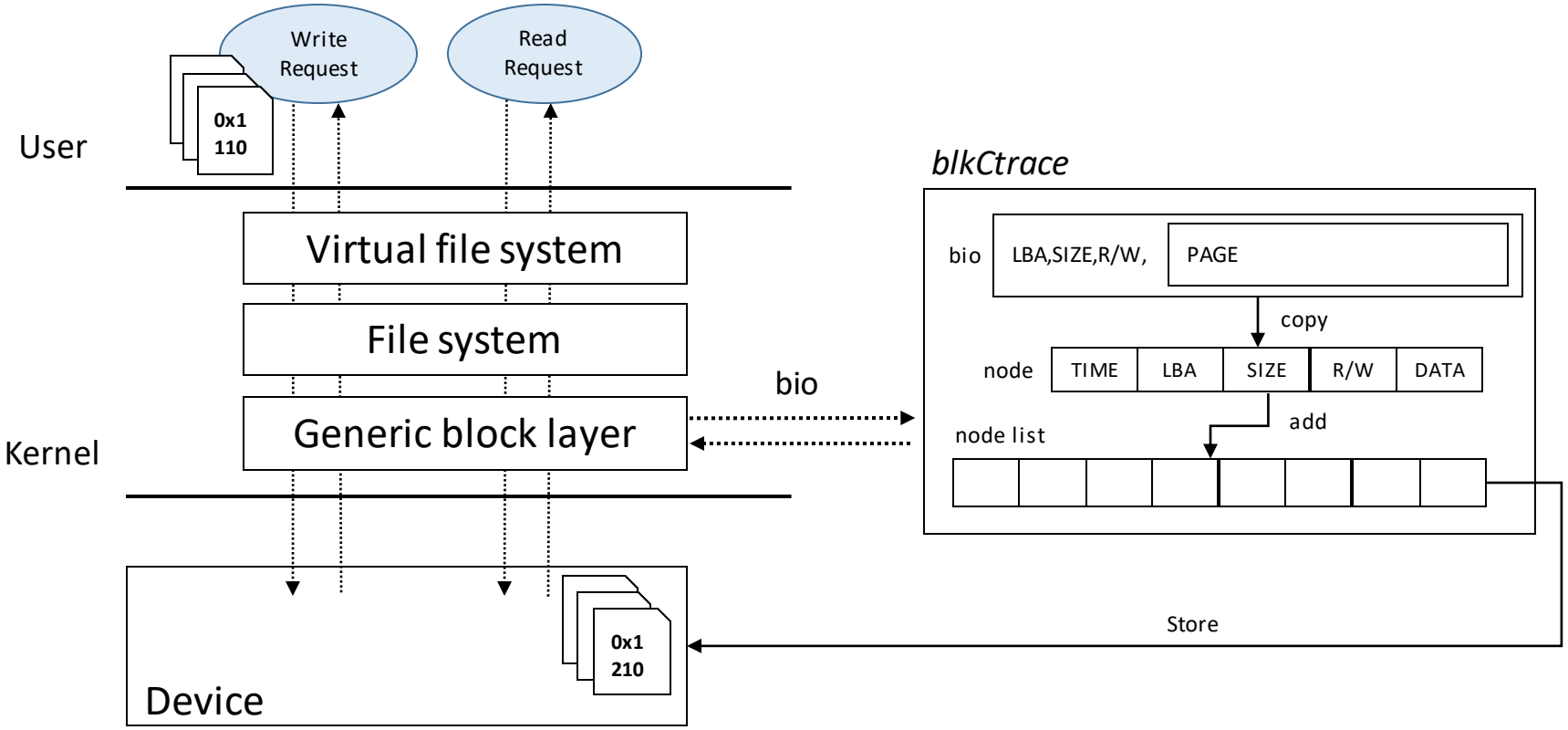
Motivation


Ransomware
0x11111111
1111111111
1111111111
1111111111
1111111111
Encrypt
0xA5E0B05
3FB6B3BAE
D76E61719
DF65B9A72
4316C194A
Read
Overwrite
User
I/O content collect
Virtual file system
File system
Generic block layer
Kernel
0x11111111
1111111111
1111111111
1111111111
1111111111
0xA5E0B05
3FB6B3BAE
D76E61719
DF65B9A72
4316C194A
Analysis
Development
Evaluation
0x11111111
1111111111
1111111111
1111111111
1111111111
Device

