Stream Ciphers: Past, Present and Future

Prof. Bart Preneel
Katholieke Universiteit Leuven, COSIC
http://www.ecrypt.eu.org/

ISCISC 2010
15 September 2010

Thanks to Joseph Lano

Outline

- Stream ciphers
- Attacks on stream ciphers
- eSTREAM Project
- Perspectives

Stream ciphers: history

- 1917 one-time pad
- 1930's Hagelin
- 1940's Enigma
- 1960's LFSRs
- 1977 **DES**
- 1978 Triple-DES
- 1987 RC4
- 1988 A5/1, A5/2
- 1987 FEAL
- 1990 Khufu/Khafre/IDEA
- 1992 SEAL
- 199? CRYPTO-1
- 1998 Panama
- 1999 E0, SNOW
- 1999 KASUMI
- 2000 **AES**
- 2001 MUGI
- 2006 SNOW 3G
- 2010 ZUC

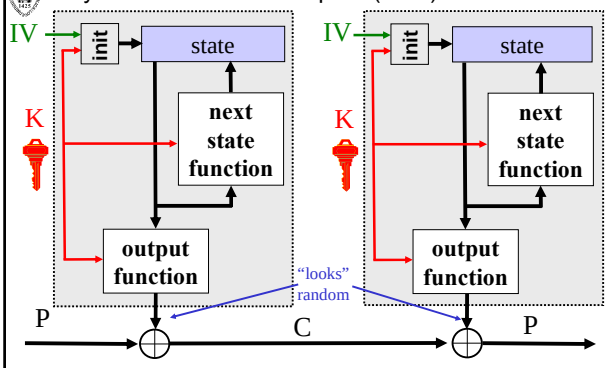
2004
2008 **estream**

FISH, ISAAC, PIKE, Shrinking generator, VEST, WAKE,….

Stream ciphers: definitions

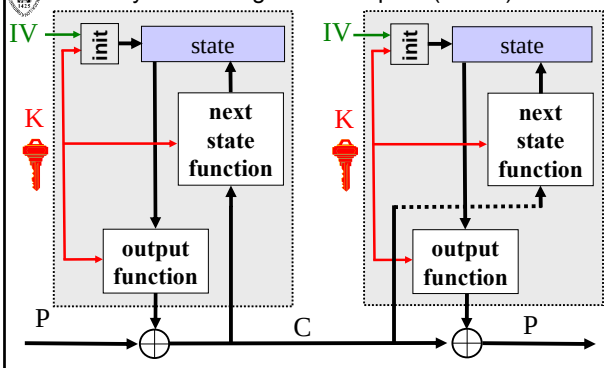
- No clear definitions but main characteristics can be identified
  - Operate on small words (bits, bytes, 32-bit words)
  - Internal state
  - Simple
- Continuum between stream ciphers and block ciphers

Synchronous Stream Cipher (SSC)



k: key size – v: IV size – s: total state size

Self-Synchronising Stream Cipher (SSSC)



k: key size – v: IV size – s: total state size

Specific properties of an SSC

- Cryptographically secure pseudo-random bit generator (PRBG)
- No error-propagation
- Resynchronization mechanism needed
- Particular model for cryptanalysis
  - Key stream independent of data

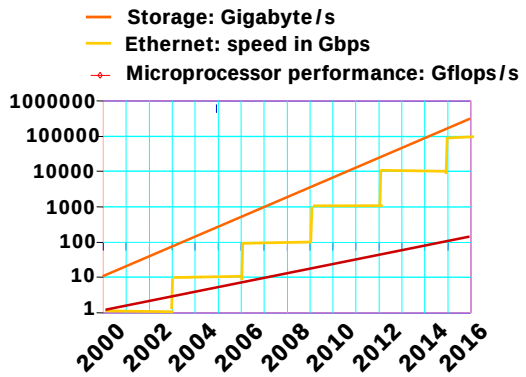
7

Advantages of stream ciphers

- Software:
  - Up to five times faster than AES
  - Useful in high throughput applications
- Hardware
  - Use less area than AES
    - < 2 kGate compared to > 3 kGate for AES
  - Higher throughput/area possible
  - Useful in restricted environments: RFID, ..

