The fuzzy tale of an `x/crypto` vulnerability

Michael McLoughlin

Gophercon 2019 Lightning Talks

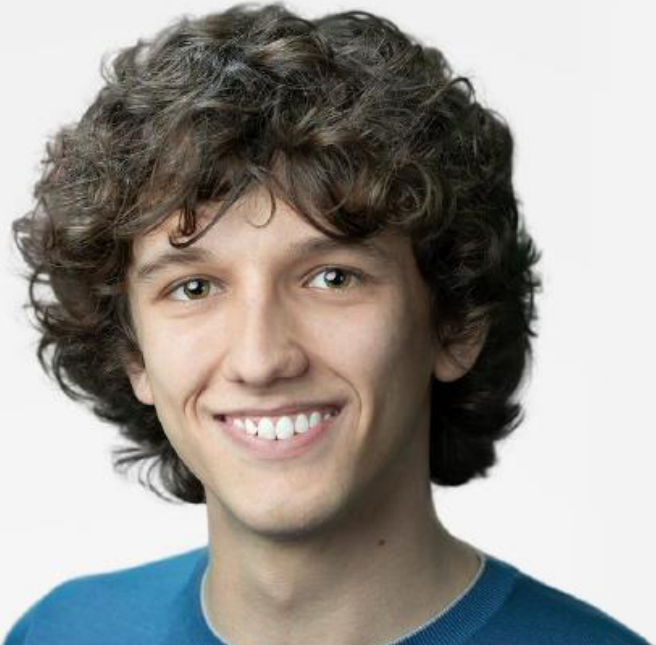
Uber Advanced Technologies Group

8,140

lines of `amd64` assembly in `crypto`

10,474

lines of `amd64` assembly in `golang.org/x/crypto`



Fuzzing

Fuzzing is an automated testing technique for hardening safety-critical software

Typically used where code must handle untrusted inputs or correctness is paramount: parsers, network protocols, cryptography, ...

github.com/dvyukov/go-fuzz

```go
func Fuzz(data []byte) int
```

```go
func Fuzz(data []byte) int {
    parse(data)
    return 0
}
```

Hit your target function with
cleverly-constructed random data.

Differential fuzzing: compare against a reference implementation.

github.com/mmcloughlin/cryptofuzz

```go
func Fuzz(data []byte) int {
    if purego(data) != asm(data) {
        panic("mismatch")
    }
    return 0
}
```

- `crypto/aes` (GCM mode)
- `crypto/elliptic` (P256)
- `crypto/sha1`
- `crypto/sha256`
- `crypto/sha512`

- x/crypto/chacha20poly1305
- x/crypto/sha3
- x/crypto/blake2b
- x/crypto/blake2s
- x/crypto/argon2
- x/crypto/poly1305

