Avalanche effect in improperly initialized CAESAR candidates

Martin Ukrop, Petr Švenda
and the EACirc team

CROCS
Centre for Research on
Cryptography and Security

icons from The Noun Project

MEMICS, 21. 10. 2016

Cryptoprimitives abuse: OpenSSL example

```
void RAND_seed(const void *buf, int num);
```

Cryptoprimitives abuse: OpenSSL example

void RAND_seed(const void *buf, int num);

OpenSSL minor update (0.9.4 >> 0.9.5):

- Adding a small check to random generator if at least some entropy had been added (proper seeding)

Cryptoprimitives abuse: OpenSSL example

void RAND_seed(const void *buf, int num);

OpenSSL minor update (0.9.4 >> 0.9.5):

- Adding a small check to random generator if at least some entropy had been added (proper seeding)

... this lead to a flood of error reports.

Initi[illegible]pose: [illegible]SSL example

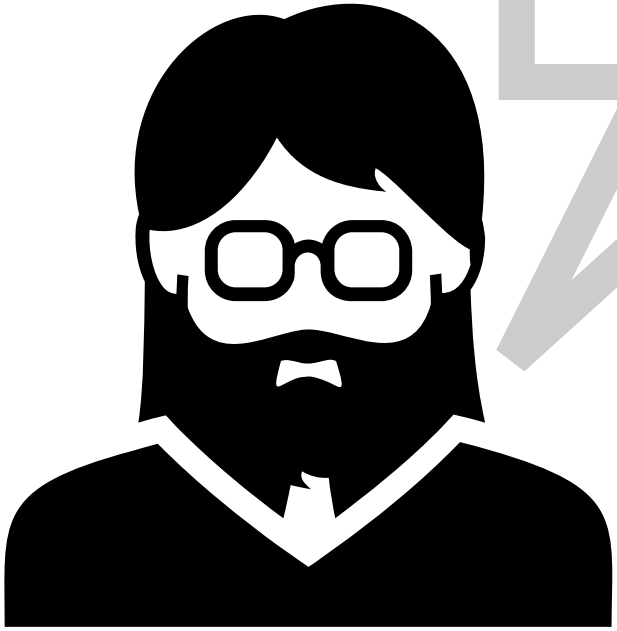
- [illegible]SL init[illegible]5)
- Add[illegible]
  if at le[illegible]
  (pro[illegible]
- [illegible]r re[illegible]

example courtesy of Peter Gutmann
Lessons learned in implementing
and deploying crypto software
USENIX 2002

- constant string
- public key components
- rand() calls
- /etc/passwd, /var/log/syslog
- downgrading
- seeding by the unseeded generator
- “patching” to disable entropy checks
- ...

A call for ...

Developer-resistant cryptography!*



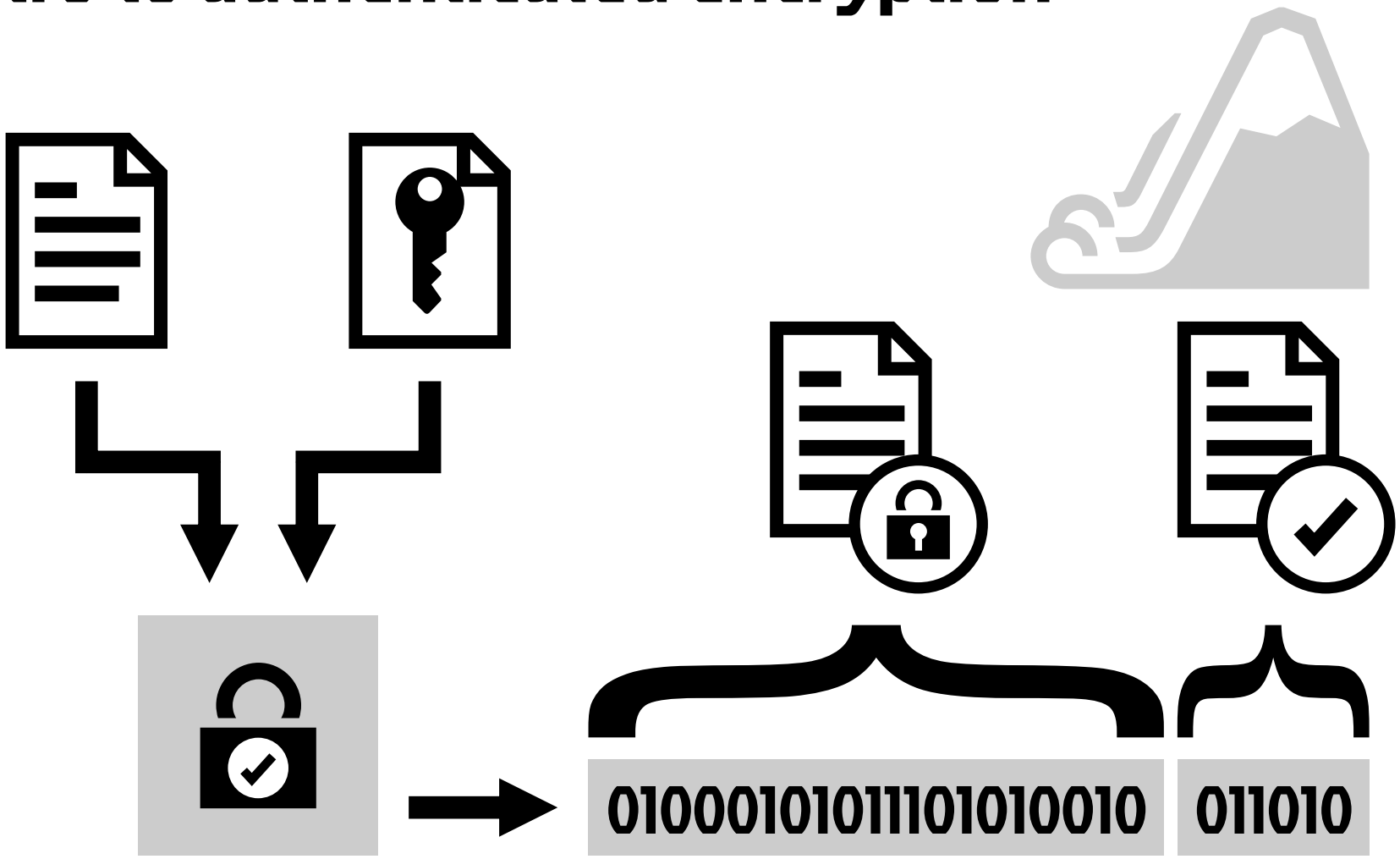
*Kelsey Cairns, Graham Steel

W3C/IAB Workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT), London, UK, February 28-March 1, 2014.