8

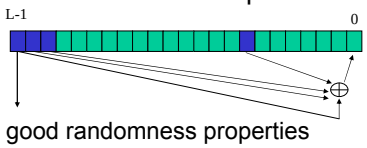
Moore's Law: computation/storage



9

Types of SSC

- LFSR based stream ciphers



+ good randomness properties
+ mathematical theory
+ maximum period (2^L-1)
+ compact in hardware
- too linear: easy to predict

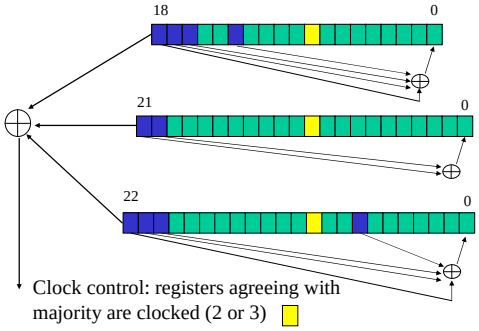
10

Types of SSC (2)

- Destroy linearity of LFSRs with
  - Boolean output function
    - E.g. Sober-t
  - additional non-linear memory
    - E.g. summation generator, SNOW 2.0, E0 (Bluetooth)
  - irregular clocking
    - E.g. A5/1 (GSM), shrinking generator, self-shrinking generator

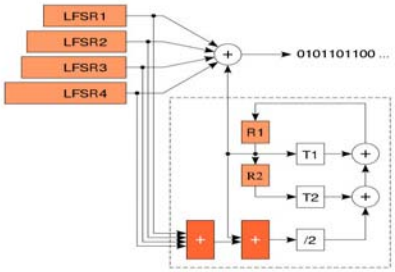
11

Example: A5/1 (GSM)



12

Example: E0 (Bluetooth)



brute force: 2^{128} steps
[Lu+05] 24 known bits of 2^{24} frames, 2^{38} computations, 2^{33} memory

13

Types of SSC (3)

- Problem with LFSR-based designs:
  - Algebraic attack and fast algebraic attack
    - [Courtois03] and [Courtois-Meier03]
  - ...
- Precluding algebraic attacks
  - **irregular** clocking
  - **nonlinear** update functions
    - Rationale: keep nonlinearity in the state, don't just introduce it in the output

14

Types of SSC (4)

- Huge variety of design strategies for **nonlinear update** mechanisms
  - Shift registers with non-linear feedback: Grain, Trivium
  - random shuffles: RC4, PY, HC-128
  - T-functions
  - FCSR (feedback with carry)
- Security model of most of these designs not well understood
  - more research needed
  - need for wild and creative ideas: think out of the box
- Block cipher based techniques:
  - SEAL, SCREAM, MUGI, Phelix, Salsa,...

15

A simple cipher: RC4

- designed by Ron Rivest (MIT)
- S[0..255]: secret table derived from user key K

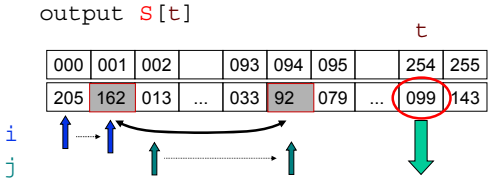
```
for i=0 to 255 S[i]:=i
j:=0
for i=0 to 255
    j:=(j + S[i] + K[i]) mod 256
    swap S[i] and S[j]
i:=0,  j:=0
```

16

A simple cipher: RC4

Generate key stream which is added to plaintext

```
i:=i+1
j:=(j + S[i]) mod 256
swap S[i] and S[j]
t:=(S[i] + S[j]) mod 256
output S[t]
```



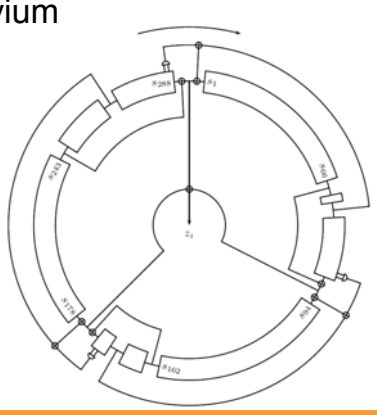
17

RC4: weaknesses

- was frequently used with 40-bit key
  - US export restrictions until Q4/2000
- best known general shortcut attack: 2^{241}
- weak keys and key setup (shuffle theory)
- some statistical deviations
  - 2nd output byte is biased (solution for this bias: drop first 256 bytes of output)
- problem with resynchronization modes (WEP)

