Comparison of 256-bit stream ciphers

D. J. Bernstein

Thanks to:
University of Illinois at Chicago
Denmark Technical University
Alfred P. Sloan Foundation
Comparison of 256-bit stream ciphers

D. J. Bernstein

Thanks to:
University of Illinois at Chicago
Denmark Technical University
Alfred P. Sloan Foundation

Cipher implementations from cipher authors

Timing tools (De Cannière)

Timings on various machines

Graphing tools (Bernstein)

Speed graphs in this talk

Security disasters

Attack claimed on YAMB: “2^{58}.”
Attack claimed on Py: “2^{72}.”
Presumably also Py6.
Attack claimed on SOSEMANUK: “2^{226}.”

Is there any dispute about these attacks?
If not: Reject YAMB etc. as competition for 256-bit AES.
Cipher implementations from cipher authors

Timing tools (De Cannière)

Timings on various machines

Graphing tools (Bernstein)

Speed graphs in this talk

Security disasters

Attack claimed on YAMB: “2^{58}.”

Attack claimed on Py: “2^{72}.” Presumably also Py6.

Attack claimed on SOSEMANUK: “2^{226}.”

Is there any dispute about these attacks? If not: Reject YAMB etc. as competition for 256-bit AES.
Cipher implementations from cipher authors

↓↓

Timing tools (De Cannière)

↙↙↙ ↙ ↙ ↙

Timings on various machines

↓↓

Graphing tools (Bernstein)

↙↙↙ ↙ ↙ ↙

Speed graphs in this talk

---

Security disasters

Attack claimed on YAMB: “$2^{58}$.”

Attack claimed on Py: “$2^{72}$.”

Presumably also Py6.

Attack claimed on SOSEMANUK: “$2^{226}$.”

Is there any dispute about these attacks?

If not: Reject YAMB etc. as competition for 256-bit AES.

---

Speed disasters

FUBUKI is slower than AES in all of these benchmarks.

Any hope of faster FUBUKI?

If not: Reject FUBUKI.

VEST is extremely slow in all of these benchmarks.

On the other hand, VEST is claimed to be faster in hardware.

Security disasters

Attack claimed on YAMB: “2^{58}.”

Attack claimed on Py: “2^{72}.”
Presumably also Py6.

Attack claimed on SOSEMANUK: “2^{226}.”

Is there any dispute about these attacks?
If not: Reject YAMB etc. as competition for 256-bit AES.

Speed disasters

FUBUKI is slower than AES in all of these benchmarks.
Any hope of faster FUBUKI?
If not: Reject FUBUKI.

VEST is extremely slow in all of these benchmarks.
On the other hand, VEST is claimed to be faster in hardware.
Security disasters

Attack claimed on YAMB: “$2^{58}$."

Attack claimed on Py: “$2^{72}$."

Presumably also Py6.

Attack claimed on SOSEMANUK: “$2^{226}$.”

Is there any dispute about these attacks?

If not: Reject YAMB etc. as competition for 256-bit AES.

Speed disasters

FUBUKI is slower than AES in all of these benchmarks.

Any hope of faster FUBUKI?

If not: Reject FUBUKI.

VEST is extremely slow in all of these benchmarks.

On the other hand, VEST is claimed to be faster in hardware.

Remaining 256-bit ciphers: CryptMT, DICING, Dragon, HC-256, Phelix, Salsa20.

Could say, e.g., “CryptMT is practically always slower than Phelix and should be eliminated;” but what if Phelix is broken?

Attacks on Py, SOSEMANUK were published in December.

Need more time for cryptanalysis.
Speed disasters

FUBUKI is slower than AES in all of these benchmarks. Any hope of faster FUBUKI? If not: Reject FUBUKI.

VEST is extremely slow in all of these benchmarks. On the other hand, VEST is claimed to be faster in hardware.

Remaining 256-bit ciphers: CryptMT, DICING, Dragon, HC-256, Phelix, Salsa20.

Could say, e.g., “CryptMT is practically always slower than Phelix and should be eliminated”; but what if Phelix is broken?

