Hashcash Proof-of-Work Experiment

Name: Samyak Gedam
Roll No: 252IS032

Objective

To experimentally analyze the effect of increasing leading zero-bit difficulty in the Hashcash Proof-of-Work system and measure the time required for generation and validation.

Theory

Hashcash is a Proof-of-Work (PoW) mechanism where a sender must find a value such that the SHA-1 hash of a message begins with a specified number of leading zero bits.

If the difficulty is b bits, the probability of success is:

$$P = \frac{1}{2^b}$$

Thus, the expected number of trials is 2^b, implying exponential growth in computation time as b increases. However, verification requires only one hash computation and therefore runs in constant time.

Commands Used

Minting:

```
time ./hashcash -m -b <bits> test@example.com
```

Validation:

```
time ./hashcash -c "<generated-stamp>"
```

SHA-1 Checksum:

```
sha1sum stamp.txt
```

Experimental Results

Zero Bits (b)	Real Time
20	0.074 s
22	0.955 s
24	3.158 s
26	7.512 s
28	12.994 s
30	2m 20.398 s
31	5m 28.175 s
32	~75 min (interrupted)

Validation Time: ~0.003 seconds

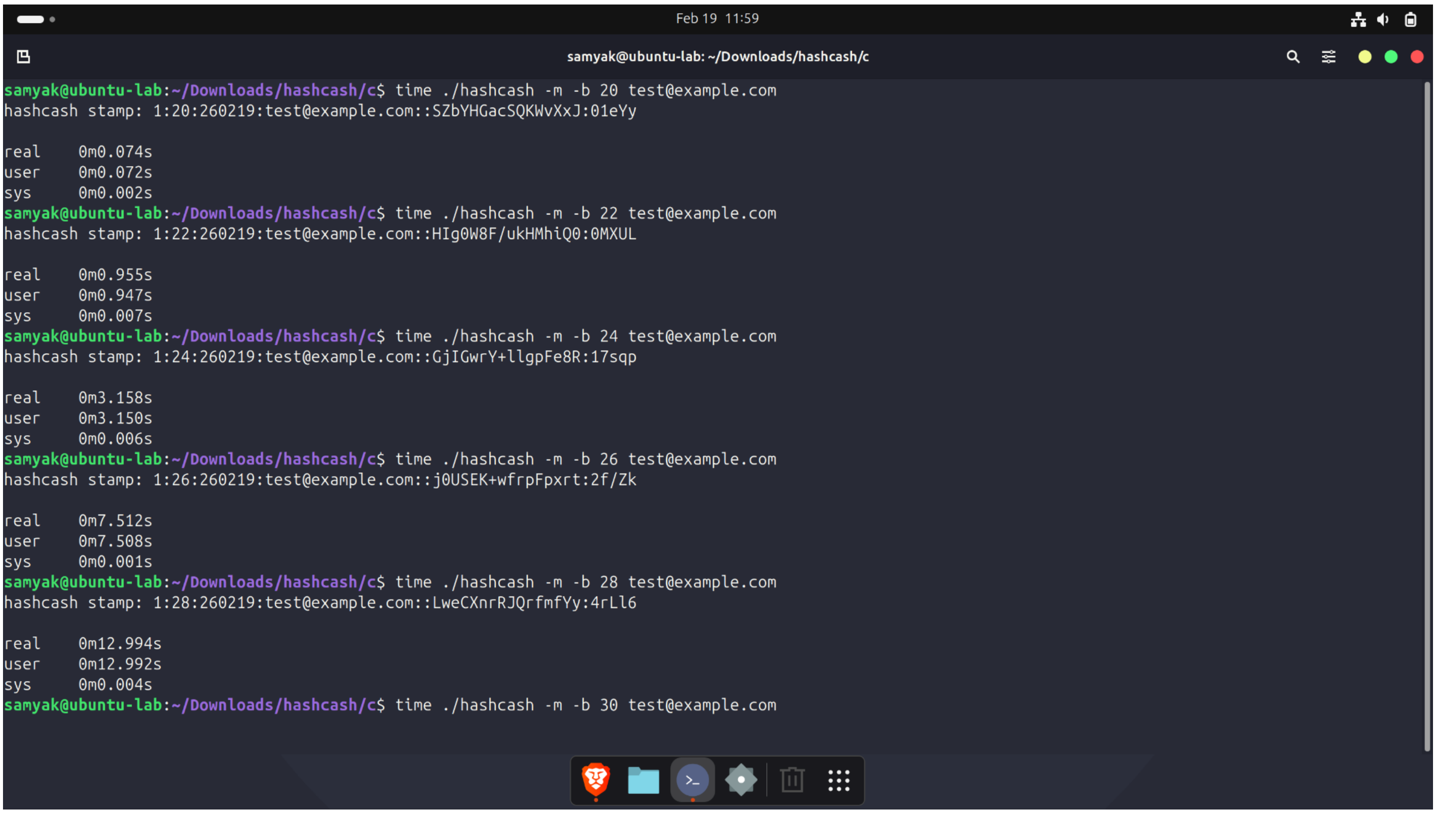
SHA-1 Checksum:

```
8fecb29dada6aedd061e332a23e24a193ac6dac7
```