⚠️ x/crypto/curve25519

✖ x/crypto/salsa20

```
2019/07/16 23:34:59 workers: 4, corpus: 5 (1s ago), crashers: 0, restarts: 1/0, execs: 0 (0/sec), cover: 0, uptime: 3s
2019/07/16 23:35:02 workers: 4, corpus: 6 (1s ago), crashers: 0, restarts: 1/6343, execs: 19031 (3171/sec), cover: 26,
2019/07/16 23:35:05 workers: 4, corpus: 6 (4s ago), crashers: 0, restarts: 1/6797, execs: 95167 (10568/sec), cover: 26,
2019/07/16 23:35:08 workers: 4, corpus: 6 (7s ago), crashers: 0, restarts: 1/7269, execs: 145385 (12113/sec), cover: 26
2019/07/16 23:35:11 workers: 4, corpus: 7 (2s ago), crashers: 0, restarts: 1/7269, execs: 203535 (13564/sec), cover: 26
2019/07/16 23:35:14 workers: 4, corpus: 7 (5s ago), crashers: 0, restarts: 1/6375, execs: 312406 (17354/sec), cover: 26
2019/07/16 23:35:17 workers: 4, corpus: 7 (8s ago), crashers: 0, restarts: 1/6063, execs: 394141 (18763/sec), cover: 26
2019/07/16 23:35:20 workers: 4, corpus: 7 (11s ago), crashers: 0, restarts: 1/2739, execs: 457416 (19055/sec), cover: 2
2019/07/16 23:35:23 workers: 4, corpus: 7 (14s ago), crashers: 0, restarts: 1/1349, execs: 457588 (16944/sec), cover: 2
2019/07/16 23:35:26 workers: 4, corpus: 7 (17s ago), crashers: 0, restarts: 1/883, execs: 457767 (15256/sec), cover: 26
2019/07/16 23:35:29 workers: 4, corpus: 7 (20s ago), crashers: 0, restarts: 1/654, execs: 457949 (13876/sec), cover: 26
2019/07/16 23:35:32 workers: 4, corpus: 7 (23s ago), crashers: 1, restarts: 1/529, execs: 458114 (12725/sec), cover: 26
2019/07/16 23:35:35 workers: 4, corpus: 7 (26s ago), crashers: 1, restarts: 1/440, execs: 458290 (11750/sec), cover: 26
2019/07/16 23:35:38 workers: 4, corpus: 7 (29s ago), crashers: 1, restarts: 1/390, execs: 469197 (11171/sec), cover: 26
2019/07/16 23:35:41 workers: 4, corpus: 7 (32s ago), crashers: 1, restarts: 1/397, execs: 512961 (11398/sec), cover: 26
2019/07/16 23:35:44 workers: 4, corpus: 7 (35s ago), crashers: 1, restarts: 1/437, execs: 572689 (11931/sec), cover: 26
2019/07/16 23:35:47 workers: 4, corpus: 7 (38s ago), crashers: 1, restarts: 1/490, execs: 647623 (12698/sec), cover: 26
2019/07/16 23:35:50 workers: 4, corpus: 7 (41s ago), crashers: 1, restarts: 1/544, execs: 726490 (13452/sec), cover: 26
2019/07/16 23:35:53 workers: 4, corpus: 7 (44s ago), crashers: 1, restarts: 1/594, execs: 803207 (14091/sec), cover: 26
2019/07/16 23:35:56 workers: 4, corpus: 7 (47s ago), crashers: 1, restarts: 1/644, execs: 880605 (14676/sec), cover: 26
2019/07/16 23:35:59 workers: 4, corpus: 7 (50s ago), crashers: 1, restarts: 1/698, execs: 963476 (15292/sec), cover: 26
2019/07/16 23:36:02 workers: 4, corpus: 7 (53s ago), crashers: 1, restarts: 1/748, execs: 1042443 (15793/sec), cover: 2
2019/07/16 23:36:05 workers: 4, corpus: 7 (56s ago), crashers: 1, restarts: 1/787, execs: 1108594 (16066/sec), cover: 2
2019/07/16 23:36:08 workers: 4, corpus: 7 (59s ago), crashers: 1, restarts: 1/831, execs: 1181187 (16404/sec), cover: 2
...
```

```
2019/07/16 23:34:59 workers: 4, corpus: 5 (1s ago), crashers: 0, restarts: 1/0, execs: 0 (0/sec), cover: 0, uptime: 3s
2019/07/16 23:35:02 workers: 4, corpus: 6 (1s ago), crashers: 0, restarts: 1/6343, execs: 19031 (3171/sec), cover: 26,
2019/07/16 23:35:05 workers: 4, corpus: 6 (4s ago), crashers: 0, restarts: 1/6797, execs: 95167 (10568/sec), cover: 26,
2019/07/16 23:35:08 workers: 4, corpus: 6 (7s ago), crashers: 0, restarts: 1/7269, execs: 145385 (12113/sec), cover: 26
2019/07/16 23:35:11 workers: 4, corpus: 7 (2s ago), crashers: 0, restarts: 1/7269, execs: 203535 (13564/sec), cover: 26
2019/07/16 23:35:14 workers: 4, corpus: 7 (5s ago), crashers: 0, restarts: 1/6375, execs: 312406 (17354/sec), cover: 26
2019/07/16 23:35:17 workers: 4, corpus: 7 (8s ago), crashers: 0, restarts: 1/6063, execs: 394141 (18763/sec), cover: 26
2019/07/16 23:35:20 workers: 4, corpus: 7 (11s ago), crashers: 0, restarts: 1/2739, execs: 457416 (19055/sec), cover: 2
