eSTREAM

Short Report on the End of the Second Phase

Contributors:

Steve Babbage (Vod)
Anne Canteaut (INRIA)
Henri Gilbert (FTRD)
Christof Paar (RUB)
Bart Preneel (KUL)
Matt Robshaw (FTRD)
Christophe de Canniére (KUL)
Carlos Cid (RHUL)
Thomas Johansson (LUND)
Matthew Parker (UiB)
Vincent Rijmen (IAIK)
Hongjun Wu (KUL)

26 March, 2007

This work has been supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

ECRYPT Network of Excellence in Cryptology

1 Introduction

The ECRYPT stream cipher project, eSTREAM, is a multi-year effort to identify promising new stream ciphers. During an initial study, which included input from industry, it was agreed that eSTREAM stream ciphers should be suited to at least one of two PROFILES:

- PROFILE 1. Stream ciphers for software applications with high throughput requirements.
- PROFILE 2. Stream ciphers for hardware applications with restricted resources such as limited storage, gate count, or power consumption.

Some experts emphasized the importance of including an authentication method and so two further profiles were also proposed:

- PROFILE 1A. Stream ciphers satisfying PROFILE 1 with an associated authentication method.
- PROFILE 2A. Stream ciphers satisfying PROFILE 2 with an associated authentication method.

Our final goal is a small portfolio of stream ciphers that might be of broad interest to the community. The ciphers in the eSTREAM portfolio are likely to mark a significant advance in the development of stream ciphers and to represent some of the most promising contemporary proposals. While the ECRYPT NoE is not a standardisation body, it is likely that some algorithms from eSTREAM may find their way into future standards.

Full information about eSTREAM can be found at

`http://www.ecrypt.eu.org/stream/.`

2 eSTREAM: Phase 1

The original call for proposals generated considerable interest with 34 proposals being submitted to the two different performance profiles. Those ciphers offering an authentication mechanism are indicated with a letter A.

Profile I		*Profile I and II*		*Profile II*	
ABC		F-FCSR		Achterbahn	
CryptMT/Fubuki		Hermes8		DECIM	
DICING		LEX		Edon-80	
DRAGON		MAG		Grain	
Frogbit	A	NLS	A	MICKEY	
HC-256		Phelix	A	MOSQUITO	
Mir-1		Polar Bear		SFINKS	A
Py		POMARANCH		Trivium	
SOSEMANUK		Rabbit		TSC-3	
		SSS	A	VEST	A
		TRBDK3 YAEA		WG	
		Yamb		Zk-Crypt	
		Salsa20			

After less than a year of analysis, more than half of the initial proposals had demonstrated weaknesses of some kind. Nevertheless, one of the strengths of the eSTREAM process was to allow designers to submit tweaks to their proposals so as to improve algorithm security and/or performance. This option to make tweaks was also extended to algorithms that had been broken.

The decision on which algorithms would move forward to a second phase was based on public discussions at SASC 2006, postings on the eSTREAM discussion phorum, and analysis by an internal committee. The net result was a second phase consisting of 28 algorithms, divided into the two profiles.

The algorithms Frogbit, MAG, Mir-1, SFINKS , SSS, TRBDK3 YAEA, and Yamb were archived and not considered in the second phase. These were ciphers for which no tweaks were proposed even though substantial weaknesses in security or performance had been identified, or for which updated code and documentation had not been received.

3 eSTREAM: Phase 2

The following algorithms were accepted to the second phase of eSTREAM. Those ciphers offering an authentication mechanism are indicated with a letter A. For tweaked ciphers we use the version-numbers provided by the authors or the mark (P2) to indicate the version considered in the second phase.

Profile I		*Profile II*	
ABC v3		Achterbahn-128/80	
CryptMT v3		DECIM v2	
DICING (P2)		Edon-80	
DRAGON		Grain v1	
HC-128 (-256)		F-FCSR-H (-16)	
LEX		Hermes8	
NLS v2	A	LEX	
Phelix	A	MICKEY v2	
Polar Bear v2		MICKEY-128 v2	
Py		MOUSTIQUE	
Rabbit		NLS v2	A
Salsa20		Phelix	A
SOSEMANUK		Polar Bear v2	
		POMARANCH v3	
		Rabbit	
		Salsa20	
		Trivium	
		TSC-4	
		VEST (P2)	A
		WG (P2)	
		Zk-Crypt (P2)	A

Since there were so many algorithms in the second phase (and time is limited in eSTREAM) we thought that we would try to concentrate cryptanalytical and implementation effort on a small group of promising ciphers. The software focus ciphers were

DRAGON HC-128 (-256) LEX Py Salsa20 SOSEMANUK

while the hardware focus ciphers were

Grain v1 MICKEY-128 v2 Phelix Trivium.