18

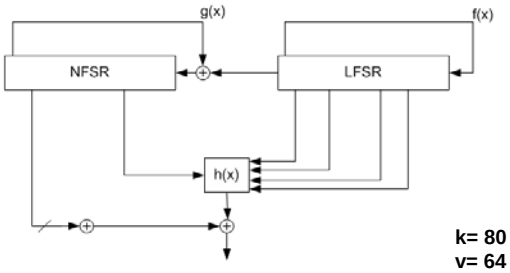
Trivium



k= 80
v= 80
s=288

19

Grain



k= 80
v= 64
s=180

20

Overview of the Presentation

- Stream ciphers
- Attacks on stream ciphers
- eSTREAM Project
- Perspectives

21

Attacks on stream ciphers

- Brute force attacks (k,v,s) size of (key, IV, state)
  - Exhaustive key search
  - Exhaustive state search
  - Time-Memory Tradeoffs
- Shortcut attacks
- Side channel attacks: active and passive

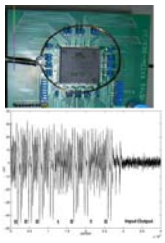
22

Shortcut attacks

- Statistical attacks:
  - (fast) correlation attacks [Meier-Staffelbach89]
  - linear attacks [Golic94]
  - χ^2 attacks
- Guess and determine attacks
- Algebraic attacks [Courtois03,Meier89]

23

Side channel attacks



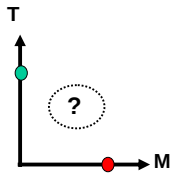
- Very powerful
- Designs to the edge
- Not surprising that these attacks are very effective in particular against **unprotected** implementations

- Realistic? Plaintext is available in encrypter and decrypter
- Specific scenarios: RFID cloning, DRM, remote (electromagnetic) attacks

24

Brute force attacks

- **Traditional key-recovery attack**
  - Time T = 2^{k-1} on average
  - Data D = negligible
  - Memory M = negligible
  - Precomputation P = negligible
- **Table-based key-recovery attack**
  - Time T = 1 step
  - Data D = negligible
  - Memory M = 2^k k-bit values
  - Precomputation P = 2^k
- **Trade-off attack**
  - Trade some T complexity for D, M and/or P complexity



25

Classical TMD trade-offs against any one-way function (including any stream cipher)

- Aim: given one key stream, recover the secret key (k bits)
- [Hellman80]
  - $T = 2^{2k/3}$, $M = 2^{2k/3}$, D=1 (k bits) and $P= 2^k$
  - Note that $T * M = 2^{4k/3}$

26

Classical TMD trade-offs against stream ciphers with unkeyed output transformation

Aim: given one key stream, recover s bits of internal state

- Birthday [Babbage-Golic]
  - $T * M = 2^s$, D=T and P=M
  - Not very flexible
- Extended birthday [Biryukov-Shamir00]
  - $T * M^2 * D^2 = 2^{2s}$, $T>D^2$ and $P=2^s/D$
  - More flexible trade-off between time, memory and data

27

Classical TMD trade-offs (2)

- Classical TMD on a stream cipher can be easily prevented by making the state size s larger than twice the key size k
  - Note: eSTREAM candidates not vulnerable to classical TMD trade-offs

28

TMK trade-offs

- time-memory-key tradeoffs [Biham96-02] [Biryukov+05], [Hong-Sarkar05]
- Attack scenario
  - Attacker intercepts a large number D_k of short key streams, coming from different keys and same IV
  - Attacker wants to recover only one of those keys
  - Realistic scenario in e.g. broadcasting one message to many users

