

# Cryptanalysis of Grain using Time / Memory / Data Tradeoffs

v1.0 / 2008-02-25

T.E. Bjørstad

The Selmer Center, Department of Informatics,  
University of Bergen, Pb. 7800, N-5020 Bergen, Norway.  
Email : tor.bjorstad@ii.uib.no

**Abstract.** Grain is one of the eight finalists in the hardware category of the eSTREAM stream cipher contest, and has shown itself to be one of the fastest and most compact stream ciphers in the competition. Time / memory / data tradeoffs are a class of generic attacks that modern stream ciphers should resist. We show that Grain has a low resistance to BSW sampling, leading to tradeoffs that in the active phase recover the internal state of Grain v1 using  $O(2^{71})$  time and memory, and  $O(2^{53.5})$  bits of known keystream. While the practical significance of these tradeoffs may be arguable due to the precomputation costs involved, their existence clearly violate stated design assumptions in the Grain specification, and parallels may be drawn to the similar cryptanalytic results on and the subsequent tweak of MICKEY v1.

## 1 Introduction

The eSTREAM [1] project is a multi-year effort to develop a portfolio of promising new stream ciphers. Sponsored by the ECRYPT Network of Excellence, the project began in 2004 with a call for proposals from industry and academia. These proposals were intended to satisfy either a software-oriented or hardware-oriented profile, and to provide some advantage over AES in at least one significant aspect. The original call for proposals generated 34 submissions in total. Grain [11] is a submission in the hardware category by Hell, Johansson and Meier. After attacks were found against the initial design, a tweaked version called Grain v1 was submitted by the authors. This cipher is one of the eight finalists in the hardware category of eSTREAM [1], and has attracted a lot of attention due to its high speed, low gate count and low power consumption [10]. There is also a 128-bit version of Grain called Grain-128 [13]. In the context of this article, “Grain” refers to Grain v1 as specified in the Phase 3 documentation of eSTREAM [12].

### 1.1 Time / Memory / Data Tradeoffs

Time / memory (TM) tradeoffs were first introduced by Hellman [14] in 1980 as a generic way of attacking block ciphers, but can be generalised to the general

problem of inverting one-way functions. In the case of function inversion (but not for block ciphers), Babbage and Golic [2,9] and later Biryukov, Shamir and Wagner [5,6] showed that the basic TM tradeoffs can be improved significantly by using several data points. This class of tradeoffs are thus known as time / memory / data (TMD) tradeoffs, and use the birthday paradox to ensure that a preimage will be found for at least one of the data points with high probability. For stream ciphers, the one-way function to be inverted is commonly taken to be the map sending an  $n$ -bit internal state of the cipher to the first  $n$  bits of keystream generated from that state. Several stream ciphers have been broken by TMD tradeoffs, most famously the GSM encryption scheme A5/1 [6].

Using a TMD tradeoff to invert a function can be split into two phases; preprocessing and an online / active phase. In the preprocessing step, which is performed *once*, the attacker builds large tables relating to the behaviour of the function in question. In the active phase, the attacker obtains a number of actual data points that she wants to invert, and tries to find a preimage of at least one value using the precomputed tables. A TMD tradeoff is thus characterised by five parameters; the size of the search space  $N$ , the time consumed by the precomputation step  $P$ , the amount of memory  $M$  used to store the precomputed tables, the time complexity of the active phase  $T$ , and the amount of data required  $D$ . In the analysis, logarithmic terms are usually ignored. In particular, the Babbage-Golic tradeoff is specified by the relations  $P = M$ ,  $N = TM$  and  $T = D$ , while the Biryukov-Shamir tradeoff is given by  $P = N/D$ ,  $N^2 = TM^2D^2$ ,  $D^2 \leq T$  [5]. The particular details of how these tradeoffs work are outside the scope of this article.