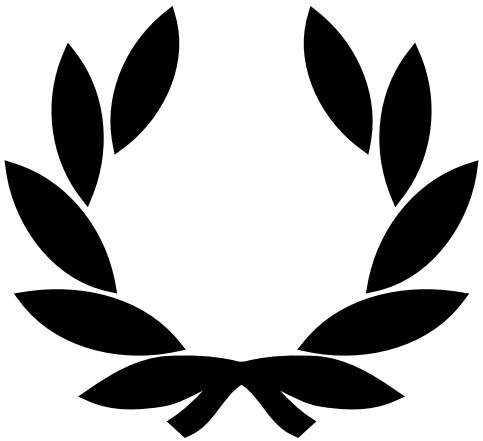
Authenticated encryption?

```
int crypto_aead_encrypt(
    unsigned char *c, unsigned long long *clen,
    const unsigned char *m, unsigned long long mlen,
    const unsigned char *ad, unsigned long long adlen,
    const unsigned char *nsec,
    const unsigned char *npub,
    const unsigned char *key
);
```

Intro to authenticated encryption



Tested AE schemes



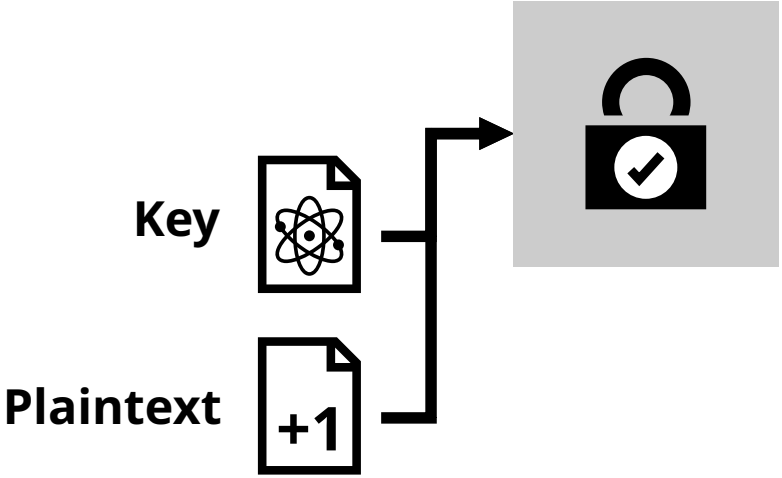
CAESAR

Competition for Authenticated Encryption: Security, Applicability and Robustness

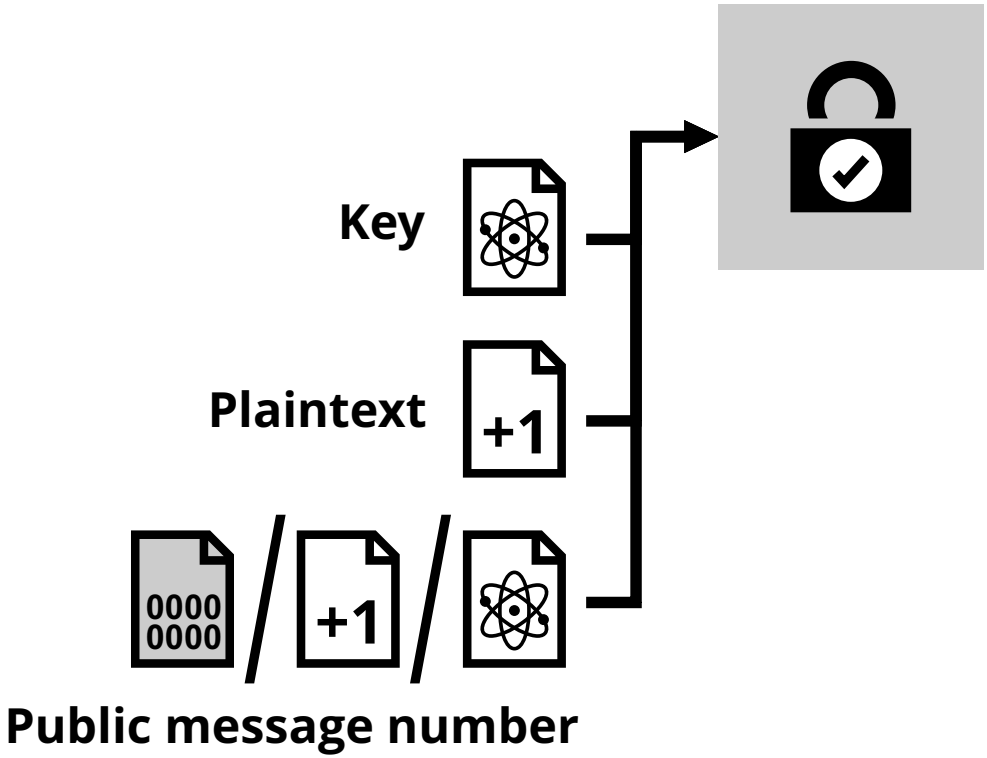
(EPFL, KU Leuven, Inria, Uni. of California, Vodafone, STMicroelectronics, ...)