29

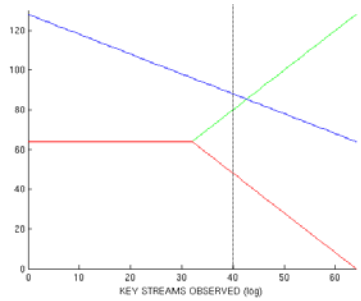
TMK trade-offs (2)

- Same trade-offs as in classical TMD, but we attack the key directly, instead of the state
- Birthday [Babbage-Golic]
  - $T * M=2^k$, $T=D_k$ and P=M
- Extended birthday [Biryukov-Shamir]
  - $T * M^2 * D_k^2 = 2^{2k}$, $T>D_k^2$ and $P=2^k/D_k$
- impact on the eSTREAM key sizes (128 bit, 80 bit)

30

TMK trade-offs (3): 128 bit key

log Time
log Memory
log Precomputation

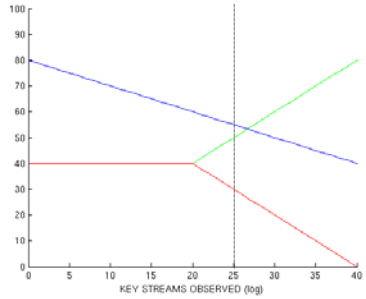


Example: $D_k=2^{40}$, then $T=2^{80}$, $M=2^{48}$, $P=2^{88}$

31

TMK trade-offs (4): 80 bit key

log Time
log Memory
log Precomputation



Example: $D_k=2^{25}$, then $T=2^{50}$, $M=2^{30}$, $P=2^{55}$

32

TMK trade-offs (5)

- This can be a problem in some cases
- Solution?
  - TMK trade-offs work because the only entropy we introduce into the state comes from the secret key
  - Solution is therefore simple: add more entropy to the initialization, either by:
    - Using full-entropy IVs [Biryukov+05]
    - Using longer keys [Bernstein06]

33

TMK trade-offs (6)

- Pros and cons of both approaches

Longer keys	Full-entropy IV
- Secure distribution and storage is needed	- Generation and transmission is necessary
- Security level differs from key length	+ correspondance between key length and security level is maintained
+ Brute-force attacks are made more expensive	- Only provides protection against trade-off attacks
- Resynchronization may become much more delicate (slide attacks) when k+v gets close to s	

34

TMK trade-offs (7)

- How much entropy is needed?
  - Can be easily calculated from formulae
  - Attack scenario 1: T,M,D,P all < 2^k
    - Can be avoided by adding entropy equivalent to ½ of the security level
  - Attack scenario 2: T,M,D all < 2^k, but we allow P>2^k
    - Can be avoided by adding entropy equivalent to once the security level

35

Cube attack [Dinur-Shamir'08]

- exploits low degree equations in stream cipher
- can break certain ciphers which could not be broken before
- ….media hype and controversy
- relation to higher order attacks (Lai) and AIDA (algebraic IV differential attack) (Vielhaber)

- Trivium:
  - key setup can be broken if number of rounds is reduced from 1024 to 793 (Aida) or 767 (cube)
  - attack may be further improved
  - solution: increase number of rounds of key/IV setup to 2048
- Grain: results announced at Crypto rump session (no details yet)

36

Outline

- Stream ciphers
- Attacks on stream ciphers
- eSTREAM Project
- Perspectives

37

ECRYPT basic facts

- European Network of Excellence in Cryptology
- ECRYPT: Feb 2004 – July 2008
- ECRYPT II: Aug 2008 – July 2010
- Total funding: 8.6 M€
- Very limited effort for person-months – mostly networking activities

38

Cryptographic competitions

- US
  - two calls for block cipher in 70ies, resulting in DES
  - AES (http://www.nist.gov/aes) (1997-2001)
  - SHA-3 (2007-2012) (http://csrc.nist.gov/groups/ST/hash/sha-3/index.html)
- Japan: CRYPTREC (http://www.ipa.go.jp/security)
- Europe
  - RIPE (Race Integrity Primitives Evaluation) (1989-1992)
  - NESSIE (2000-2003) (http://www.cryptonessie.org)
  - eSTREAM (2004-2008)