Modern TMD attacks have been well known to cipher designers for many years, and are taken into account by cipher designers. In 1995, Babbage stated that as a general design rule [2],

“if a secret key length of  $k$  bits is required, a state size of at least  $2k$  bits is desirable”

and this remains the rule of thumb used for new stream ciphers designs today. To provide insight into the motivation for this bound, notice that for a state of size  $N = 2^{2k}$ , the particular parameter choices  $T = M = D = 2^k$  and  $T = M = 2^k, D = 2^{k/2}$  are points on the Babbage-Golic and Biryukov-Shamir tradeoff curves, respectively.<sup>1</sup>

---

<sup>1</sup> It should, however, be noted that the given tradeoffs are much less feasible than regular brute-force attacks, even at this “break-even” point. In the case of the Babbage-Golic tradeoff, the requirement that  $T = D$  translates into a requirement of  $2^k$  bits of keystream, whereas most ciphers offering  $k$ -bit security specify the maximum amount of keystream generated by a single key to be significantly less. For the Biryukov-Shamir tradeoff, the complexity of the precomputation phase is a whopping  $N/D = O(2^{3k/2})$  steps.

## 1.2 Description of Grain

In this section we describe the parts of Grain which are relevant to the purposes of this article. For the full specification, we refer to [12]. Grain is bit-oriented stream cipher taking an 80-bit key and a 64-bit IV. The cipher consists of two 80-bit shift registers, one linear and one non-linear, whose states at time  $i$  is denoted by  $\{l_i, \dots, l_{i+79}\}$  and  $\{n_i, \dots, n_{i+79}\}$ .

During key generation, Grain outputs a single bit  $z_i$  at each clock cycle, which is computed from 12 of the internal state bits by the equation

$$z_i = n_{i+1} + n_{i+2} + n_{i+4} + n_{i+10} + n_{i+31} + n_{i+43} + n_{i+56} + h(l_{i+3}, l_{i+25}, l_{i+46}, l_{i+64}, n_{i+63}). \quad (1)$$

In the above,  $h(\cdot)$  is a non-linear boolean function of degree 3. Meanwhile, the new LFSR bit is computed using a primitive feedback polynomial  $f(x) = 1 + x^{18} + x^{29} + x^{42} + x^{57} + x^{67} + x^{80}$  which yields the update equation

$$l_{i+80} = l_{i+0} + l_{i+13} + l_{i+23} + l_{i+38} + l_{i+51} + l_{i+62}. \quad (2)$$

The NFSR is updated with a non-linear boolean function  $g(\cdot)$  of degree six whose output is masked with the low-order LFSR bit:

$$n_{i+80} = l_i + g(n_i, n_{i+9}, n_{i+14}, n_{i+15}, n_{i+21}, n_{i+28}, n_{i+33}, n_{i+37}, n_{i+45}, n_{i+52}, n_{i+60}, n_{i+62}, n_{i+63}). \quad (3)$$

A notable aspect of Grain is that every bit that enters the registers remains unused for at least 16 clock cycles, since the highest register indices occurring in the state update functions are  $n_{i+63}$  and  $l_{i+64}$  respectively. This enables implementors to compute up to 16 clocks of the cipher in parallel, greatly speeding up the cipher with only a moderate increase in gate count. Finally, we note that the state update function of Grain is invertible both during keystream generation and key initialisation. This implies that if we recover the state of the cipher at some time  $t$ , we can clock it backwards to recover the key used.

## 1.3 Related work

The best current attack on Grain v1 is an observation by De Canniere, Küçük and Preneel [7] on sliding properties in key-setup of Grain, which on average allows a brute-force attacker to test keys twice as fast as usual, yielding a worst-case brute force attack complexity of  $2^{79}$  trials. The authors also analyse the key initialisation procedure of Grain.

In the eSTREAM contest itself, there has been relatively little focus on TMD tradeoffs, presumably because the cipher designers have been aware of this class of attacks and have avoided them by designing ciphers whose internal states are

not only twice the length of the secret key, but usually much larger. However, the TMD tradeoffs proposed by Hong and Kim against MICKEY v1 [15] are very noteworthy, as they are very similar to those found for Grain in this article. The parallels between Grain and MICKEY v1 in this respect will be explored further in the next section.

Research has also been done on TMD tradeoffs for a different inversion problem, namely that of inverting the one-way function mapping the initial cipher input of key + IV to keystream. Recently, Dunkelman and Keller [8] proposed a new TMD tradeoff based on chosen IVs which appears to lead to interesting tradeoffs, particularly for schemes where the IV is chosen deterministically.

