
# Jasmin: high-assurance high-speed cryptography

Miguel Quaresma  Santiago Arranz Olmos

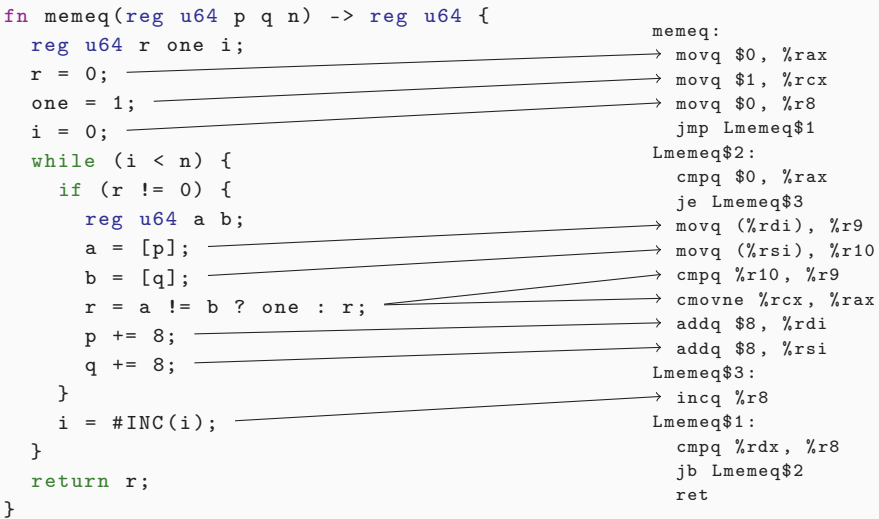
September 4, 2024

Max Planck Institute for Security and Privacy

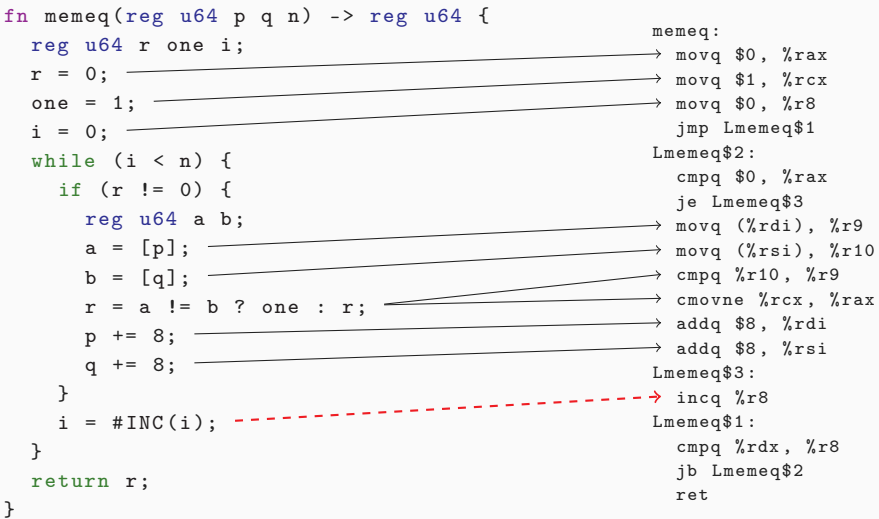
# Efficient, correct, safe, and secure

```
fn memeq(reg u64 p q n) -> reg u64 {
  reg u64 r one i;
  r = 0;
  one = 1;
  i = 0;
  while (i < n) {
    if (r != 0) {
      reg u64 a b;
      a = [p];
      b = [q];
      r = a != b ? one : r;
      p += 8;
      q += 8;
    }
    i = #INC(i);
  }
  return r;
}
```

# Efficient, correct, safe, and secure



# Efficient, correct, safe, and secure



# Efficient, correct, safe, and secure

```
fn memeq(reg u64 p q n) -> reg u64 {
  reg u64 r one i;
  r = 0;
  one = 1;
  i = 0;
  while (i < n) {
    if (r != 0) {
      reg u64 a b;
      a = [p];
      b = [q];
      r = a != b ? one : r;
      p += 8;
      q += 8;
    }
    i = #INC(i);
  }
  return r;
}
```

```
memeq:
  movq $0, %rax
  movq $1, %rcx
  movq $0, %r8
  jmp Lmemeq$1
Lmemeq$2:
  cmpq $0, %rax
  je Lmemeq$3
  movq (%rdi), %r9
  movq (%rsi), %r10
  cmpq %r10, %r9
  cmovne %rcx, %rax
  addq $8, %rdi
  addq $8, %rsi
Lmemeq$3:
  incq %r8
Lmemeq$1:
  cmpq %rdx, %r8
  jb Lmemeq$2
  ret
```

# More online


**Jasmin:** `github.com/jasmin-lang/jasmin`

**EasyCrypt specifications:** `github.com/formosa-crypto/crypto-specs`

**Libjade:** `github.com/formosa-crypto/libjade`