Machine Specification

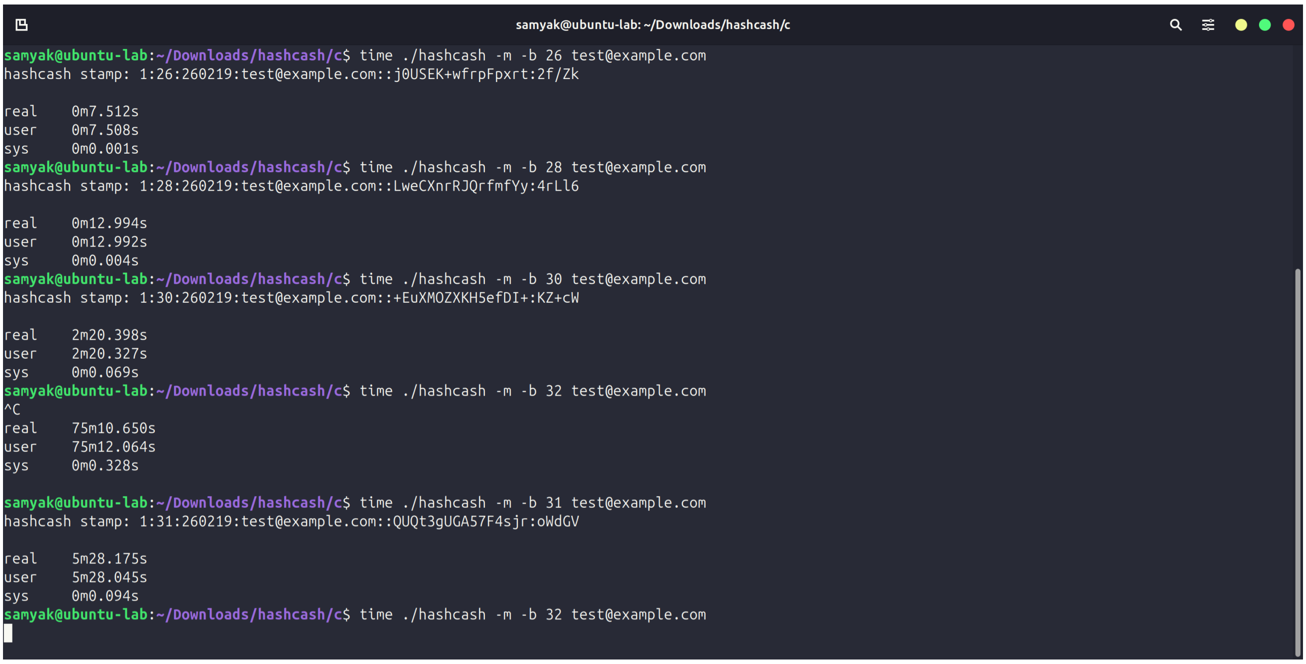
Architecture: x86_64 (64-bit)
Processor: AMD Ryzen 5 5500U with Radeon Graphics
CPU Cores: 6
Virtualization: KVM (Full Virtualization)
RAM: 3.8 GiB
Operating System: Ubuntu 24.04.1
Kernel: Linux 6.17.0-14-generic