2019/07/16 23:35:23 workers: 4, corpus: 7 (14s ago), crashers: 0, restarts: 1/1349, execs: 457588 (16944/sec), cover: 2
2019/07/16 23:35:26 workers: 4, corpus: 7 (17s ago), crashers: 0, restarts: 1/883, execs: 457767 (15256/sec), cover: 26
2019/07/16 23:35:29 workers: 4, corpus: 7 (20s ago), crashers: 0, restarts: 1/654, execs: 457949 (13876/sec), cover: 26
2019/07/16 23:35:32 workers: 4, corpus: 7 (23s ago), crashers: 1, restarts: 1/529, execs: 458114 (12725/sec), cover: 26
2019/07/16 23:35:35 workers: 4, corpus: 7 (26s ago), crashers: 1, restarts: 1/440, execs: 458290 (11750/sec), cover: 26
2019/07/16 23:35:38 workers: 4, corpus: 7 (29s ago), crashers: 1, restarts: 1/390, execs: 469197 (11171/sec), cover: 26
2019/07/16 23:35:41 workers: 4, corpus: 7 (32s ago), crashers: 1, restarts: 1/397, execs: 512961 (11398/sec), cover: 26
2019/07/16 23:35:44 workers: 4, corpus: 7 (35s ago), crashers: 1, restarts: 1/437, execs: 572689 (11931/sec), cover: 26
2019/07/16 23:35:47 workers: 4, corpus: 7 (38s ago), crashers: 1, restarts: 1/490, execs: 647623 (12698/sec), cover: 26
2019/07/16 23:35:50 workers: 4, corpus: 7 (41s ago), crashers: 1, restarts: 1/544, execs: 726490 (13452/sec), cover: 26
2019/07/16 23:35:53 workers: 4, corpus: 7 (44s ago), crashers: 1, restarts: 1/594, execs: 803207 (14091/sec), cover: 26
2019/07/16 23:35:56 workers: 4, corpus: 7 (47s ago), crashers: 1, restarts: 1/644, execs: 880605 (14676/sec), cover: 26
2019/07/16 23:35:59 workers: 4, corpus: 7 (50s ago), crashers: 1, restarts: 1/698, execs: 963476 (15292/sec), cover: 26
2019/07/16 23:36:02 workers: 4, corpus: 7 (53s ago), crashers: 1, restarts: 1/748, execs: 1042443 (15793/sec), cover: 2
2019/07/16 23:36:05 workers: 4, corpus: 7 (56s ago), crashers: 1, restarts: 1/787, execs: 1108594 (16066/sec), cover: 2
2019/07/16 23:36:08 workers: 4, corpus: 7 (59s ago), crashers: 1, restarts: 1/831, execs: 1181187 (16404/sec), cover: 2
...
```

```
$ cat crashers/ed31ec2f4f2f123330e58557cac892bebda17549.output
counter=303030303030303030303030303030
key=303030303030303030303030303030303030303030303030303030303030303030
data=3030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030
out=cdc5b5296d5857a6328bb222e00f2a1818a2320541b9996c5de9e336f3db7ef338759022120a91263b098d4ea4d7b397fce8a9b24fa39a2931f
ref=cdc5b5296d5857a6328bb222e00f2a1818a2320541b9996c5de9e336f3db7ef338759022120a91263b098d4ea4d7b397fce8a9b24fa39a2931f
panic: mismatch

goroutine 1 [running]:
github.com/mmcloughlin/cryptofuzz/target/salsa20.Fuzz(0x1571000, 0x130, 0x200000, 0xc000080f58)
^^I/var/folders/p5/84p384bs42v7pbgfx0db9gq80000gn/T/go-fuzz-build019783832/gopath/src/github.com/mmcloughlin/cryptofuzz
go-fuzz-dep.Main(0x10e84e8)
^^I/var/folders/p5/84p384bs42v7pbgfx0db9gq80000gn/T/go-fuzz-build019783832/goroot/src/go-fuzz-dep/main.go:54 +0xb6
main.main()
^^I/var/folders/p5/84p384bs42v7pbgfx0db9gq80000gn/T/go-fuzz-build019783832/gopath/src/github.com/mmcloughlin/cryptofuzz
```

```
$ cat crashers/ed31ec2f4f2f123330e58557cac892bebda17549.output
counter=3030303030303030303030303030303030
key=30303030303030303030303030303030303030303030303030303030303030303030
data=303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030303030
out=cdc5b5296d5857a6328bb222e00f2a1818a2320541b9996c5de9e336f3db7ef338759022120a91263b098d4ea4d7b397fce8a9b24fa39a2931f
ref=cdc5b5296d5857a6328bb222e00f2a1818a2320541b9996c5de9e336f3db7ef338759022120a91263b098d4ea4d7b397fce8a9b24fa39a2931f
panic: mismatch

goroutine 1 [running]:
github.com/mmcloughlin/cryptofuzz/target/salsa20.Fuzz(0x1571000, 0x130, 0x200000, 0xc000080f58)
^^I/var/folders/p5/84p384bs42v7pbgfx0db9gq80000gn/T/go-fuzz-build019783832/gopath/src/github.com/mmcloughlin/cryptofuzz
go-fuzz-dep.Main(0x10e84e8)
^^I/var/folders/p5/84p384bs42v7pbgfx0db9gq80000gn/T/go-fuzz-build019783832/goroot/src/go-fuzz-dep/main.go:54 +0xb6
main.main()
^^I/var/folders/p5/84p384bs42v7pbgfx0db9gq80000gn/T/go-fuzz-build019783832/gopath/src/github.com/mmcloughlin/cryptofuzz
```