56 designs (172 shemes) in 1st round

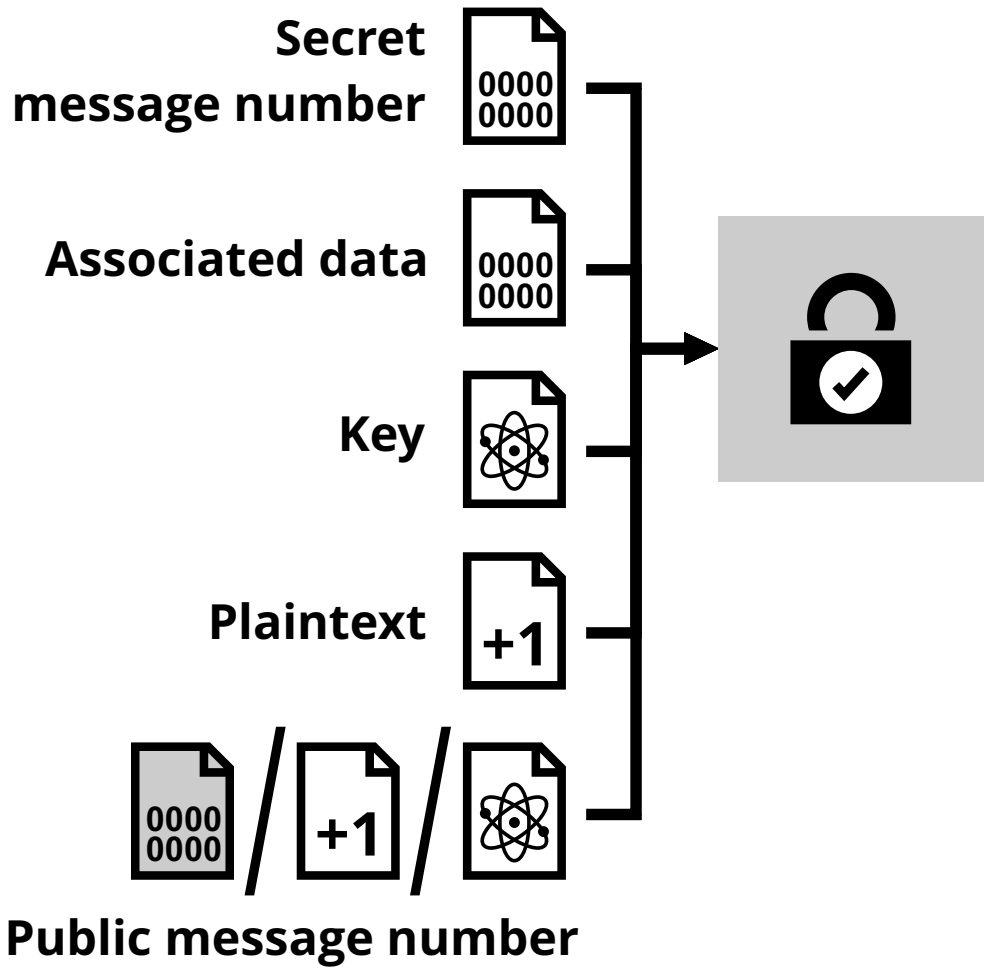
Generating stream of data



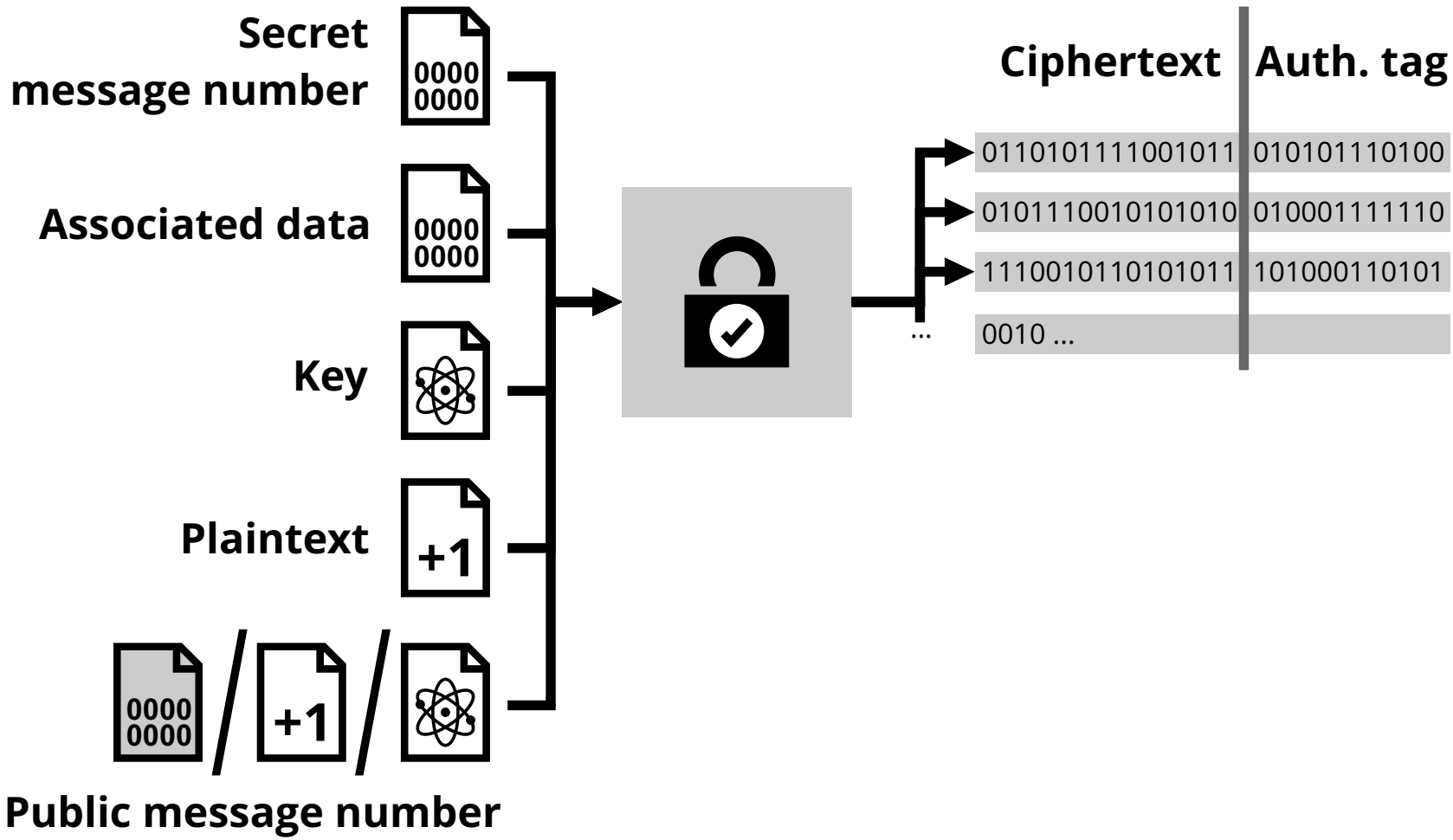
Generating stream of data



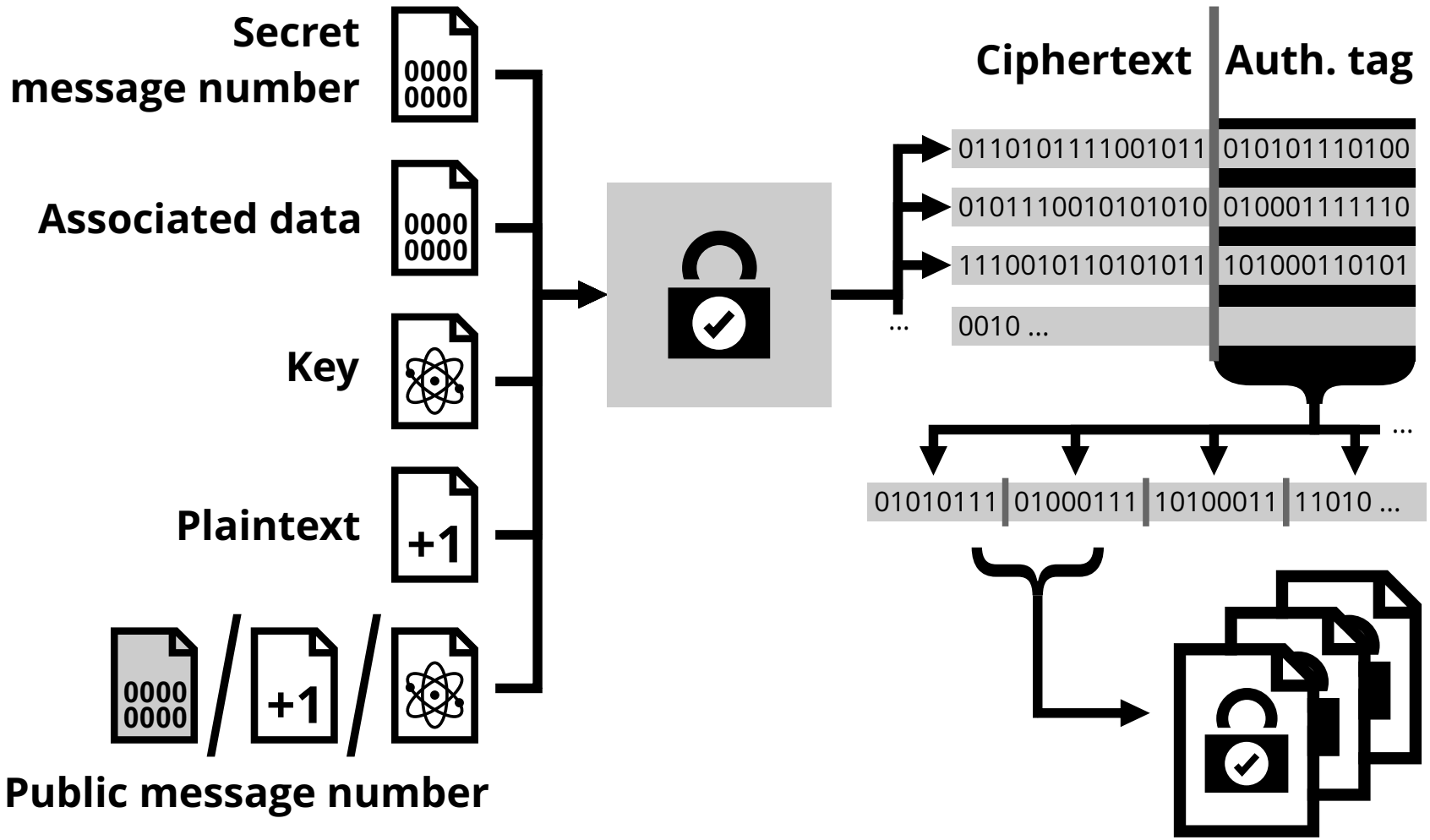
Generating stream of data



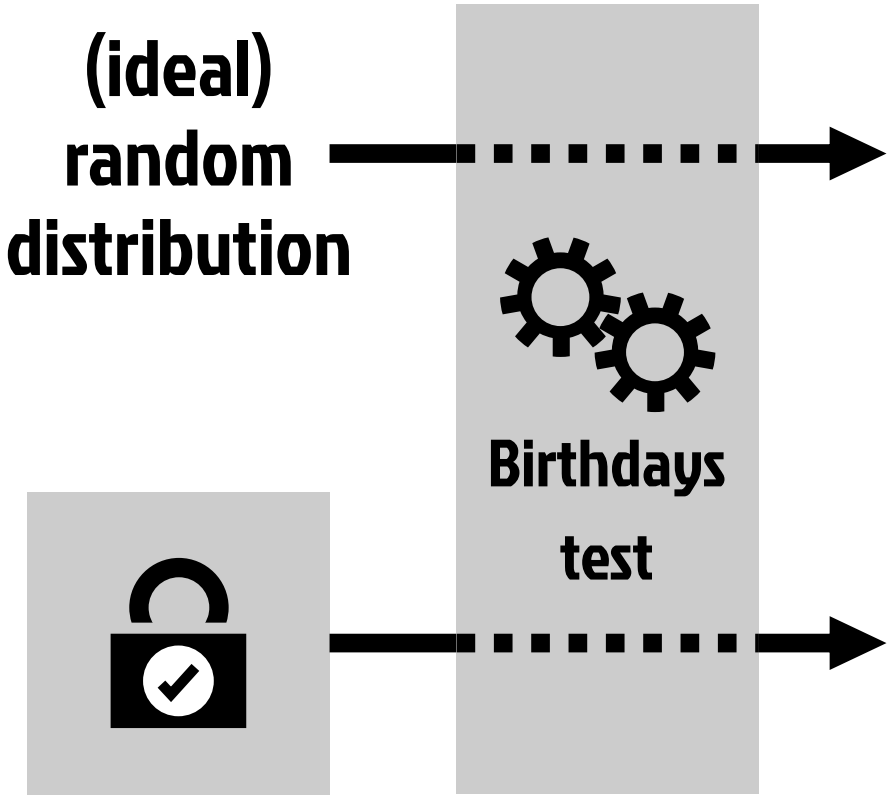
Generating stream of data



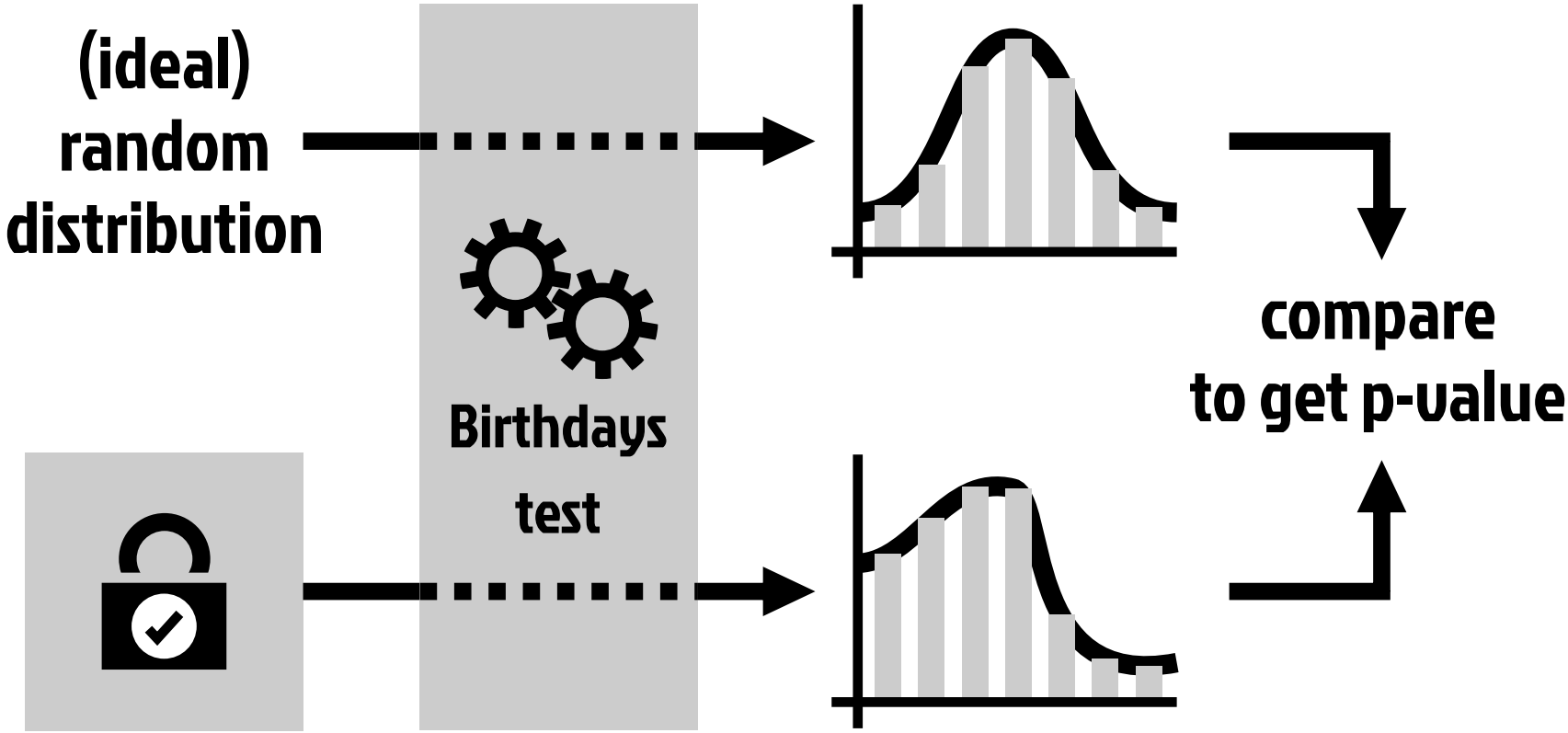
Generating stream of data



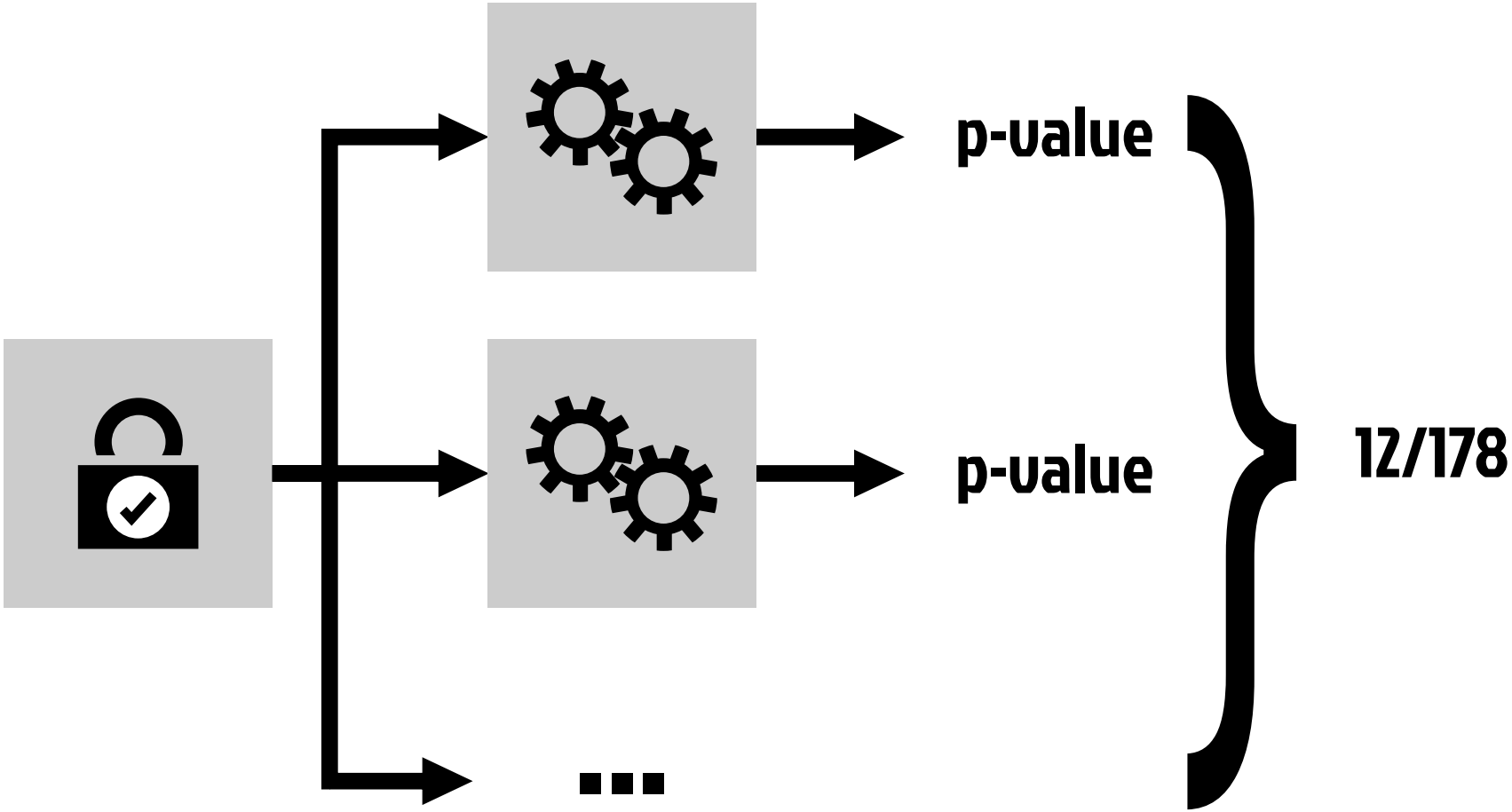
Statistical testing of randomness



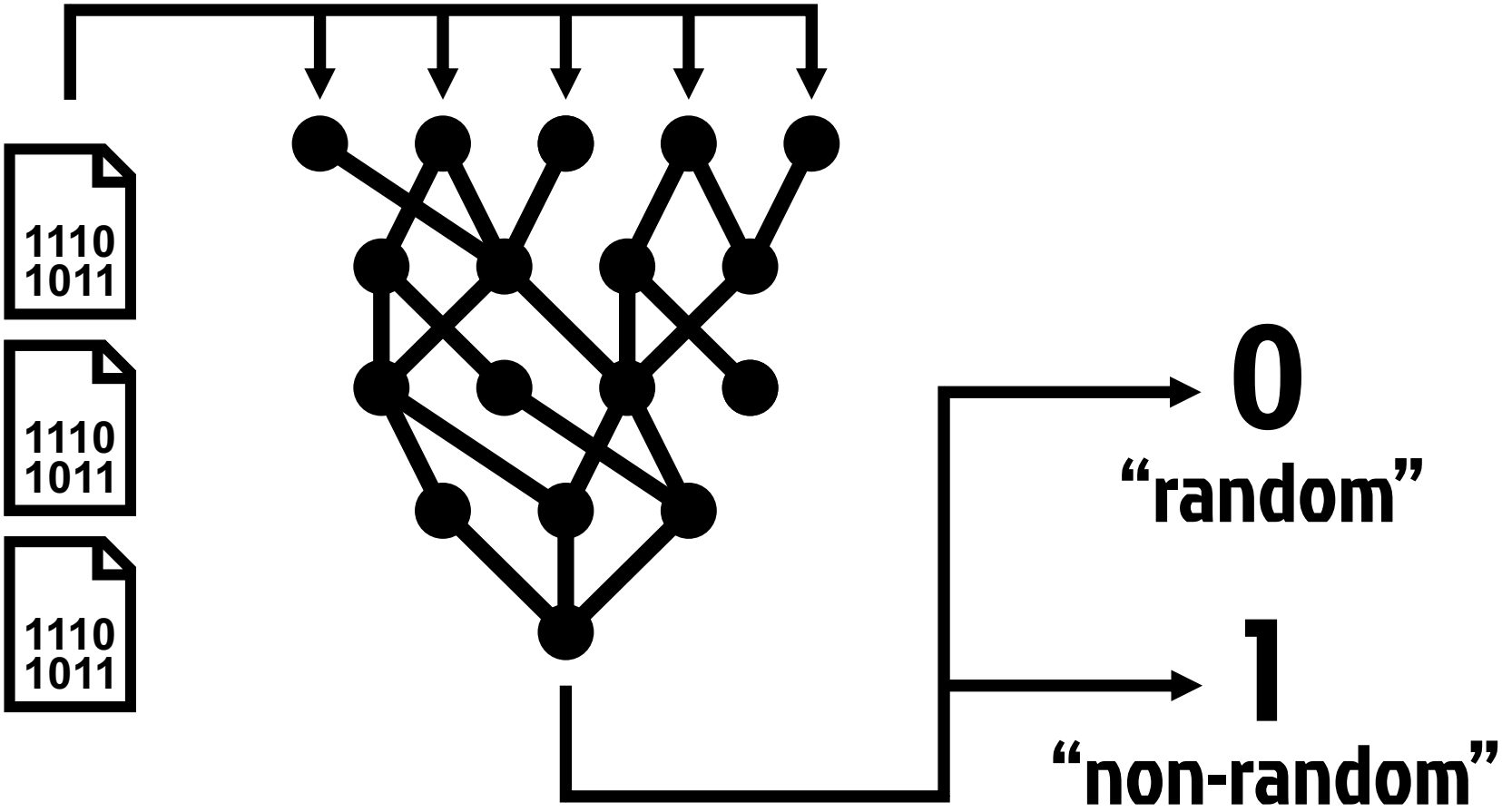
Statistical testing of randomness