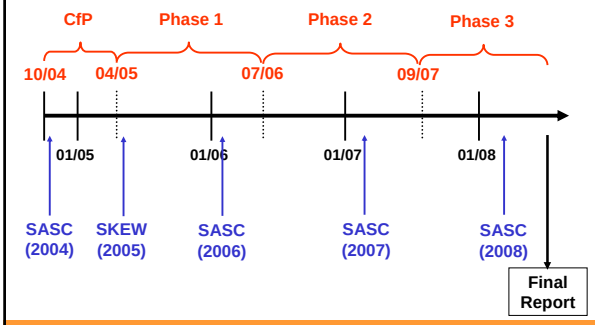
39

eSTREAM Project

- Project within ECRYPT (2004-2008)
- Goals:
  - Identify a set of promising stream ciphers
  - Promote a greater understanding of stream cipher design and analysis
- Differences compared to AES/SHA-3 competition
  - *eSTREAM is not a standardisation effort*
  - We offered the designers more flexibility with regards to "tweaking" because we did not want to miss out on good design ideas because of an easily avoided flaw

40

eSTREAM timeline



41

NESSIE Stream ciphers

- ~~BMGL* (Ericsson, KTH Stockholm)~~
- ~~Leviathan (Cisco Systems Inc.)~~
- ~~LILI-128 (Queensland Univ. of Technology, Univ. of Belgrade)~~
- ~~SNOW* (Lund Univ.)~~
- ~~Sobert-t16 (Qualcomm Int.)~~
- ~~Sobert-t32 (Qualcomm Int.)~~

42

Call for Primitives

- eSTREAM solicits stream ciphers for two types of environments
  - PROFILE 1
    - Stream ciphers for high throughput SW applications
    - 128-bit key, 64 and 128-bit IVs
  - PROFILE 2
    - Stream ciphers for HW with restricted resources
    - 80-bit key, 32-bit and 64-bit IVs
- Associated authentication mechanisms will also be studied (tag lengths from 32 to 128 bits)

43

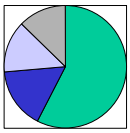
CfP: Evaluation criteria

- Security
- Performance
  - when compared to the AES
  - when compared to other submissions
- Justification and supporting analysis
- Simplicity and flexibility
- Completeness and clarity of submission

44

CfP: results

- 34 submissions
  - 31 SSC and 2 SSSC
  - 9 SW, 12 HW, 13 SW+HW
  - 20 countries: Australia, Belgium, France, Sweden, Germany, USA, Canada, Russia,...
  - 10 LFSR, 6 NLFSR, 3 T-functions, 2 SPN
  - several combine multiple ideas
  - 7 offer an authentication method
  - many wild ideas…



let the games begin

45

Submission results

- 22/34 broken in Phase 1

PROFILE I	PROFILE I+II	PROFILE II
ABC	F-FCSR	Achterbahn
CryptMT	Hermes8	DECIM
DICING	LEX	EDON-80
DRAGON	MAG	Grain
FROGBIT	NLS	MICKEY (128)
HC-256	Phelix	MOSQUITO
Mir-1	Polar Bear	SFINKS
Py	POMARANCH	Trivium
SOSEMANUK	Rabbit	TSC-3
	Salsa20	VEST
	SSS	WG
	TRBDK3 YAEA	ZK-Crypt
	Yamb	

46

Phase 2 algorithms (many tweaked)

Profile I	Profile II	
ABC v3	Achterbahn-128/80	POMARANCH v3
CryptMT v3	DECIM v2	Rabbit
DICING (P2)	EDON-80	Salsa20
DRAGON	Grain v1	Trivium
HC-128 and HC-256	F_FCSR-H v2	TSC-4
LEX	Hermes8	Vest (P2) A
NLS v2 A	LEX	WG (P2)
Phelix A	Mickey v2	ZKcrypt (P2) A
Polar Bear v2	Mickey-128 v2	
PY	MOUSTIQUE	
Rabbit	NLS v2 A	
Salsa20	Phelix A	
SOSEMANUK	Polar Bearv2	

focus **Authenticated encryption**

47

Phase 3 algorithms

SW Phase 3	HW Phase 3
CryptMT*	**DECIM v2***
DRAGON	**Edon-80***
HC-128 (-256)	**F-FCSR-H v2***
LEX	**Grain v1**
NLS (encrypt only)	**MICKEY-128 v2**
Rabbit*	**MOUSTIQUE***
Salsa20	**POMARANCH v3***
SOSEMANUK	**Trivium**