Salsa20 Stream Cipher

Plaintext	57	65	20	69	6e	74	65	6e	64	20	74	6f	20	62	65	67	···

$\oplus$

Keystream	2b	35	8d	90	67	9c	cc	95	cc	83	ce	86	ef	af	da	ec	···

$=$

Ciphertext	7c	50	ad	f9	09	e8	a9	fb	a8	a3	ba	e9	cf	cd	bf	8b	···

Plaintext	57	65	20	69	6e	74	65	6e	64	20	74	6f	20	62	65	67	···

$\oplus$

Keystream	2b	35	8d	90	67	9c	cc	95	cc	83	ce	86	ef	af	da	ec	···

$=$

Ciphertext	7c	50	ad	f9	09	e8	a9	fb	a8	a3	ba	e9	cf	cd	bf	8b	···

Plaintext	57	65	20	69	6e	74	65	6e	64	20	74	6f	20	62	65	67	···

$\oplus$

Keystream	2b	35	8d	90	67	9c	cc	95	cc	83	ce	86	ef	af	da	ec	···

$=$

Ciphertext	7c	50	ad	f9	09	e8	a9	fb	a8	a3	ba	e9	cf	cd	bf	8b	···

Plaintext	57	65	20	69	6e	74	65	6e	64	20	74	6f	20	62	65	67	···

$\oplus$

Keystream	2b	35	8d	90	67	9c	cc	95	cc	83	ce	86	ef	af	da	ec	···

$=$

Ciphertext	7c	50	ad	f9	09	e8	a9	fb	a8	a3	ba	e9	cf	cd	bf	8b	···

block(0x0000000000000000, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

block(0x0000000000000001, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

block(0x0000000000000002, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

block(0x0000000000000003, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

block(0x0000000000000004, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

block(0x0000000000000005, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

block(0x0000000000000006, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

block(0x0000000000000007, key)

```
2b 35 8d 90 67 9c cc 95 cc 83 ce 86 ef af da ec d7 91 47 ae 2d ef e3 ea ef ac 00 8b e8 c1 2d 91
29 ef bb 93 f7 41 14 47 b7 23 6b 72 25 a9 ab c7 11 51 25 f2 91 39 12 f8 6e 05 d1 75 d5 24 14 fc
10 63 7d e6 1f 02 6b 22 d1 66 7c e1 75 41 e9 58 21 8f a6 21 8e 0a 5b 16 46 d9 5b 69 5b 19 57 ca
d9 28 b8 b5 1d da 3f 97 e8 8d 9a cb 34 b6 f3 e0 74 2a 2f 35 e8 00 1a c9 d5 d7 35 c2 a8 eb 23 ae
8f 94 05 78 59 ea 25 8e 76 2d 75 62 02 88 fc 31 cc 8e 3e cd 18 61 95 16 c7 bc 4b 9b 0b 86 08 4e