After another opportunity for the submission of optimised code along with many new results on cryptanalysis and implementation, the SASC 2007 workshop provided the basis for the start of the third phase of eSTREAM.

4 eSTREAM: Phase 3

The algorithms selected for the next phase are listed below. We no longer use the concept of a "focus cipher" since, in effect, all ciphers in the third phase are focus ciphers. NLS v2 advances as encryption-only.

Profile I	*Profile II*
CryptMT v3	DECIM v2 (-128)
DRAGON	Edon-80
HC-128 (-256)	F-FCSR-H (-16)
LEX	Grain v1 (-128)
NLS v2	MICKEY v2 (-128)
Rabbit	MOUSTIQUE
Salsa20	POMARANCH v3
SOSEMANUK	Trivium

4.1 Evaluation criteria

The assessment of algorithms for Phase 3 used the results presented at SASC 2007 as the starting point. During the assessment process we strived to be as objective as possible. Clearly, submissions with identified security issues such as (partial) key or state recovery attacks could not be advanced. Looking to the final stages of eSTREAM, we were particularly concerned that the presentation of some ciphers might be too opaque to welcome independent cryptanalysis and implementation. With little more than one year remaining in eSTREAM this became a very compelling factor. Our final decision, therefore, depended on a combination of the following issues:

1. Security.
2. Performance when compared to the AES and other submissions.
3. Justification and supporting analysis.
4. Simplicity and flexibility.
5. Completeness and clarity of submission.

Our decision at the end of Phase 2 has been completely independent of the IP status of any cipher.

4.2 Comments on Phase 2 Algorithms

Our decisions at the close of the second phase are detailed below.

4.2.1 Algorithms that were advanced from Profile I

Our focus in Profile I is on stream ciphers for software applications with high throughput requirements. We believe that the following algorithms show sufficient potential, in one way or another, to be advanced to the next phase.

CryptMT v3	DRAGON	HC-128 (HC-256)	NLS v2
LEX	Rabbit	Salsa20	SOSEMANUK

While our focus is on ciphers with 128-bit keys, we have retained companion versions that support 256-bit keys. Note that NLS v2 progresses as encryption-only. Poor performance of the authentication component means that we no longer consider this feature in eSTREAM.

4.2.2 Algorithms that were advanced from Profile II

Our focus in Profile II is on hardware applications with restricted resources. Our primary criteria at this stage of eSTREAM, after security, has been the space requirements and we believe that the following algorithms might permit fruitful implementation trade-offs that should be explored further in the next phase.

DECIM v2 (-128)	Edon-80	Grain v1 (-128)
F-FCSR-H (-16)	MICKEY v2	MICKEY-128 v2
MOUSTIQUE	POMARANCH v3	Trivium

While our focus is on ciphers with 80-bit keys, we have retained companion versions that support 128-bit keys.

4.2.3 Algorithms that were archived from Profile I

ABC v3

This is one of the fastest stream ciphers among the eSTREAM submissions. However, there are security problems. The original ABC was broken by Berbain and Gilbert with a divide-and-conquer attack [2]. The next version, ABC v2, had weak keys [8] which resulted in a heavily biased keystream. The third version, ABC v3, eliminated the previously identified weak keys but had a related weakness [11]. All three versions of ABC have been attacked and the design approach appears to be flawed.

DICING (P2)

While no attacks have been report against DICING (P2), it does not seem to offer any significant performance advantages over the AES or over other candidates submitted to eSTREAM. As a consequence we have decided not to advance the cipher to the third phase.

Phelix

Phelix is a stream cipher with built-in authentication function and encouraging performance. However, Wu and Preneel [10] have published an attack allowing efficient recovery of the secret key, *if the attacker can re-use the same nonce value more than once*. This paper has been the subject of much debate. The usage rules for Phelix explicitly forbid nonce re-use; and it is clear that, for any algorithm like this, security properties fail (specifically, MAC forgery is possible) if nonce re-use is permitted. Thus many observers have asserted that [10] does not count as an attack.

However, the Phelix designers do claim that key recovery should not be possible even if the nonce is re-used and [10] clearly invalidates this claim. Furthermore, we believe that the attack does constitute a genuine threat against real life systems using Phelix. It does seem plausible that an attacker would be able to mount an attack against such a system, re-using nonces, and that recovering the key would be a serious outcome. Attackers are not usually bound by usage rules. With some regret we have decided not to advance Phelix. However we believe the algorithm has good features and we encourage further research along these lines.