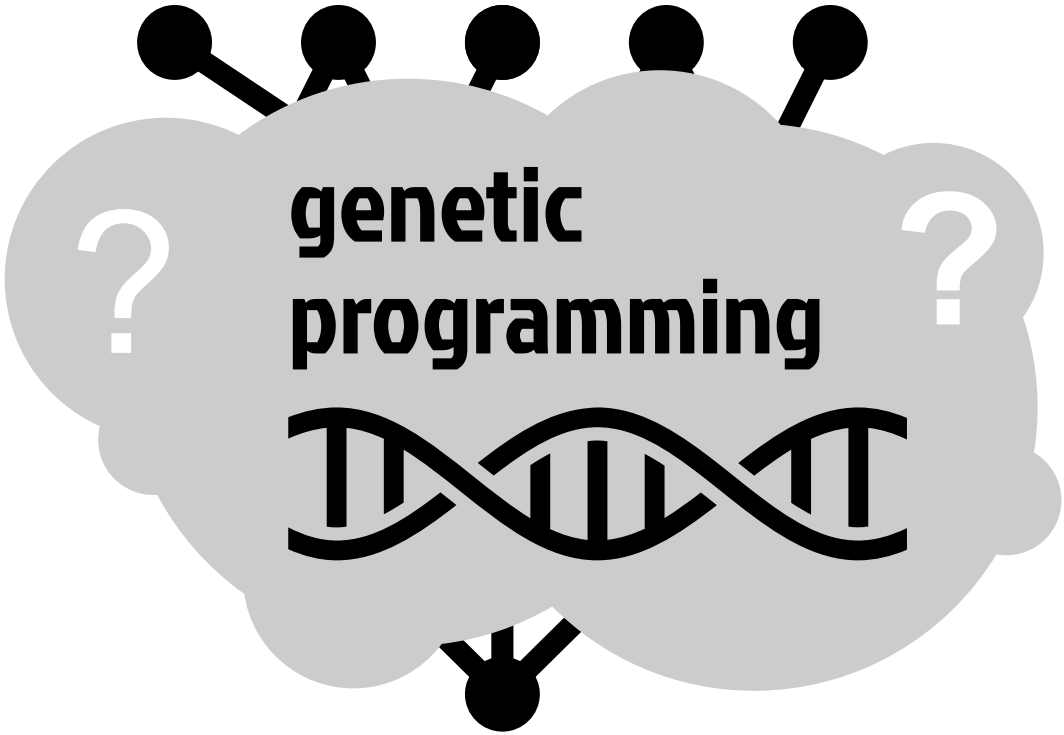
Statistical batteries



Distinguish randomness with EACirc



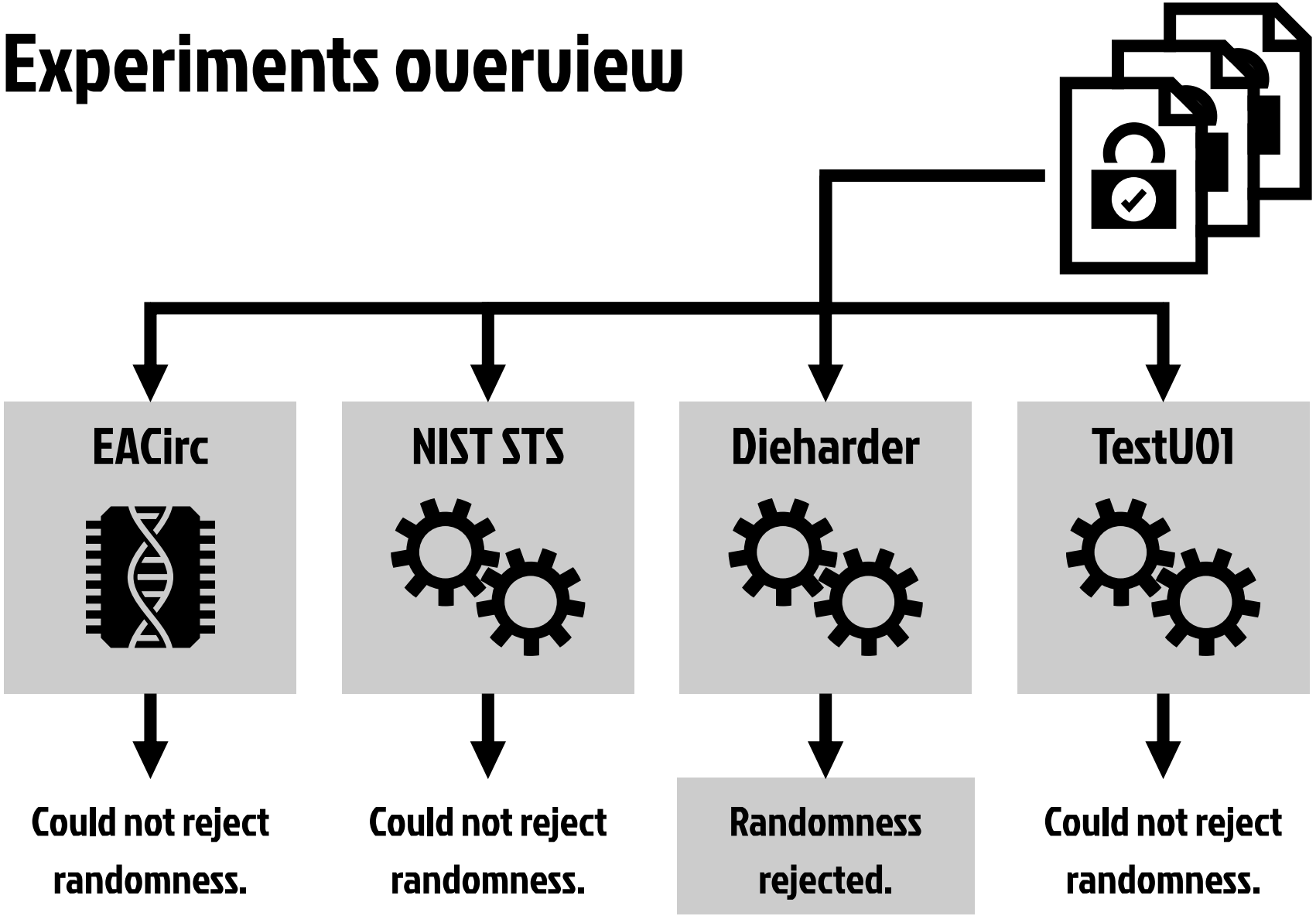
...using evolutionary algorithms



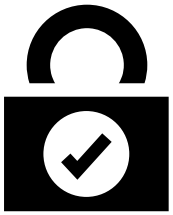
EACirc

Fork me on GitHub!
github.com/crocs-muni/EACirc

Experiments overview



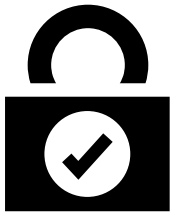
Tested scenarios



Tested data streams

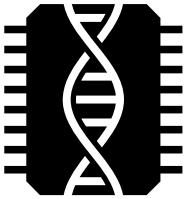
- authentication tags from CAESAR candidates
- 52/56 designs tested (168/172 schemes)