5c 42 1b d0 93 aa a0 3f 7c 68 0c b6 c3 59 1e 4f 87 68 3b 41 d4 2f 1d 9d e6 8a e7 19 54 62 fa ea
c0 ab f8 a9 a2 2a b7 33 ef d2 10 46 ba 71 c5 86 c0 3c 6e b9 c7 fa 50 57 3d 0f 9b 8b 0b 3d 21 a7
bd 62 fc 5f b7 4e 21 d5 6f b5 27 57 68 ff 6e a4 b1 a0 51 06 f5 b2 11 cd 46 d8 be 3e ad a1 be 3d
9f e1 89 46 6c 99 e6 83 f3 82 d5 bb b4 bd d5 0c c5 4e b4 66 49 1c 99 b4 cc d0 92 d1 c8 16 75 ac
e8 70 ac ba ee 3b 0a 05 00 b3 bd 77 28 08 24 c9 96 fe f5 a0 03 ab 8c ba 1a 66 15 e4 99 21 59 e6
4d 19 89 18 0c ef 63 6a fa 05 4d bf 36 ea ce 32 53 4b f4 c6 38 3c e1 0c 85 c1 c7 0c e3 dd a8 da
de 04 c8 a2 19 bc 8d 53 43 ac e3 b2 10 4b 11 ec 54 c2 a5 cb 49 3f c9 2c f6 e2 5a e4 27 11 41 62
4c da 33 7c fe a8 11 f0 0c 20 c9 63 9c 34 98 54 39 81 41 cc 2f 8e 94 4d 27 49 77 3f 22 55 7d 45
48 26 17 04 29 1a 6f 71 7d 42 0d 2a 75 35 b9 cd fe 05 5e 10 96 48 b6 4b bd 4e 91 29 c7 96 ef 9a
33 64 4f 52 9b 5d 09 46 03 09 a4 a2 09 f8 32 7f 7f 4c 0d a4 e0 f7 7b c3 08 79 96 fb 00 81 13 67
2b 7e 74 6a 66 15 60 03 19 28 f0 36 5a a2 42 13 3f 6c c9 33 40 ac 72 f0 82 85 4e 78 73 06 65 f1
```

Crasher Observations

Param	Length	Value
counter	8	0x3030303030303030
key	32	0x3030 … 303030
plain	512	0x3030 … 303030

Crasher Observations

Param	Length	Value
counter	8	0x30303030303030 30
key	32	0x3030 … 303030
plain	512	0x3030 … 303030

- High 32-bits of counter non-zero
- Input at least 256 bytes

Diving into Assembly: BYTESATLEAST256

```
BYTESATLEAST256:
MOVL 16(SP),DX
MOVL  36 (SP),CX
MOVL DX,288(SP)
MOVL CX,304(SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX, 292 (SP)
MOVL CX, 308 (SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
```

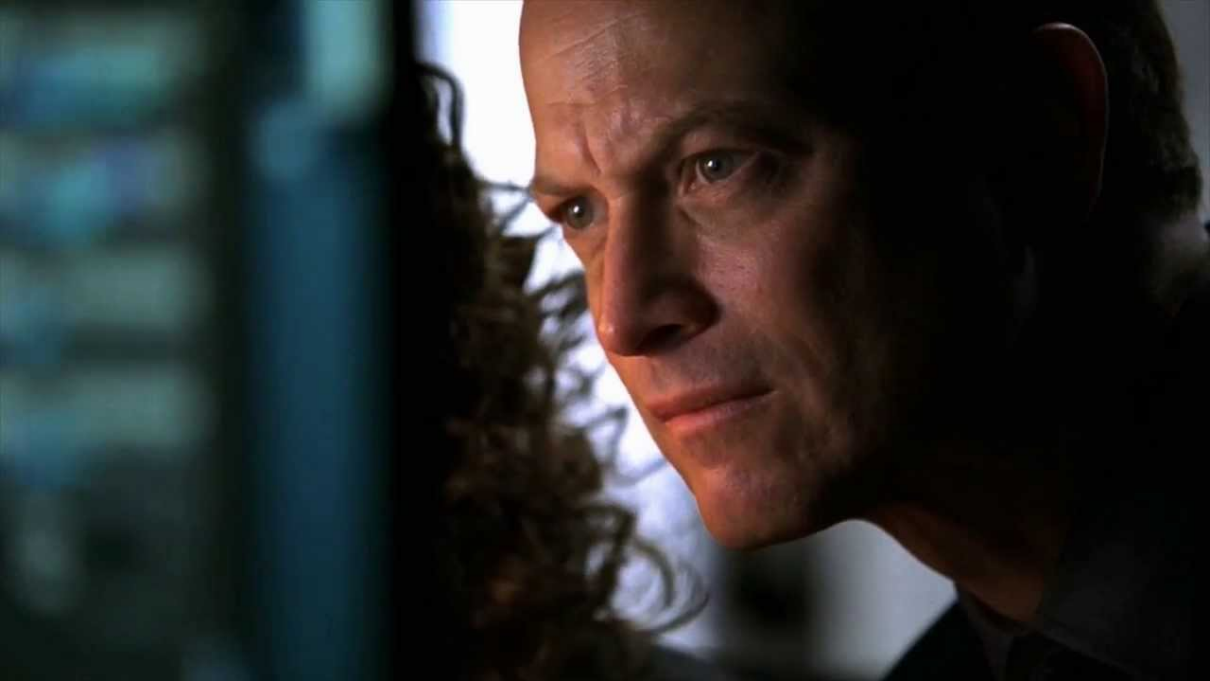
```
MOVL DX, 296 (SP)
MOVL CX, 312 (SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX, 300 (SP)
MOVL CX, 316 (SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,16(SP)
MOVL CX, 36 (SP)
```