Faster than AES-CTR 128-bit More compact than AES 80-bit

*** not focus in Phase 2**

All algorithms are the most recent version

48

Phase 3

- No algorithms with authenticated encryption left
- Limited cryptanalytic results
  - EDON-80
  - LEX
  - MOUSTIQUE
  - POMARANCH
- Performance played an important role in final choice

49

The eSTREAM Portfolio

(http://www.ecrypt.eu.org/stream)

Software	Hardware
HC-128	~~F-FCSR-H v2~~
Rabbit	Grain v1
Salsa20/12	MICKEY v2
Sosemanuk	Trivium

(In alphabetical order)

Goal: promote research, not one standard

50

STVL Tools and Resources

- Stream Cipher SW Performance Framework
  - www.ecrypt.eu.org/stream/perf/

51

Software performance: Pentium M (cycles/byte)

	40 bytes	576 bytes	1500 bytes	IMIX	Long streams
Copy	2.72	0.61	0.34	0.66	0.22
AES (CTR)	22.7	16.2	16.1	16.6	16.0
SNOW2.0	24.3	5.80	5.28	6.88	4.75
HC-128	768	56.3	23.8	93.1	3.52
Rabbit	22.7	4.85	4.46	5.93	9.94
Salsa20/12	22.1	7.65	7.83	8.71	7.43
Sosemanuk	36.0	10.1	8.60	11.4	5.60

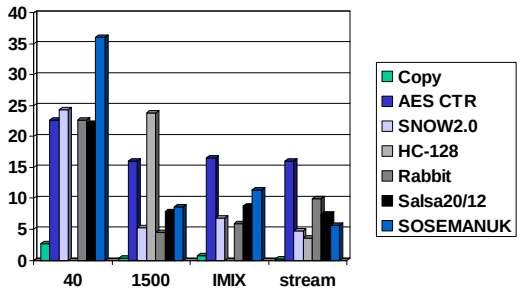
52

Software performance: AMD64 (cycles/byte)

	40 bytes	576 bytes	1500 bytes	IMIX	Long streams
Copy	1.36	0.27	0.28	0.35	0.29
AES (CTR)	18.1	13.4	13.4	13.7	13.4
SNOW2.0	23.2	5.77	5.34	6.81	4.83
HC-128	587	43.2	18.4	71.4	2.86
Rabbit	14.6	5.55	5.25	6.06	4.98
Salsa20/12	12.1	5.00	5.07	5.52	4.85
Sosemanuk	25.3	7.20	6.20	8.03	4.07

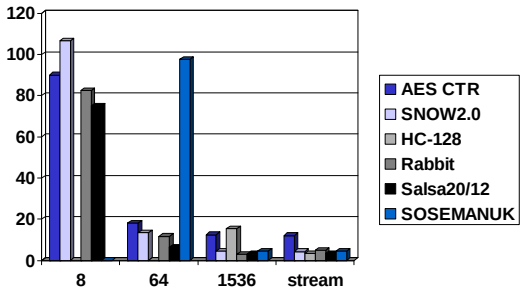
53

Software performance (cycles/byte): Pentium M



54

Software performance (cycles/byte):
AMD Phenom II X6 1055T 2.8 GHz



55

eSTREAM chip
Tim Good (University of Sheffield)

eSCARGOT

European Stream Ciphers are Ready to Go

Full implementation of all the hardware-oriented eSTREAM finalists

Decim v2, Edon-80, F-FCSR-H v2, Grain v1, MICKEY v2, Moustique, Pomaranch v3, Trivium

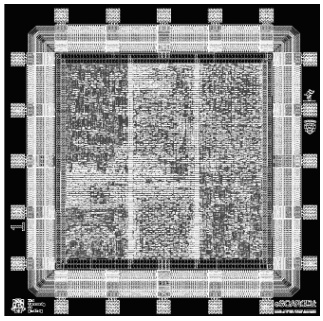
Both the 80-bit and 128-bit key versions