## 2 Time / Memory / Data Cryptanalysis of Grain

Since the 160-bit internal state of Grain is twice the length of the key, the known time / memory / data tradeoff curves can not be applied directly to the function mapping the 160-bit internal state to the subsequent 160 bits of keystream. In this section, we show how better tradeoffs may be obtained.

### 2.1 Sampling Resistance of Grain

In [5,6] the authors introduce the concept of BSW-sampling, which can be used to obtain wider choices of tradeoff parameters for the Biryukov-Shamir tradeoff curve. The main idea of BSW-sampling is to find an efficient way to generate and enumerate “special” cipher states, from which the first subsequent output bits of the cipher are a fixed string (such as a run of consecutive 1-bits). If this can be done for a run of  $l$  bits, the *sampling resistance* of the cipher is defined to be  $R = 2^{-l}$ . This is usually possible (and is not a cause for concern) for small values of  $l$ , but leads to improved tradeoff attacks if  $l$  is moderately large. We will proceed to show that Grain is vulnerable to BSW-sampling.

**Lemma 1.** *Given the value of 133 particular state bits of Grain and the first 18 keystream bits produced from that state, another 18 internal state bits may be deduced efficiently.*

*Proof.* Recall the form of output function of Grain from Eq. 1, in which the non-linear function  $h(\cdot)$  is computed and the result is masked with 7 bits taken from the NFSR state. Our strategy is to exploit the distance between the masking bits  $n_{i+10}$  and  $n_{i+31}$ , together with the fact that the non-linear feedback of Grain does not affect the output function until the cipher has been clocked 17 times.

At a particular point in time, it is clear that the value of  $n_{i+10}$  can be computed directly from the values of  $z_i$  and the 11 other state variables occurring in the output equation. Doing this 16 times for  $i = 0 \dots 15$  yields

$$n_{10} = z_0 + n_1 + n_2 + n_4 + n_{31} + n_{43} + n_{56} + h(l_3, l_{25}, l_{46}, l_{64}, n_{63}), \quad (4)$$

$\vdots$

$$n_{25} = z_{15} + n_{16} + n_{17} + n_{19} + n_{46} + n_{58} + n_{71} + h(l_{18}, l_{40}, l_{61}, l_{79}, n_{78}), \quad (5)$$

where at each step the computed NFSR bit has not occurred in any of the previous equations. At this point we have fixed the values of 57 NFSR bits and 64 LFSR bits, and deduced 16 of the NFSR bits.

In step 16, the value of  $l_{80}$  is used as input to  $h(\cdot)$ , and we have to fix the values of some additional LFSR bits. Since we are not trying to deduce anything about the LFSR this is not a problem, and we subsequently compute  $n_{26}$  without any trouble.

On the 17th step, the value of  $n_{80}$  is also used as input to  $h(\cdot)$ , which means that we also have to take the non-linear feedback into account. In particular, this forces us to fix the value of  $n_{28}$  (which is part of the input to the NFSR update function used to compute  $n_{80}$  at time  $i = 0$ ). Apart from this complication, we are able to deduce  $n_{27}$ .

However, in the 18th step the value of  $n_{28}$  was just assigned in the previous step, as are the values of the 6 other linear masking bits. Hence we are not able to keep deducing more bits with this strategy. To sum up, we have recovered the 18 state bits  $n_{10} \dots n_{27}$  using 59 bits of the NFSR state and 74 bits of the LFSR state. The remaining 9 state bits did not occur in any of the equations.  $\square$

It might be possible to use a different strategy in order to deduce even more bits. An obvious approach which has not yet been explored is to try to fix only the state values needed to linearise  $h(\cdot)$  in each step, and solve the resulting linear system of equations in the remaining free variables after a number of steps. It is not known<sup>2</sup> whether this approach will enable us to obtain even more state bits.

**Corollary 1.** *The sampling resistance  $R$  of Grain is at most  $2^{-18}$ .*

Given that the sampling resistance of Grain is only  $2^{-18}$ , we can define<sup>3</sup> a family of one-way functions  $\pi_S : \{0, 1\}^{142} \rightarrow \{0, 1\}^{142}$  as follows:

1. Fix a specific function by choosing an 18-bit string  $S$ , e.g.  $1^{18}$ .
2. Given an 142-bit input value  $x$ , expand it to 160 bits by interpreting it as a partial state value for Grain and computing the remaining 18 bits by treating  $S$  as the first 18 bits of keystream.
3. Clock Grain 160 steps, generating a 160-bit output string  $S|y$ .
4. Output  $y$ .