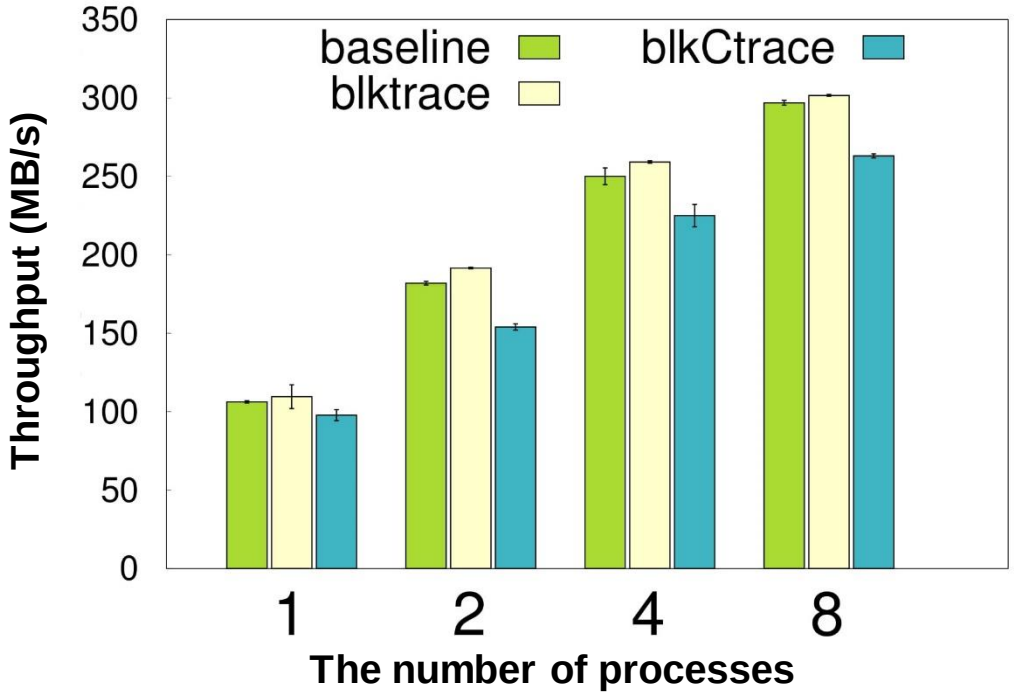
Overview


Write Request
Read Request
0x1 110
User
Virtual file system
File system
Generic block layer
Kernel
bio
blkCtrace
bio
LBA,SIZE,R/W,
PAGE
copy
node
TIME
LBA
SIZE
R/W
DATA
add
node list
Store
0x1 210
Device

Overview


Write Request
Read Request
0x1 110
User
Virtual file system
File system
Generic block layer
Kernel
bio
blkCtrace
bio
LBA,SIZE,R/W,
PAGE
copy
node
TIME
LBA
SIZE
R/W
DATA
add
node list
Store
0x1 210
Device

Preliminary Result



- We used the *fio benchmark* to measure the performance overhead of *blkCtrace* by varying the number of processes for a total of 1GB of *random read/write(50:50) workload* to an SSD.
- The overhead of *blkCtrace* is an average 11.2% compared to the baseline.

Future Work

- Tracing I/O content requires a large amount of memory space due to a vast amount of data to be collected, which may cause performance degradation.

- **Suggestion**

1. **Kernel bypass** : The collected data are directly stored in a file in the kernel without transferring it to user space.
2. **Overlapping** : Capturing I/O content in memory and writing it in a file can be overlapped.
3. **Idle time** : When the I/O arrival rate is low enough, the collected data on the kernel memory can be flushed to a file.