A fundamental evaluation of 80 bit keys employed by hardware oriented stream ciphers

Iain Devlin[1] and Alan Purvis[1]

Centre for Electronic Systems, Durham University,
Durham, DH1 3LH, UK
{iain.devlin, alan.purvis}@durham.ac.uk

Abstract. In this paper the security afforded by the 80bit keys of hardware focused stream ciphers is analysed from the perspective of brute force attack susceptibility. A FPGA based architecture capable of performing a successful key search is detailed and simulated. If fully implemented with current technology this system would, given a small data element, be capable of regularly recovering an 80bit key within a year for a Structured ASIC chip cost of €6 million. The implications of such capability on the use of eStream Profile II stream ciphers for commercial applications are outlined.

1 Introduction

When in 2004 a call was made for stream cipher primitives [1] as part of the competitive ECRYPT project eStream, two differing key length requirements were made for proposals aimed at software *(eStream Profile I)* and those aimed at hardware environments *(eStream Profile II)*. Profile I ciphers required that conventional 128bit keys be employed while hardware Profile II ciphers were given the lesser requirement of 80bit keys: a length short enough to be deemed insecure by ECRYPT [2] for medium-term protection of data when faced with a government funded adversary. Algorithmic security is a significant issue in the field of stream cipher design but the low key length raises a more fundamental question: if the algorithm is ideal what is the economic value of the data protection provided?

The key length security trade-off is a reasonable one if speed and efficiency advantages are convincing and the cost an adversary faces in retrieving protected data outweighs its value to them. This value is determined by a number of factors in the commercial world but will typically have a time dependancy: in some situations such as financial transactions data will have a high initial value but then decays rapidly within fractions of a second; others, such as proprietary multimedia transmission, will have a lower value to an attacker but a relatively gentle value decay curve that may extend to that of weeks or even months. Value is also determined by the availability of other methods for the attacker to receive the data and for example Eve who wants some data from Alice Corp. would probably find it much cheaper to bribe Andy who works there than analyse the

algorithms used and construct a €100 million key search machine. Nonetheless there is a tipping point where advantages to an adversary of a cipher-breaking machine will exceed other possibilities. This paper chooses to focus on such machines so as to help allow a well-informed decision to be made.

1.1 Motivation

Parallel brute force attacks on key space have many advantages when all practicalities are considered in that their methodology is simple leading to efficient and low cost implementations [3]. These attacks unlike algorithmic attacks are generic in nature and relevant to all ciphers, however this does not necessarily imply universal equal efficiency of operation.

Well designed stream ciphers should always outperform ciphers such as AES in the hardware environment since modern block ciphers must process a full block of data alongside the key while the operation of a synchronous stream cipher is almost data independent. Furthermore, block cipher implementations often need to trade speed for low area implementation where as a stream cipher can usually provide both properties without need for trade-offs. Consequently brute force attacks on a stream cipher may potentially be many 'bits' easier than raw differences between block and stream cipher key length would suggest.

However, although much research has gone into hardware brute force key search machines for block ciphers [4][5] the area of hardware key search machines for stream ciphers has received little attention. This paper attempts to address the increased relevance of brute force to efficient hardware stream ciphers and provides detail on the practical issues that must be considered when building a brute force key search system.

1.2 Use of Grain

One hardware focused entry to the eStream project is that of Grain [6] which in its basic form produces, after a 160 cycle set-up, 1bit of keystream every clock using a 160bit register, two feedback functions and an output function. Grain is further designed to be accelerated through the use of additional feedback loops which leads to a version that requires 10 cycles for initialisation and has sixteen times the throughput. This puts Grain among the highest performing and lowest resource entries to eStream [7] and with the addition of a parallel loading key interface Grain can load, initialise and produce two bytes of keystream in just 12 clock cycles. It is this high efficiency in key agility and resource usage that ironically identify the cipher as the most susceptible to potential brute force key search and it is consequently used as the keystream generator in the designs described in this paper to establish an upper limit on the potential of 80bit key search machines. The recent algorithmic insecurity [8] should not affect the validity of these results especially as the subsequent modification of Grain [9] secures the cipher without causing significant penalties for speed or resource usage.