Plus some optimised versions of Grain v1 and Trivium

Samples available : www.shef.ac.uk/eee/escargot

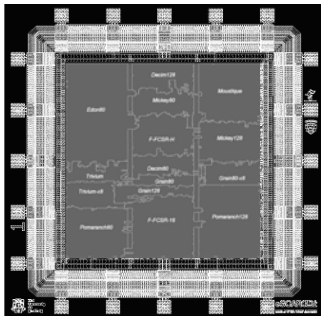
56

eSC@RGO



57

eSC@RGO



58

Hardware performance (100 kHz)

	Throughput (Mbps)	Power (µW)	Area (µm²)	Throughput/area (kbps x µm⁻²)
F-FCSR-H v2	0.8	10.6	25000	0.032
Grain	0.1	3.3	6667	0.015
Grain (x4)	0.4	4.5	8696	0.046
Grain (x8)	0.8	6.1	11429	0.070
Grain (x16)	1.6	9.3	16842	0.095
MICKEYv2	0.1	7.1	25000	0.004
Trivium	0.1	5.5	12500	0.008
Trivium (x4)	0.4	6.0	13793	0.029
Trivium (x8)	0.8	7.0	15385	0.052
Trivium (x16)	1.6	7.4	16326	0.098

59

Software

- HC-128: very fast; large tables so slow setup
- Rabbit: broad performance, oldest algorithm (2003) but no support for large keys and IP encumbered
- SALSA20/12: simple, clean and scalable
- SOSEMANUK: builds on earlier designs, robust security margin

- (SNOW v2): not submitted

60

Hardware

- F-FCSR-H v2: broken
- Grain v1: simple, effective, nice tradeoffs
- Mickey v2: clarity and simplicity; less tradeoffs but perhaps more robust
- Trivium: simple and clear; exceptional hardware performance

61

Outline

- Stream ciphers
- Attacks on stream ciphers
- eSTREAM Project
- Perspectives

62

Future perspectives

- Further work on eSTREAM portfolio
- Lightweight crypto
  - Area? Energy? Power? Throughput?
- Provable secure stream ciphers
- Need authenticated encryption
  - Stream cipher + GCM?
  - Innovative designs: duplex: sponge-based (e.g. Keccack)

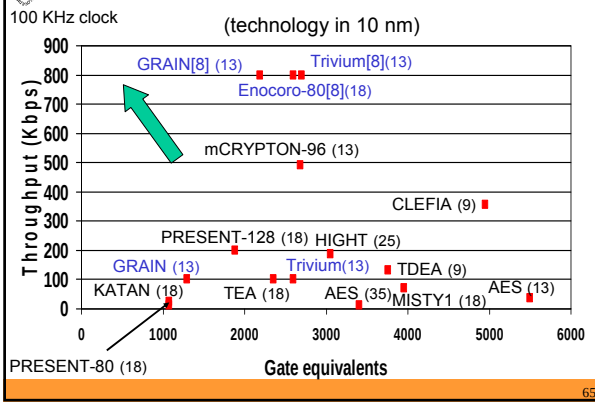
63

AES implementations: efficient/compact

- software
  - 7.6 cycles/byte on Core 2 or 110 Mbyte/s bitsliced [Käsper-Schwabe'09]
- co-processor in Intel Westmere
  - new AES instruction: 0.75 cycles/byte ['09-'10]
- hardware
  - fast 43 Gbit/s in 130 nm CMOS ['05]
  - most compact: 3600 gates

64

Low cost hw: throughput versus area



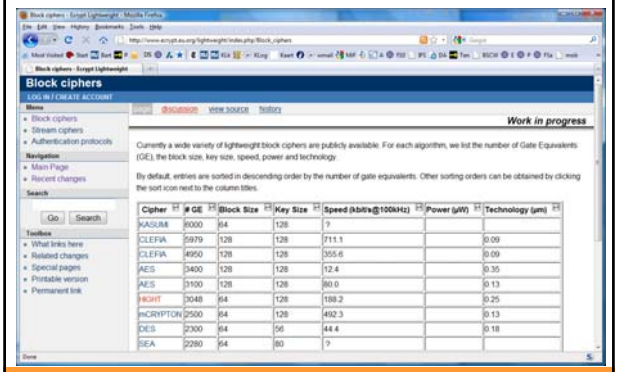
65

Are block ciphers the future for lightweight crypto?