It is clear that computing inverses for a particular function  $\pi_S$  is equivalent to the problem of inverting the function mapping 160-bit Grain states to 160-bit keystream segments, restricted to a “special” subset of the states. Furthermore,

<sup>2</sup> From (unsuccessful) attempts at algebraic cryptanalysis of Grain, the experience of the author is that by fixing around 120 – 130 bits of the state at random and using a small amount of known keystream, a linear system of equations is usually obtained in the remaining variables. By solving these equations the missing state bits can be recovered. However, it must be emphasised that the approach needs more keystream than just the initial  $l$  bits, and that one has not demonstrated that a linear system always can be obtained deterministically by this method.

<sup>3</sup> This construction was first given in [5].

we are easily able to construct these states and compute the function  $\pi_S$  (more or less) as efficiently as for the full state function. Finally, given  $D$  bits of actual keystream, the expected number of special states encountered is  $DR$ . Hence, we will consider the cost of inverting  $\pi_S$  rather than the full cipher.

## 2.2 Tradeoff Parameters

The Biryukov-Shamir tradeoff curve for inverting the function  $\pi_S$  can be written as  $(RN)^2 = TM^2(RD)^2$ , since the domain of the one-way function is reduced by a factor  $R$ , but the amount of keystream needed to obtain  $D$  data points is increased correspondingly. Hence the real gain from BSW-sampling is that we get a wider choice of parameters on the original tradeoff curve,<sup>4</sup> by relaxing the condition  $D^2 \leq T$  to  $(RD)^2 \leq T$ . Given that we can choose  $R = 2^{-18}$  with the sampling scheme proposed, we can use this to increase the total amount of keystream  $D$  beyond  $2^{40}$ , and reduce  $T$  and  $M$  correspondingly. So under the restriction that we want the magnitude of  $T$  and  $M$  to be at most  $2^{78}$ , some possible parameter choices are given in Fig. 1.

Time complexity ( $T$ )	Memory complexity ( $M$ )	Keystream ( $D$ )	Preprocessing time ( $P$ )
$2^{78}$	$2^{64}$	$2^{57}$	$2^{103}$
$2^{72}$	$2^{70}$	$2^{54}$	$2^{106}$
$2^{71}$	$2^{71}$	$2^{53.5}$	$2^{106.5}$
$2^{70}$	$2^{72}$	$2^{53}$	$2^{107}$
$2^{64}$	$2^{78}$	$2^{50}$	$2^{110}$

**Fig. 1.** Possible TMD-tradeoffs for Grain using BSW sampling.

For instance, we see that we can find an attack whose online complexity is bounded by  $O(2^{71})$  time and memory, using  $2^{53.5}$  bits of keystream. This appears to be, if not entirely *practical* per se, a significant improvement on the  $O(2^{79})$  complexity of brute-force keysearch.

## 2.3 Implications

To be fair, there is little consensus among researchers whether the given tradeoffs in itself constitute an “attack” against Grain, because the precomputation step is more expensive than brute force. Worse yet, even if the sampling resistance of Grain can be decreased even further, the Biryukov-Shamir tradeoff will *never* yield a precomputation cost less than  $2^{80}$ . If a very resourceful adversary wishes to recover a very large number of different keys, and is able to obtain the required

<sup>4</sup> The Babbage-Golic tradeoff is not helped by BSW-sampling, since the total amount of keystream needed remains constant. Even if the tradeoff is changed from  $N = TM$  to  $RN = TM$ , the corresponding amount of keystream needed to produce  $T$  data points grows to  $T/R$ .

amount of keystream for each key, the amortized cost of this attack does not drop below the cost of brute-force until at least  $2^{27}$  keys have been cracked. This leads to, admittedly, rather brave assumptions on the capabilities of an attacker for whom using this approach will pay off.

While some researchers will argue that the result shows that Grain can be broken faster than brute force (at least, as we see, under certain extreme circumstances) and that the cost of the precomputation step can be ignored in analysis, others hold that an “attack” is not relevant as long as the cost of a single (or a few) state recovery attacks remains much greater than brute force. This article does not in any way aim to settle this debate.

