Hardware-Based Techniques for Memory Safety
Who am I

• 럼 럼 (Yeongpil Cho)
  ▪ B.S. POSTECH, EE
  ▪ Ph.D. Seoul National University, ECE
  ▪ Before Soongsil University, Software
  ▪ Now Hanyang University, CS

Applications

Operating Systems

Secure Operating Systems

Secure Applications

Secure Applications

ARM TrustZone

NDSS’19

Security’19

TOPS’19

DAC’17

NDSS’17

TMC’20

ATC’16
Bounds comparison

- **Intel MPX** (Memory Protection eXtensions)
  - Pointer-level bounds comparison
  - Hardware-assisted per-pointer bounds managements and checks
    - Trie-structured bounds table
    - four bounds registers
  - Recent work
    - BOGO: buy spatial memory safety, get temporal memory safety (almost) free, ASPLOS 2019

```c
void init() {
    char A[10];
    char B[20];
    bnd0 = bndmk A, A+10
    bndstx A, bnd0
    ...
    strcpy (A, B, 10);
}

void strcpy (char *dest, char *src) {
    bndldx bnd0, dest
    bndldx bnd1, src
    while (*src != \0) {
        bndcl bnd0, dest
        bndcu bnd0, dest
        bndcl bnd1, src
        bndcu bnd1, src
        *dest++ = *src++;
    }
}
```
Tag comparison

• **ARM MTE (Memory Tagging Extension)**
  - Pointers are allowed to access the memory with the matching tag
  - Hardware-assisted efficient tag management
  - Recent work
    - Memory Tagging and how it improves C/C++ memory safety, Report

```c
void init() {
    char A[10];
    char B[20];
    irg A, A
    for (i=0; i<10; i++)
        stg A[i];
    irg B, B
    for (i=0; i<10; i++)
        stg A[i];
    ...
    strcpy (A, B, 10);)

void strcpy (char *dest, char *src) {
    while (*src != '\0') {
        *dest++ = *src++;
    }
}
```
Pointer protection

- **MAC (Message Authentication Code)**
  - [Image 1]
  - [Image 2]
  - [Image 3]

- **ARM PA (Pointer Authentication)**
  - Attaches PAC (Pointer Authentication Code) to a pointer
  - Recent work
    - PAC it up: Towards Pointer Integrity using ARM Pointer Authentication

- **Diagram**
  - Secret key
  - Message
  - MAC
  - PAC
  - LR, SP
  - Push
  - BLR
  - Pop
  - Autia
  - Ret

- **Code Snippet**

```assembly
pacia lr, sp  } prologue
push lr
...
blr ...  } body
...
pop lr
autia lr, sp  } epilogue
ret lr
```