Attacks on Py, SOSEMANUK were published in December. Need more time for cryptanalysis.
FUBUKI is slower than AES in all of these benchmarks. 
Any hope of faster FUBUKI? If not: Reject FUBUKI.
VEST is extremely slow in all of these benchmarks. On the other hand, VEST is claimed to be faster in hardware.

Remaining 256-bit ciphers:
CryptMT, DICING, Dragon, HC-256, Phelix, Salsa20.

Could say, e.g., "CryptMT is practically always slower than Phelix and should be eliminated"; but what if Phelix is broken?

Attacks on Py, SOSEMANUK were published in December. Need more time for cryptanalysis.

Speedup: security margin
Can speed up AES by reducing rounds from 14 to, e.g., 10. No known attacks.
Can speed up Salsa20 by reducing rounds from 20 to, e.g., 12 or 8. No known attacks.

Do any other submissions have a security margin?
Remaining 256-bit ciphers: CryptMT, DICING, Dragon, HC-256, Phelix, Salsa20.

Could say, e.g., “CryptMT is practically always slower than Phelix and should be eliminated”; but what if Phelix is broken?

Attacks on Py, SOSEMANUK were published in December. Need more time for cryptanalysis.

Speedup: security margin

Can speed up AES by reducing rounds from 14 to, e.g., 10. No known attacks.

Can speed up Salsa20 by reducing rounds from 20 to, e.g., 12 or 8. No known attacks.

Do any other submissions have a security margin?
Remaining 256-bit ciphers:
CryptMT, DICING, Dragon,
HC-256, Phelix, Salsa20.

Could say, e.g.,
"CryptMT is practically always
slower than Phelix
and should be eliminated";
but what if Phelix is broken?
Attacks on Py, SOSEMANUK
were published in December.
Need more time for cryptanalysis.

Speedup: security margin
Can speed up AES
by reducing rounds
from 14 to, e.g., 10.
No known attacks.

Can speed up Salsa20
by reducing rounds
from 20 to, e.g., 12 or 8.
No known attacks.

Do any other submissions
have a security margin?

Slowdown: forgeries
Packets must be authenticated.
State of the art: Poly1305,
around 4 cycles per byte
plus encrypting 16 bytes.

Fastest encryption implies
fastest authenticated encryption?
Not necessarily!
Phelix includes authentication.
Benchmarks need to cover this.
Speedup: security margin

Can speed up AES by reducing rounds from 14 to, e.g., 10. No known attacks.

Can speed up Salsa20 by reducing rounds from 20 to, e.g., 12 or 8. No known attacks.

Do any other submissions have a security margin?

Slowdown: forgeries

Packets must be authenticated. State of the art: Poly1305, around 4 cycles per byte plus encrypting 16 bytes.

Fastest encryption implies fastest authenticated encryption? Not necessarily! Phelix includes authentication. Benchmarks need to cover this.
**Speedup: security margin**

- AES
  - Can speed up by reducing rounds from 14 to, e.g., 10.
  - No known attacks.
- Salsa20
  - Can speed up by reducing rounds from 20 to, e.g., 12 or 8.
  - No known attacks.

Do any other submissions have a security margin?

---

**Slowdown: forgeries**

- Packets must be authenticated.
- State of the art: Poly1305, around 4 cycles per byte plus encrypting 16 bytes.
- Fastest encryption implies fastest authenticated encryption? Not necessarily!
- Phelix includes authentication.
- Benchmarks need to cover this.

**Slowdown: timing attacks**

- Typical AES software leaks key through timing.
- Often attacker can see timing.
- Constant-time AES software is considerably slower.
- Slowdown depends on cipher.
  - CryptMT, Phelix, Salsa20: 0.
  - DICING, Dragon, HC-256: ?
- Benchmarks need to cover this.
Slowdown: forgeries

Packets must be authenticated.

State of the art: Poly1305, around 4 cycles per byte plus encrypting 16 bytes.

Fastest encryption implies fastest authenticated encryption? Not necessarily!
Phelix includes authentication.
Benchmarks need to cover this.

Slowdown: timing attacks

Typical AES software leaks key through timing. Often attacker can see timing.

Constant-time AES software is considerably slower.

Slowdown depends on cipher. CryptMT, Phelix, Salsa20: 0. DICING, Dragon, HC-256: ?

Benchmarks need to cover this.