Screenshots



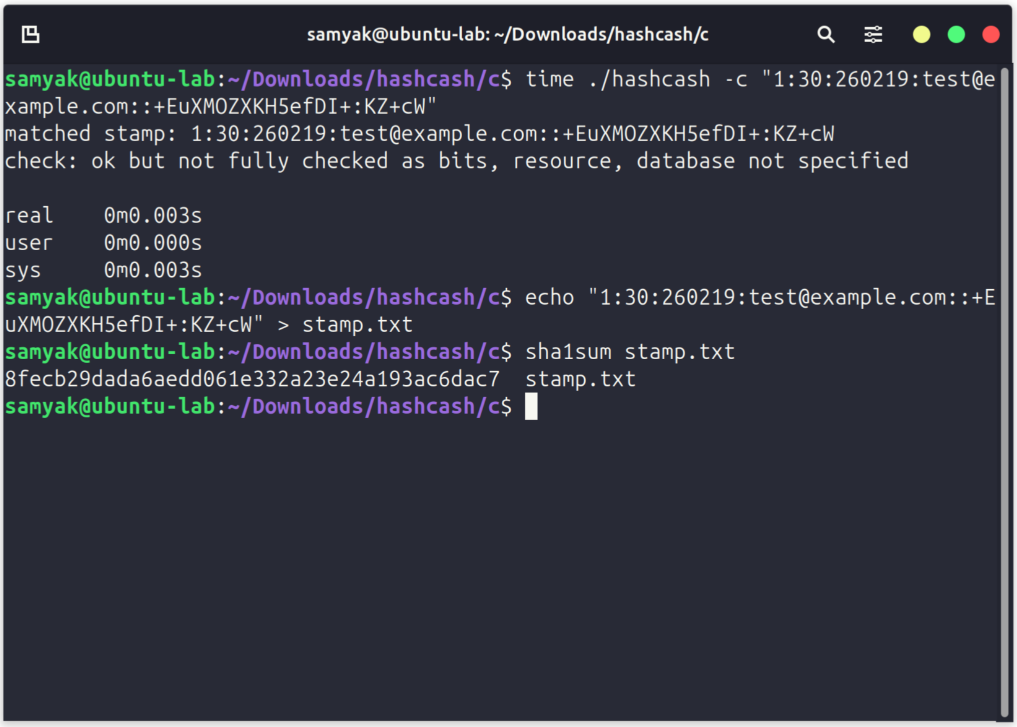
```
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 20 test@example.com
hashcash stamp: 1:20:260219:test@example.com::SZbYHGacSQKWvXxJ:01eYy

real    0m0.074s
user    0m0.072s
sys     0m0.002s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 22 test@example.com
hashcash stamp: 1:22:260219:test@example.com::HIg0W8F/ukHMhiQ0:0MXUL

real    0m0.955s
user    0m0.947s
sys     0m0.007s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 24 test@example.com
hashcash stamp: 1:24:260219:test@example.com::GjIGwrY+llgpFe8R:17sqp

real    0m3.158s
user    0m3.150s
sys     0m0.006s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 26 test@example.com
hashcash stamp: 1:26:260219:test@example.com::j0USEK+wfrpFpxrt:2f/Zk

real    0m7.512s
user    0m7.508s
sys     0m0.001s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 28 test@example.com
hashcash stamp: 1:28:260219:test@example.com::LweCXnrRJQrfmfYy:4rLl6

real    0m12.994s
user    0m12.992s
sys     0m0.004s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 30 test@example.com
```

Figure 1: Minting with increasing zero-bit difficulty



```
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 26 test@example.com
hashcash stamp: 1:26:260219:test@example.com::j0USEK+wfrpFpxrt:2f/Zk

real	0m7.512s
user	0m7.508s
sys	0m0.001s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 28 test@example.com
hashcash stamp: 1:28:260219:test@example.com::LweCXnrRJQrfmfYy:4rLl6

real	0m12.994s
user	0m12.992s
sys	0m0.004s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 30 test@example.com
hashcash stamp: 1:30:260219:test@example.com::+EuXMOZXKH5efDI+:KZ+cW

real	2m20.398s
user	2m20.327s
sys	0m0.069s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 32 test@example.com
^C
real	75m10.650s
user	75m12.064s
sys	0m0.328s

samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 31 test@example.com
hashcash stamp: 1:31:260219:test@example.com::QUQt3gUGA57F4sjr:oWdGV

real	5m28.175s
user	5m28.045s
sys	0m0.094s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -m -b 32 test@example.com
```

Figure 2: Extended execution showing exponential increase in computation time



```
samyak@ubuntu-lab:~/Downloads/hashcash/c$ time ./hashcash -c "1:30:260219:test@e
xample.com::+EuXMOZXKH5efDI+:KZ+cW"
matched stamp: 1:30:260219:test@example.com::+EuXMOZXKH5efDI+:KZ+cW
check: ok but not fully checked as bits, resource, database not specified

real	0m0.003s
user	0m0.000s
sys	0m0.003s
samyak@ubuntu-lab:~/Downloads/hashcash/c$ echo "1:30:260219:test@example.com::+E
uXMOZXKH5efDI+:KZ+cW" > stamp.txt
samyak@ubuntu-lab:~/Downloads/hashcash/c$ sha1sum stamp.txt
8fecb29dada6aedd061e332a23e24a193ac6dac7  stamp.txt
samyak@ubuntu-lab:~/Downloads/hashcash/c$
```

Figure 3: Validation and SHA-1 checksum

Conclusion

The experiment demonstrates that Hashcash minting time increases exponentially with the number of leading zero bits, consistent with the expected complexity of $O(2^b)$.

In contrast, validation requires only a single SHA-1 computation and therefore operates in constant time $O(1)$.

This asymmetry between computation and verification forms the core principle of Proof-of-Work systems used in blockchain networks.