Binary

Quartet

0

0000

Quartet

1

0001

register: `esp`

4 quartets

opcode: `add`

0001 $\langle reg \rangle$ $\langle rval \rangle$

$reg = reg + rval$

Quartet

2

0010

register: `ebp`

4 quartets

opcode: `sub`

0010 $\langle reg \rangle$ $\langle rval \rangle$

$reg = reg - rval$

Quartet

3

0011

register: `eip`

4 quartets

opcode: `xor`

0011 $\langle reg \rangle$ $\langle rval \rangle$

$reg = reg \oplus rval$

Quartet

4

0100

register: `eax`

2 quartets

opcode: `jmp`

0100 $\langle ra \rangle$

ra must be a reg or addr

Quartet

5

0101

register: ebx

2 quartets

opcode: jgz

0101 $\langle ra \rangle$ $\langle rval \rangle$

ra must be a reg or addr

Quartet

6

0110

register: `ecx`

2 quartets

opcode: `jne`

0110 $\langle ra \rangle$ $\langle rval \rangle$ $\langle rval \rangle$

ra must be a reg or addr

Quartet

7

0111

register: `edx`

2 quartets

opcode: `ret`

Quartet

8

1000

register: `eex`

4 quartets

opcode: `call`

1000 $\langle ra \rangle$

ra must be a reg or addr

Quartet

9

1001

register: `efx`

4 quartets

opcode: `nop`

Quartet

A

1010

opcode: `load`

1010 $\langle reg \rangle$ $\langle ra \rangle$

$reg \leftarrow ra$

ra is a reg or addr

Quartet

B

1011

opcode: `store`

1011 $\langle ra \rangle$ $\langle reg \rangle$

$ra \leftarrow reg$

ra is a reg or addr

Quartet

C

1100

type: register

1100 **r**

opcode: `push`

1100 $\langle reg \rangle$

$reg \rightarrow$ top of stack

Quartet

D

1101

type: literal

1101 **h l**

$16 \times h + l$

opcode: pop

1101 $\langle reg \rangle$

top of stack $\rightarrow reg$

Quartet

E

1110

type: address

1110 **x1 x2 x3 x4**

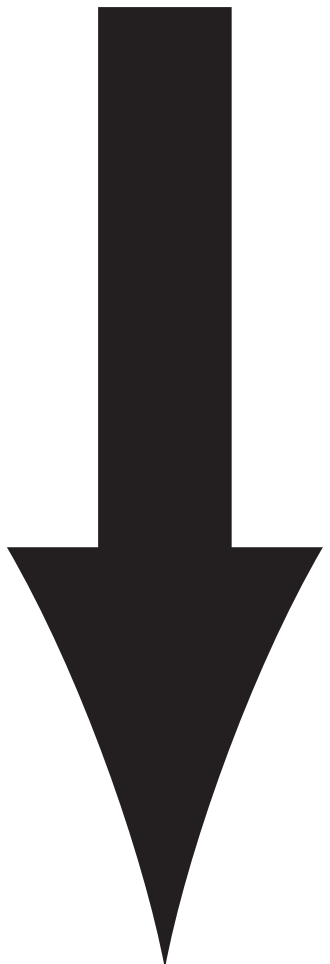