However, it is necessary to point out that the tradeoffs given in this article are rather hasty “first attempts” on part of the author<sup>5</sup>, and that some degree of improvement along this line of attack may be achievable. As importantly, we hold that our analysis is very relevant with regards to basic assumptions made in the Grain specification [12] about the cipher’s resistance to TMD tradeoff attacks. Specifically, quoting Sect. 6.3:

“The cost of time/memory/data tradeoff attacks on stream ciphers is  $O(2^{n/2})$ , where  $n$  is the number of inner states of the stream cipher [...]”

“[...] the resulting sampling resistance [of Grain] is large, and thus time / memory / data tradeoff attacks are expected to have complexity not lower than  $O(2^{80})$ .”

As we have seen, the sampling resistance of Grain is at most  $2^{-18}$ , which hardly can be said to be particularly large. Furthermore, when assessing the breakeven point for the cost of TMD tradeoff attacks on stream ciphers in the first part of the quote, the authors themselves seem to be neglecting the precomputation cost of TMD tradeoffs. Hence it would appear that Grain v1 has been designed with a security level of 80 bits in mind, also with respect to the *online* phase of a TMD tradeoff attack. Considering only the online phase of our tradeoff, our obtained attack complexity of  $O(2^{71})$  time and memory is a clear improvement on this bound.

The comparison between the sampling resistance of Grain and that of MICKEY v1 [3] also appears to be relevant. In [15], Hong and Kim show that the sampling resistance of MICKEY v1 is at most  $2^{-27}$ , and that this leads to TMD tradeoffs against that cipher with online complexity  $O(2^{67})$  and precomputation cost  $O(2^{100})$ , whereas the complexity of brute force keysearch is  $O(2^{80})$ . The authors write,

“Owing to the pre-computation complexity larger than exhaustive search of key, some would not view this technically as a *break* of MICKEY. But still, it does show that we cannot treat MICKEY as providing absolutely full 80-bit security.”

---

<sup>5</sup> This has been a necessary though unfortunate consequence of the perceived need to make a public draft available *before* the end of eSTREAM.

The response of MICKEY authors Babbage and Dodd [4] is interesting:

“Some authors seem to ignore precomputation time completely, and consider only online complexity to matter; others would say that an attack requiring overall complexity greater than exhaustive search is of no practical significance. Although we incline more towards the second view, we recognise that some will deem the cipher less than fully secure if such attacks exist.”

As a consequence of the TMD tradeoff attack discovered by Hong and Kim (as well as concerns about entropy loss in the state update function), a tweaked version of MICKEY v1 called MICKEY 2.0 was proposed [4]. In the tweaked cipher, the size of the state was increased from 160 to 200 bits, which is sufficient to eliminate the possibility of *any* state recovery attack based on BSW-sampling and the Biryukov-Shamir tradeoff curve.

If one wishes to make Grain secure against the tradeoffs found in this article, what may be done? Looking more closely at our results, the root causes of our obtained tradeoff appears to be threefold. The large spacing between the taps of masking bits  $n_{i+10}$  and  $n_{i+31}$  together with the long delay before any non-linear feedback in the NFSR starts affecting the output keystream (which makes diffusion slow but speeds up the keystream generation) allows us to perform BSW-sampling, while the minimalistic state size of the cipher enables us to use the lack of sampling resistance to obtain good tradeoffs. Tweaking any of these factors will defeat the attack as it stands.

The immediate opinion of the author is that simply modifying the position of masking bit taps might have unintended consequences or be possible to work around, while decreasing the feedback delay prevents the fast parallel implementation which is a main selling point of Grain. Either way, tweaking these factors does not eliminate the possibility of using BSW-sampling in TMD tradeoffs, only the particularly simple sampling scheme used here. Meanwhile, the solution used to tweak MICKEY 2.0 appears more prudent, as increasing the cipher state size to  $(2^k)^{2.5}$  bits forces at least one of  $T$ ,  $M$  or  $D$  to be greater than  $2^k$  if one wishes to use the Biryukov-Shamir tradeoff  $N^2 = TM^2D^2$  [4]. This can probably be done in a straightforward manner by extending the length of the shift registers, at a moderate increase in the overall hardware footprint of the cipher.