Diving into Assembly: BYTESATLEAST256

```
BYTESATLEAST256:
MOVL 16(SP),DX
MOVL  36 (SP),CX
MOVL DX,288(SP)
MOVL CX,304(SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX, 292 (SP)
MOVL CX, 308 (SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
```

```
MOVL DX, 296 (SP)
MOVL CX, 312 (SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX, 300 (SP)
MOVL CX, 316 (SP)
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,16(SP)
MOVL CX, 36 (SP)
```

4 × Counter Update



```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
 DX  00000000ffffffff  ‹ low 32
 CX  0000000000000000  ‹ high 32
ctr  00000000ffffffff  ‹ full counter
```

```
ADDQ $1,DX     ‹ Increment low 32 bits
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
 DX  0000000100000000  ‹ low 32
 CX  0000000000000000  ‹ high 32
ctr  00000000ffffffff  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX    ‹ Shift high into place
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
 DX  0000000100000000  ‹ low 32
 CX  0000000000000000  ‹ high 32
ctr  00000000ffffffff  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX          ‹ Add high into low
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

DX	0000000100000000	‹ low 32
CX	0000000000000000	‹ high 32
ctr	00000000ffffffff	‹ full counter

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX   ‹ Copy full 64-bit result
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
  DX  0000000100000000  ‹ low 32
  CX  0000000100000000  ‹ high 32
 ctr  00000000ffffffff  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX     ‹ Extract high 32 bits
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
  DX  0000000100000000  ‹ low 32
  CX  0000000000000001  ‹ high 32
 ctr  00000000ffffffff  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)    ‹ Store low 32 bits
MOVL CX,308(SP)
```

```
 DX  0000000100000000  ‹ low 32
 CX  0000000000000001  ‹ high 32
ctr  0000000000000000  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)  ‹ Store high 32 bits
```

```
 DX  0000000100000000  ‹ low 32
 CX  0000000000000001  ‹ high 32
ctr  0000000100000000  ‹ full counter
```

```
ADDQ $1,DX     ‹ Increment low 32 bits
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
  DX  0000000100000001  ‹ low 32
  CX  0000000000000001  ‹ high 32
 ctr  0000000100000000  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX    ‹ Shift high into place
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
 DX  0000000100000001  ‹ low 32
 CX  0000000100000000  ‹ high 32
ctr  0000000100000000  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX          ‹ Add high into low
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

DX	0000000200000001	‹ low 32
CX	0000000100000000	‹ high 32
ctr	0000000100000000	‹ full counter

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX   ‹ Copy full 64-bit result
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
  DX  0000000200000001  ‹ low 32
  CX  0000000200000001  ‹ high 32
 ctr  0000000100000000  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX     ‹ Extract high 32 bits
MOVL DX,292(SP)
MOVL CX,308(SP)
```

```
 DX  0000000200000001  ‹ low 32
 CX  0000000000000002  ‹ high 32
ctr  0000000100000000  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)    ‹ Store low 32 bits
MOVL CX,308(SP)
```

```
 DX  0000000200000001  ‹ low 32
 CX  0000000000000002  ‹ high 32
ctr  0000000100000001  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)  ‹ Store high 32 bits
```

```
 DX  0000000200000001  ‹ low 32
 CX  0000000000000002  ‹ high 32
ctr  0000000200000001  ‹ full counter
```