- 1 reference design (AES/GCM)
- 3 different public message number modes

Tested scenarios



Tested data streams

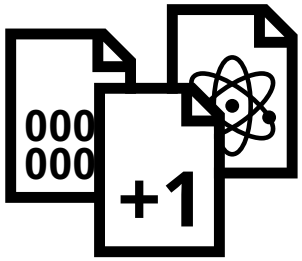
- authentication tags from CAESAR candidates
- 52/56 designs tested (168/172 schemes)
- 1 reference design (AES/GCM)
- 3 different public message number modes



Randomness assessment tools

- statistical batteries (NIST STS, Dieharder, TestU01)
- genetically inspired framework (EACirc)

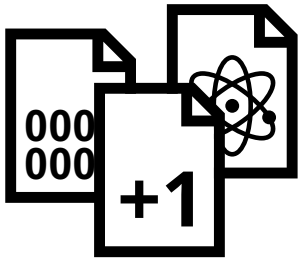
Results interpretation



Public message number modes

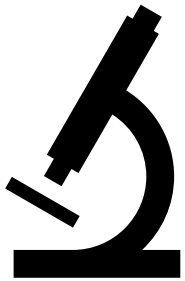
- zero: all candidates failed
- counter: Marble, AES-CMCC, Raviyoyla
- random: Marble, (AES-CMCC), ((Raviyoyla))

Results interpretation



Public message number modes

- zero: all candidates failed
- counter: Marble, AES-CMCC, Ravivoyla
- random: Marble, (AES-CMCC), ((Ravivoyla))



High-level observations

- as expected: zero < counter < random
- worse candidates not in further rounds of CAESAR
- statistical tools generally comparable
- EACirc significantly worse (1 exception)

Study limitations

Testing stream length

- NIST STS: 12MiB (~700 000 tags)
- Dieharder: 916 MiB (~60 000 000 tags)
- TestU01: 128 MiB (~8 400 000 tags)
- EACirc: 2.24 GiB (~150 000 000 tags)

Not inspected

- ciphertext randomness
- other plaintext modes
- ...

Definitions of good security

Semantic security

- No probabilistic polynomial-time algorithm with ciphertext has advantage to PPTAs having only message length
- No plaintext info can be <u>feasibly</u> extracted from ciphertext

Perfect secrecy

- information-theoretical analogy to sematic security
- ciphertext reveals no information about plaintext

Chosen-plaintext attack ciphertext indistinguishability

- equivalent to semantic security (Goldwasser and Micali, 1982)

Conclusion

“Not even state-of-the-art AE schemes have avalanche effect strong enough for zero-initialized PMN mode.”

no direct practical attack

yet semantic security broken

Thank you!

The avalanche of questions is very welcome.

Pre-print, presentation and all data avaliable at crcs.cz/papers/memics2016