- Comparisons are unfair
  - Technologies different
  - Security is different: birthday attacks for 1 key
    - block cipher with block length of 64 bits: 2^{32}
    - stream cipher with 80-bit key and 160-bit state: 2^{80}
- All novel designs have very small nonlinear component, so the cost is the memory
  - hardware: its costs between 2 and 8 gates to store 1 bit
  - software: RAM usage is critical factor
    - 256 bytes on low-end 8-bit processor (such as PIC10-16, RS08TM, HC08TM, COP8, 80C51TM)
- Stream ciphers offer better throughput per area

66

http://www.ecrypt.eu.org/lightweight/



67

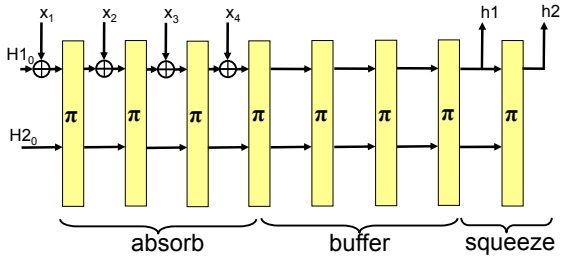
Quad [Berbain-Gilbert-Patarin'06]

- iteration of a randomly chosen multivariate quadratic system
- Key/IV setup – similar requirements to a block cipher
- k=v=80; s=160
- 2915 PIV cycles/byte

- Cryptanalysis [Yang-Chen-Bernstein-Chen'07]
  - Conclusion: need at least k=v=160; s=320

68

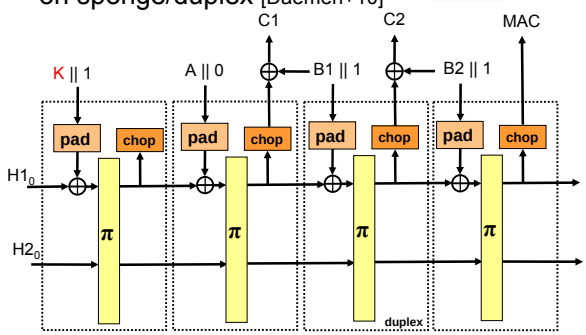
Permutation (π) based: sponge [Daemen+08]



Examples: Panama, RadioGatun, Grindahl, Keccak (no buffer)

69

Authenticated encryption based on sponge/duplex [Daemen+10]



A associated data; B plaintext; C ciphertext

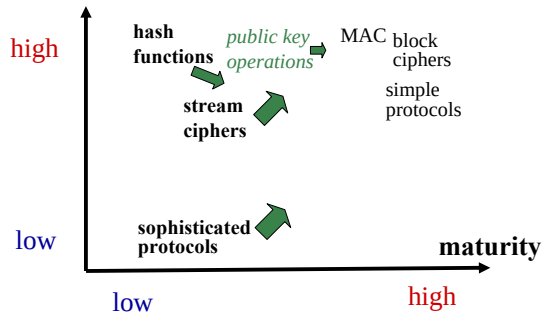
70

Authenticated encryption based on sponge/duplex [Daemen+10]

- Building block of SHA-3 candidate Keccak can be used for authenticated encryption with associated data
- More efficient than GCM/CCM
- Not patented
- Pseudo-random bit generator with reseeding

- More details: see paper of Daemen et al. at Aug 2010 SHA-3 conference

71



72

Conclusions

- stream ciphers are not dead
  - even if AES gets faster, they offer interesting trade-offs not reachable by AES
- open issues
  - authenticated encryption
  - stream ciphers based on permutations and state permutations
  - lightweight crypto
  - self-synchronizing stream ciphers
- many designs still very new and close to the edge
- standardization for the coming years

73