Polar Bear v2

Polar Bear v2 fixes an initial flaw in Polar Bear and in the second phase no attack against Polar Bear v2 was reported. Furthermore, some modifications to the algorithm during the tweak lead to an improved performance. However, generally speaking, the software performance of Polar Bear v2 doesn't compare too well to many other eSTREAM candidates. As a consequence we have decided not to advance the cipher to the next phase.

Py

Py and its variants demonstrate a promising approach that might offer exceptional performance. Unfortunately, however, there is sufficient analysis [4, 6, 9] to suggest that the submitted versions of the cipher demonstrate a weakness in the design. While this means that we cannot advance the cipher to the next phase of eSTREAM, we find the approach interesting and hope to see it explored in other contexts.

4.2.4 Algorithms that were archived from Profile II

Achterbahn-128/80

The phase two version of Achterbahn has variants with 80-bit and 128-bit keys. Key-recovery attacks against both versions have been presented by Hell and Johansson [3] and Naya-Plasencia [7]. These attacks essentially exploit the independence of internal registers and the existence of a good approximation of the combining function. The current best attacks achieve key-recovery against Achterbahn-80 with around 2^{61} operations, while a similar attack against Achterbahn-128 requires 2^{61} space and $2^{80.6}$ time.

Similar key-recovery attacks still apply when maximum frame length is limited to 2^{56} bits (resp. to 2^{52} bits) for the 128-bit version (resp. for the 80-bit version). Unfortunately the two versions of Achterbahn do not appear to provide enough security and the ciphers cannot be advanced to the third phase.

Hermes8

There are two versions of Hermes8 currently under consideration: Hermes8 (as submitted originally to eSTREAM), and the faster Hermes8F. Hermes8F is subject to a devastating cryptanalytic attack [1]. This attack does not seem to extend directly to Hermes8; however, the paper does identify serious flaws in the design principles of the Hermes8 family as a whole. As a consequence we feel unable to advance either version of Hermes8 to the next phase.

LEX

LEX, like the AES, will lend itself to hardware implementation in general. But in terms of the most compact implementations, it is not clear that LEX will offer any significant space advantages over the AES.

NLS v2

While it is true that many of the operations in NLS v2 can be performed without great complexity in hardware, the cipher is unlikely to be a realistic candidate for very resource-constrained hardware. NLS v2 may still lend itself to general hardware implementation.

Phelix

Independently of our comments for the software evaluation of Phelix, the form of Phelix suggests that it is unlikely to be among the ciphers most suited to our target environment of very resource-constrained hardware.

Polar Bear v2

No attacks have been reported against Polar Bear v2. However we believe that the S-box that is used in Polar Bear makes it ill-suited to our target applications that require compact implementations.

Rabbit

While Rabbit offers many trade-offs and may lend itself to hardware implementation in general, the available figures for Rabbit suggest that it is unlikely to provide substantial advantages over the AES in a compact hardware implementation.

Salsa20

While Salsa20 may lend itself to hardware implementation in general, the available results on Salsa20 suggest that it is unlikely to be a realistic candidate for our target environment of very resource-constrained hardware.

TSC-4

While no weaknesses have been reported in this version of the cipher, previous versions were broken in a range of attacks. Since there is little supporting security analysis of this cipher, the reliability of the underlying construction might be open to some question. With regards to performance, the available metrics do not suggest that TSC-4 would offer particularly compact hardware advantages over the AES or over other designs submitted to eSTREAM. Since this is the main focus of Profile II, we have decided not to advance the cipher.

VEST (P2)

A paper due to Joux and Reinhard [5] describes an attack against the submitted version of VEST. This practical attack allows the recovery of internal state and means that the cipher cannot be advanced to the next phase of eSTREAM.

WG (P2)

No attacks have been reported against WG (P2). However, since the linear complexity of the keystream is around 2^{45}, the cipher is fully compromised after only a slight relaxation of the restriction that no more than 2^{45} bits be generated from a single key/IV pair. At the same time, the information we have to hand on hardware implementation suggests that WG (P2) will be larger than we would like. Thus we have decided not to advance WG to the third phase.

Zk-Crypt (P2)

Zk-Crypt has poor documentation. This is a great obstacle to anyone trying to attempt cryptanalysis. In particular, within a limited timeframe, anyone looking over the set of eSTREAM submissions with a view to attempting cryptanalysis on one of them, will almost certainly pick an algorithm that can be understood more readily. We suspect that there has been little independent security analysis of Zk-Crypt and, with the documentation to hand, we would expect this to continue in the third phase. We feel that Zk-Crypt cannot be advanced to the next phase.

5 Looking Ahead to Phase 3

We expect that the final eSTREAM portfolio will be announced in May of 2008. As has now become traditional, there will be an ECRYPT workshop devoted to stream ciphers—SASC 2008—in February of 2008. During this final phase of eSTREAM, security will remain the top priority. We are also very keen to receive implementation results on the Profile II ciphers that have yet to receive independent analysis.