Quartet

F

1111

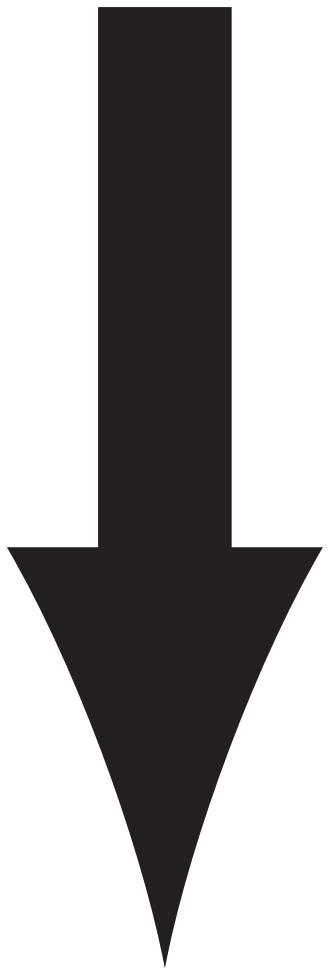
Instruction Pointer

`eip`



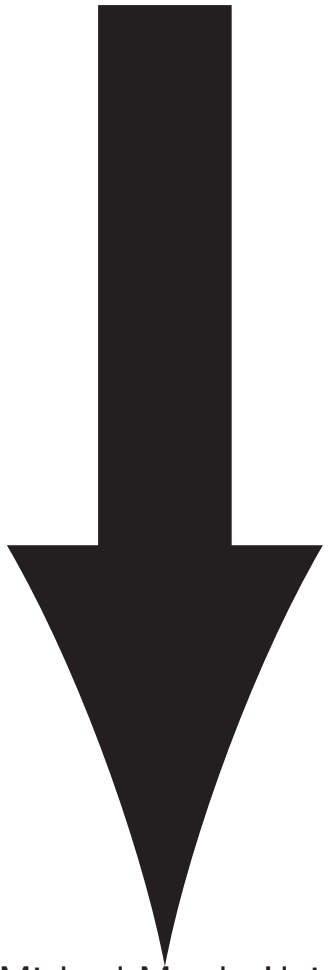
Return Pointer

ret

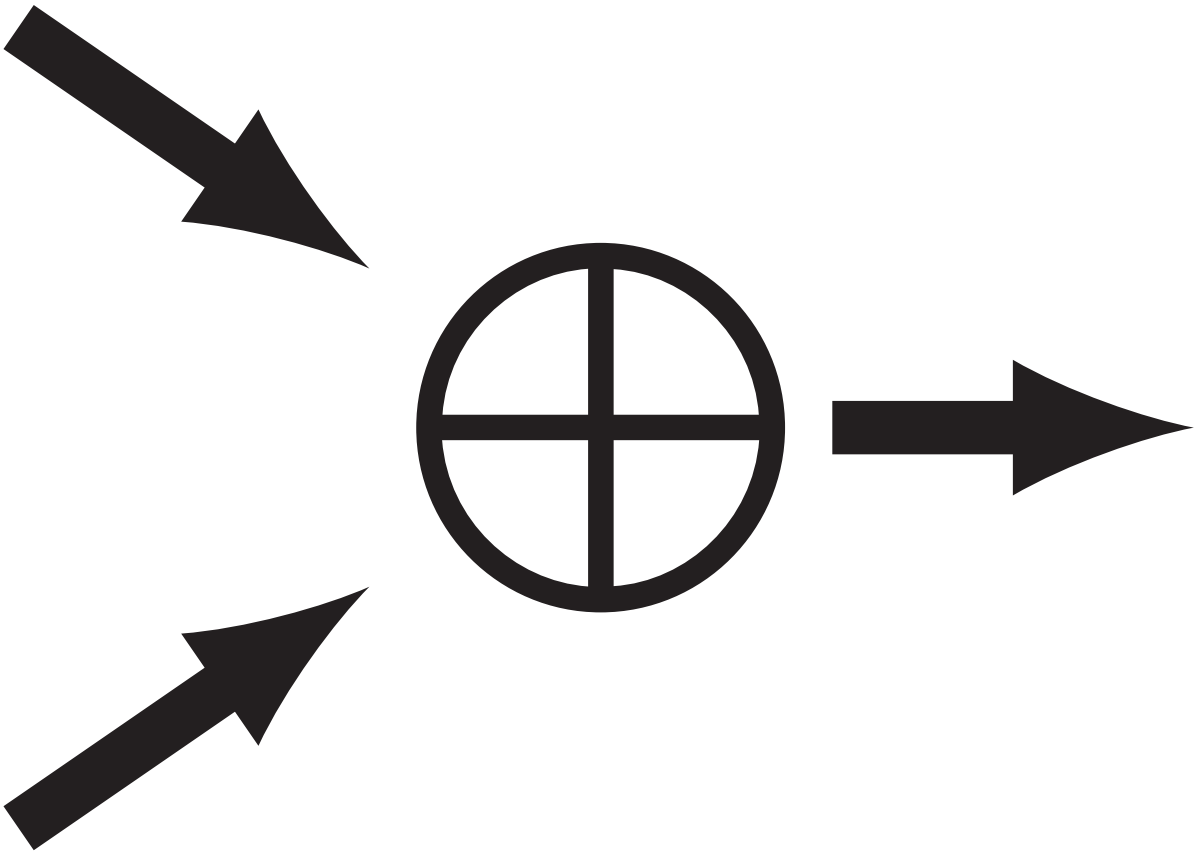


Stack Variable

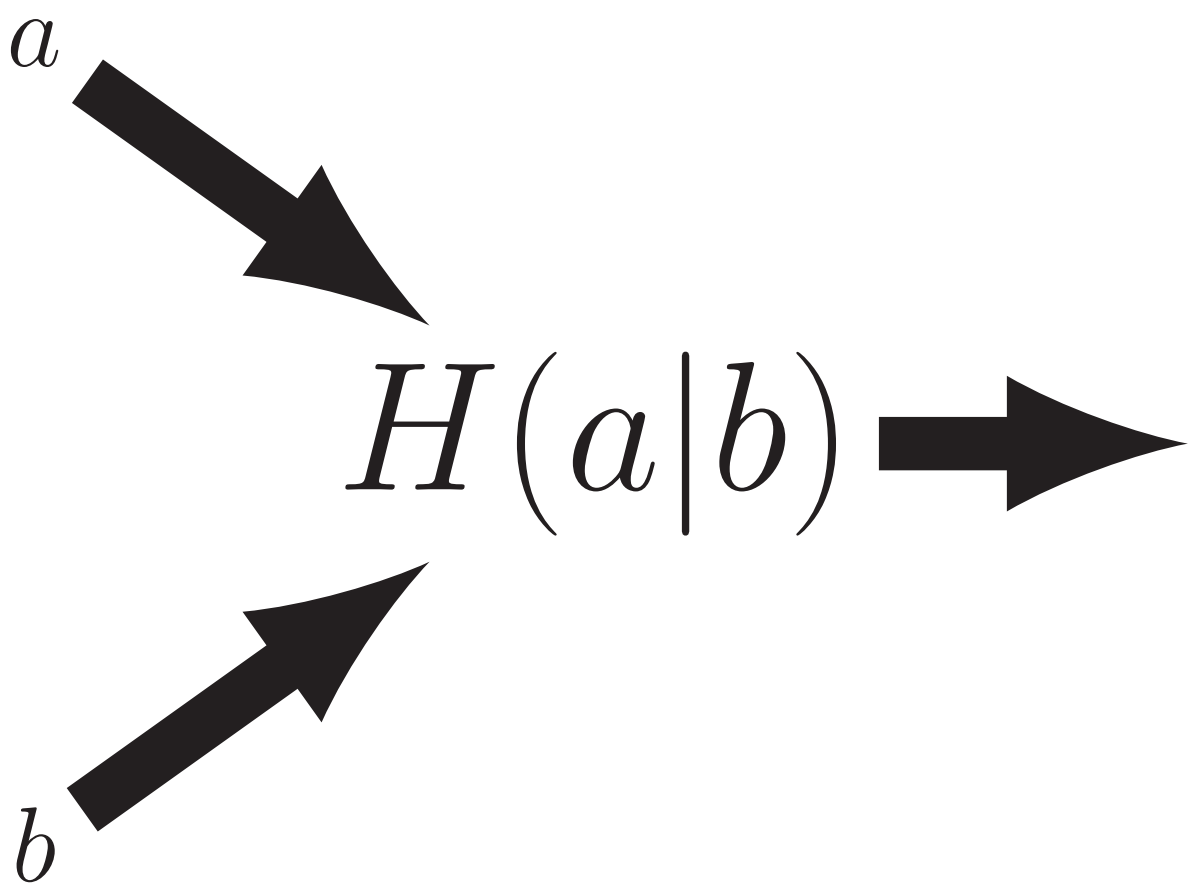
a



XOR



Hash



Role

Server

Role

Client

Role

Attacker

Role

Victim

Interface

Input Field

Name:

Interface

Image

Input

Terminator

’.
’;

</style>

Input

URL

`http://example.com/file.exe`

Attack

Script

<script>

do_bad_thing()

</script>

Attack Effect

Insert Data

INSERT INTO Table ...

store(...)

Attack Effect

Extract Data

SELECT * FROM Table ...

return(...)

get_cookie(...)

Attack Effect

Modify Data

UPDATE Table ...

set(...)

Server

Process Form

Server

Respond

Server

Echo

Server

Bind Input to Variable

Server

Variable

$input