Finally, with respect to Grain-128, it can easily be verified that that cipher as specified has a sampling resistance of at most  $2^{-22}$ , by considering the spacing between the output masking taps at  $n_{i+15}$  and  $n_{i+36}$ . A TMD tradeoff similar to the one shown here for Grain v1 should thus be attainable, with an attack complexity in the active phase of  $O(2^{117})$  using  $O(2^{80.5})$  bits of keystream, and a precomputation cost of  $O(2^{175.5})$  steps. In this case it is the spacing of the masking bit taps rather than the non-linear feedback that prevents us from doing any further sampling.

### 3 End notes

We have shown that there exists a time/memory/data tradeoff to recover the state of Grain v1 with online time and memory complexity  $O(2^{71})$  after a pre-computation of  $O(2^{106.5})$  steps and using  $O(2^{53.5})$  bits of known keystream. Other parameter choices are also available. While the relevance of these attacks can be disputed owing to the huge cost of the precomputation, they still show that, at least under certain extreme assumptions, Grain keys can be recovered faster than by brute force search of the key space. This violates assumptions on the resistance to Grain to TMD tradeoffs stated in the official cipher specification, and indicate that Grain's minimalist internal state may in fact be too aggressively specified for some people's taste.

#### 3.1 Acknowledgements

The author would like to thank Orr Dunkelman, Håvard Raddum and Matthew Parker for early feedback and discussion.

### References

1. eSTREAM, ECRYPT stream cipher project. <http://www.ecrypt.eu.org/stream/>.
2. S. Babbage. A space/time tradeoff in exhaustive search attacks on stream ciphers. In *European Convention on Security and Detection*, volume No. 408, 1995.
3. S. Babbage and M. Dodd. The stream cipher MICKEY (version 1). eSTREAM, ECRYPT Stream Cipher Project, 2005.
4. S. Babbage and M. Dodd. The stream cipher MICKEY 2.0. eSTREAM, ECRYPT Stream Cipher Project, 2006. <http://www.ecrypt.eu.org/stream/mickeyp3.html>.
5. A. Biryukov and A. Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In *Proceedings of ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 1 – 13. Springer–Verlag.
6. A. Biryukov, A. Shamir, and D. Wagner. Real time cryptanalysis of A5/1 on a PC. In *Proceedings of PKC 2001*, volume 1978 of *Lecture Notes in Computer Science*, pages 37–44. Springer–Verlag.
7. C. De Cannière, Ö. Küçük, and B. Preneel. Analysis of grain's initialization algorithm. Presented at SASC 2008. <http://www.ecrypt.eu.org/stv1/sasc2008/>.
8. O. Dunkelman and N. Keller. Treatment of the initial value in time-memory-data tradeoff attacks on stream ciphers. Presented at SASC 2008. <http://www.ecrypt.eu.org/stv1/sasc2008/>.
9. J. Golic. Cryptanalysis of alleged A5 stream cipher. In *Proceedings of EUROCRYPT 1997*, volume 1233 of *Lecture Notes in Computer Science*, pages 239–255. Springer–Verlag.
10. T. Good and M. Benaïssa. Hardware performance of eSTREAM phase-III stream cipher candidates. Presented at SASC 2008. <http://www.ecrypt.eu.org/stv1/sasc2008/>.

11. M. Hell, T. Johansson, and W. Meier. Grain – a stream cipher for constrained environments. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/010, 2005.
12. M. Hell, T. Johansson, and W. Meier. Grain – a stream cipher for constrained environments. eSTREAM, ECRYPT Stream Cipher Project, 2006. <http://www.ecrypt.eu.org/stream/grainp3.html>.
13. M. Hell, T. Johansson, and W. Meier. A stream cipher proposal: Grain-128. eSTREAM, ECRYPT Stream Cipher Project, 2006. <http://www.ecrypt.eu.org/stream/grainp3.html>.
14. M. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26:401–406, 1980. <http://www-ee.stanford.edu/~hellman/publications/36.pdf>.
15. J. Hong and W.-H. Kim. Tmd-tradeoff and state entropy loss considerations of streamcipher MICKEY. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/055, 2005.