Acknowledgements

The most important contributors to eSTREAM have been the cipher designers. We are grateful for all the work that has gone into preparing a submission and to all those that have cryptanalysed, implemented, and commented on the candidates. It is the nature of eSTREAM that not every submission will be in the final portfolio. While many submitters will be disappointed that their algorithm has not been advanced to later stages of the project, we would like to acknowledge their contributions to collectively advancing the understanding of stream ciphers by a very significant margin.

References

[1] S. Babbage, C. Cid, N. Pramstaller and H. Raddum. Cryptanalysis of Hermes8F. www.ecrypt.eu.org/stream/papersdir/2007/009.pdf.

[2] C. Berbain and H. Gilbert. Cryptanalysis of ABC. www.ecrypt.eu.org/stream/papersdir/2005/048.pdf.

[3] M. Hell and T. Johansson. Cryptanalysis of Achterbahn-128/80. www.ecrypt.eu.org/stream/papersdir/2006/054.pdf.

[4] T. Isobe, T. Ohigashi, H. Kuwakado, and M. Morii. How to Break Py and Pypy by a Chosen IV-attack. www.ecrypt.eu.org/stream/papersdir/2007/035.pdf.

[5] A. Joux and J.-R. Reinhard. Overtaking VEST. www.ecrypt.eu.org/stream/papersdir/2007/021.pdf.

[6] S. Paul, B. Preneel, and G. Sekar. Distinguishing Attacks on the Stream Cipher Py. www.ecrypt.eu.org/stream/papersdir/2005/081.pdf.

[7] M. Naya-Plasencia. Cryptanalysis of Achterbahn-128/80. www.ecrypt.eu.org/stream/papersdir/2007/019.pdf.

[8] H. Wu and B. Preneel. Cryptanalysis of ABC v2. www.ecrypt.eu.org/stream/papersdir/2006/029.pdf.

[9] H. Wu and B. Preneel. Key Recovery Attack on Py and Pypy with Chosen IVs. www.ecrypt.eu.org/stream/papersdir/2006/052.pdf.

[10] H. Wu and B. Preneel. Differential-Linear Attacks against the Stream Cipher Phelix. www.ecrypt.eu.org/stream/papersdir/2006/056.pdf.

[11] H. Zhang, L. Li, and X. Wang. Fast Correlation Attack on Stream Cipher ABC v3. www.ecrypt.eu.org/stream/papersdir/2006/049.pdf.

Voting at SASC 2007

Voting forms were distributed at SASC 2007 to attendees. A simple ranking scheme can be applied[1] with larger positive scores reflecting a greater enthusiasm for the algorithm. We were not bound by the vote, though our decisions are roughly in line with attendee preference.

Algorithm (*Profile I*)	*Move to Next Round*	*Consider For Next Round*	*Not Very Suitable For Next Round*	*Archive*	*Weighted Score*
Salsa20	35	9	0	0	1.80
Sosemanuk	20	12	4	3	1.03
HC 128/256	22	5	6	3	1.08
LEX	20	11	6	4	0.90
Rabbit	18	13	7	3	0.88
Dragon	12	10	11	3	0.47
Phelix	13	12	9	8	0.31
CryptMT	7	18	5	8	0.29
NLS	0	13	10	12	−0.60
Polar Bear	0	8	12	10	−0.80
Dicing	0	7	13	12	−0.94
ABC	0	9	10	25	−1.16
Py	0	4	15	28	−1.43
Grain	39	7	1	2	1.63
Trivium	39	6	3	2	1.54
Salsa20	20	12	7	2	1.00
Mickey128	18	7	8	5	0.66
Rabbit	11	13	11	6	0.29
F-FCSR	5	16	10	3	0.29
LEX	9	14	8	8	0.21
Phelix	11	11	10	8	0.18
Mickey	11	9	7	10	0.11
Edon80	2	19	11	6	0.00
Pomaranch	3	12	12	10	−0.38
Decim	0	13	11	7	−0.39
Moustique	1	9	15	9	−0.65
Vest	7	4	12	18	−0.73
WG	0	9	11	13	−0.85
ZK-Crypt	2	6	9	17	−0.97
NLS	0	8	10	16	−1.00
Polar Bear	0	6	10	15	−1.10
Hermes8	0	6	14	18	−1.16
TSC4	0	6	7	19	−1.22
Achterbahn	1	2	8	35	−1.61

[1]Scores in the first column were multiplied by 2, in the second by 1, in the third by -1, and by -2 in the last. The weighted sum was divided by the total votes cast for that algorithm.