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

DX	0000000400000002	‹ low 32
CX	0000000000000004	‹ high 32
ctr	0000000400000002	‹ full counter

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

DX	0000000800000003	‹ low 32
CX	0000000000000008	‹ high 32
ctr	0000000800000003	‹ full counter

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

DX	0000001000000004	‹ low 32
CX	0000000000000010	‹ high 32
ctr	0000001000000004	‹ full counter

```
ADDQ $1,DX
SHLQ $32,CX
ADDQ CX,DX
MOVQ DX,CX
SHRQ $32,CX
MOVL DX,292(SP)
MOVL CX,308(SP)
```

DX	0000002000000005	‹ low 32
CX	0000000000000020	‹ high 32
ctr	0000002000000005	‹ full counter

Counter update doubles the high 32 bits.

Once the counter hits 2^{32} the 1-bit in the high half will be shifted out shortly after.

Counter cycles back to beginning.

Verified by encrypting 256+ GiB!

Discovery in Upstreams

- Go implementation ported from `SUPERCOP`
- Confirmed to still have the bug
- More seriously: also present in `NaCl`

Disclosure

security@golang.org

8 salsa20/salsa/salsa2020_amd64.s → salsa20/salsa/salsa20_amd64.s

```diff
@@ -99,30 +99,24 @@ TEXT ·salsa2020XORKeyStream(SB),0,$456-40 // frame = 424 + 32 byte alignment
 99          MOVL  36 (SP),CX
100          MOVL DX,288(SP)
101          MOVL CX,304(SP)
102 -        ADDQ $1,DX
103          SHLQ $32,CX
104          ADDQ CX,DX
105          MOVQ DX,CX
106          SHRQ $32,CX
107          MOVL DX, 292 (SP)
108          MOVL CX, 308 (SP)
109          ADDQ $1,DX
110 -        SHLQ $32,CX
111 -        ADDQ CX,DX
112          MOVQ DX,CX
113          SHRQ $32,CX
114          MOVL DX, 296 (SP)
115          MOVL CX, 312 (SP)
116          ADDQ $1,DX
117 -        SHLQ $32,CX
118 -        ADDQ CX,DX
119          MOVQ DX,CX
120          SHRQ $32,CX
121          MOVL DX, 300 (SP)
122          MOVL CX, 316 (SP)
123          ADDQ $1,DX
124 -        SHLQ $32,CX
125 -        ADDQ CX,DX
126          MOVQ DX,CX
127          SHRQ $32,CX
128          MOVL DX,16(SP)
```

```diff
99          MOVL  36 (SP),CX
100         MOVL DX,288(SP)
101         MOVL CX,304(SP)
102         SHLQ $32,CX
103         ADDQ CX,DX
104 +       ADDQ $1,DX
105         MOVQ DX,CX
106         SHRQ $32,CX
107         MOVL DX, 292 (SP)
108         MOVL CX, 308 (SP)
109         ADDQ $1,DX
110         MOVQ DX,CX
111         SHRQ $32,CX
112         MOVL DX, 296 (SP)
113         MOVL CX, 312 (SP)
114         ADDQ $1,DX
115         MOVQ DX,CX
116         SHRQ $32,CX
117         MOVL DX, 300 (SP)
118         MOVL CX, 316 (SP)
119         ADDQ $1,DX
120         MOVQ DX,CX
121         SHRQ $32,CX
122         MOVL DX,16(SP)
```

Today we published a security fix for golang.org/x/crypto/salsa…. If you generated more than 256 GiB of output from a single key+nonce pair, it would loop due to a counter overflow. Found by @mbmcloughlin's fuzzers. groups.google.com/d/msg/golang-a…

Filippo Valsorda @FiloSottile
8:21pm - 20 Mar 2019

"Let's take code from the SUPERCOP benchmarking framework. Does this file supercop/crypto_stream/salsa20/e/amd64-xmm6/warning-256gb mean anything? Probably not." [Time passes] "BREAKING NEWS: We found that this implementation doesn't work after 256GB!" groups.google.com/forum/#!msg/go…

Daniel J. Bernstein @hashbreaker
0:51am - 21 Mar 2019

Thanks

Filippo Valsorda and Adam Langley

https://fuzzbuzz.io

https://github.com/mmcloughlin/cryptofuzz

@mbmcloughlin