1.3 Terminology

The term "plainstream" is used throughout this paper and is defined as a key-stream derived from knowledge of a plaintext-ciphertext pair.

2 Implementing Stream Cipher Key Search

2.1 Introduction

When the aim is brute force key search not only must cipher throughput be considered but the time required for set-up must also be allowed for. Indeed for stream ciphers this is the factor that dominates when doing a key search: even if the stream cipher produces a single bit of keystream per clock cycle after just one iteration half of searches can be discontinued (e.g. the keystream bit is a 1 but the search target starts with 0) and after gathering a byte's worth of material %99.6 of searches may be stopped. Set-up time is still required in block ciphers for key expansion but this takes up proportionately less of the key search effort simply on account of the fact, that due to the full 128 bits of output being generated with every search, their resource usage is fundamentally greater .

In light of this, possibly the most important improvement available for stream cipher key search is due to the independent keystream generation which allows simultaneous checking of multiple plainstreams at effectively zero time cost. This attack variation to brute force can prove highly effective at reducing the computation requirements of such a key search: if eight plainstreams are available the problem of finding a key-plainstream match suddenly reduces from $\frac{1}{2^{80}}$ to $\frac{1}{2^{77}}$ as there are now 2^3 as many outcomes considered positive and this translates to over 5×10^{23} fewer searches. It should be noted that the attack variation does require a consistent initialisation vector (IV) between plainstreams for the advantage to be realised.

The rest of this paper assumes that a small number of plainstreams relating to the some known IV are available to the attacker. This is a reasonable assumption in many situations as often a known file format header may make up the first few bytes of a transmitted message while if a counter has been used for IV generation then it is likely that only a small subset of IVs will be being utilised. Moreover, as IVs are transmitted in the clear these values must be assumed easily obtainable.

2.2 ASIC vs. FPGA

Before design of a system could begin a considered choice of platform had to be made between ASICs (application-specific integrated circuits) and FPGAs (field programmable gate arrays).

The ASIC allows system design in rawest form with control of individual transistors possible. It provides the maximum performance per unit of substrate area and offers the potential of a very efficient and high-speed key search engine. However, ASIC design is currently a complex, high cost process with shrinking

feature sizes constantly introducing new challenges to digital design due to changing physical device characteristics. These problems are only likely to escalate.

FPGAs offer an alternative approach and add a extra layer of abstraction in the design process, which allows the designer to concentrate on system design and not physical layout. Their ever growing popularity has led to FPGA progress being incredibly rapid over the past two decades (see Sect. 3.2) with falling prices and increased logic element counts. This is a trend which looks set to continue as the high overheads of nanometer processes encourage migration away from ASIC. The FPGA has no overhead costs associated with purchases and offers a flexible, inexpensive environment for digital designs. The programmable nature of the devices has a further effect especially relevant to attack architectures in that this allows an adversary to buy blank devices: there is no need to explain to a large multinational why a hundred thousand chips with ciphers on them are needed.

The rapid turn-around, low design costs, covert purchasing and low unit cost are very attractive features to a commercially based adversary and combined with the reasonable circuit speeds of modern FPGAs a decision was made that the design work described in the rest of this paper should focus on FPGA based systems.

2.3   System Design

Possibly the simplest brute force key search system would be one that takes a single stream cipher and varying key to produce unique keystreams for comparison with a plainstream. Assuming the cipher uses an 80bit key, then checking 80bits of keystream material with the plainstream would on average produce one positive match over the entire key space and so when eventually a match is detected it is likely that key in question was also used to produce the original plainstream.

As it has been assumed that multiple plainstreams with matching IV would be available, it is reasonable to extend the system so that it stores the keystream and checks each plainstream in turn while the cipher is reinitialising. This leads to the design shown in Figure 1. Also, if the number of plainstreams is great enough the memory comparison may take much longer than key loading and initialisation making the use of additional memory comparison units desirable. Such units have then to be identified and as in Figure 1 it is convenient for this identifier to form part of the key.

Although simple this architecture presents number of disadvantages to high speed, resource efficient key search with the majority of these founded on the need for 80bit comparison of plainstream and keystream. Reliable comparison is desirable to minimise the need for external computation but by introducing a two stage process efficiency can be greatly improved: the first stage comparing a small amount of keystream material eliminating the majority of keys and the second stage performing a full comparison on those that are left. For large systems the additional resource required for a back-end element is far outweighed
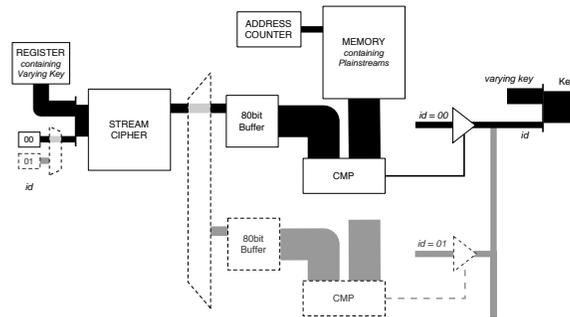
Fig. 1. System diagram of a simple key search system.

by the reduced memory requirements, logic and routing complexity seen at the front-end as long as the key elimination rate is high.

The 16bit comparisons shown in the design outlined in Figure 2(a) are sufficient to eliminate 99.9985% of keys and combined with the FIFO buffer allow a small key checking unit in the back-end to function in unison with a front-end that is orders of magnitude higher in performance. When the number of plainstreams available is large this serial search system becomes highly efficient as each sequential memory search can continue without performance penalty while other search units are being loaded with new keystreams.

The other theoretical key search system design considered (Fig. 2(b)) is much less memory based and involves a stream cipher producing a small amount of keystream which is immediately checked against multiple plainstreams. When the number of plainstream comparisons is small this leads to a very low resource high performance key search system without the control complexities required for memory based approaches.

The memory based architecture (Fig. 2(a)) described can utilise a much higher data element than the parallel search system (Fig. 2(b)) and if an application generates a large number of useful plainstreams for a potential attacker then this serial search design would likely have a lower resource overhead and so be the optimal low cost option to key search. Nonetheless, in designing a key search machine it is not only desirable to minimise the brute force computation required and the addition of a large memory element precludes a system from use in many otherwise applicable situations. Parallel search has its optimal data requirement low enough that a machine based on this system would be applicable to a wide range of applications and thus is a good compromise.

2.4 A working key search module

Basing the design on the theoretical parallel search system a working key search module was developed as shown in Figure 3. The architecture chosen is based upon a 16bit data path for the front-end which helps minimise the necessary
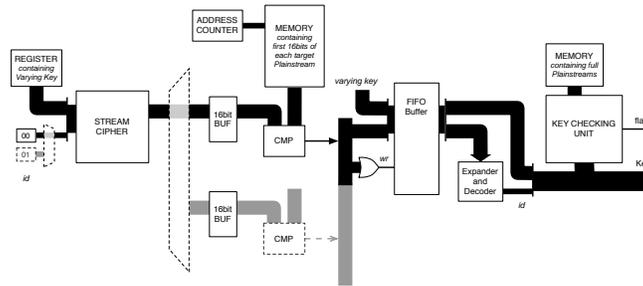
(a) Serial search - a memory based search system.
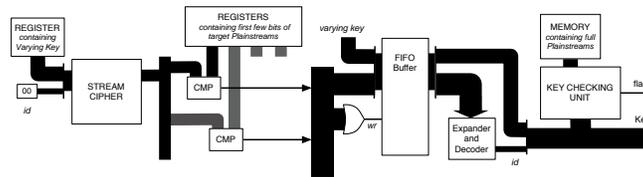
(b) Parallel search - data modified brute force.

Fig. 2. System diagram of two approaches to key search.

depth of the FIFO buffer and allows the Grain stream cipher to run with a minimum 10 clock cycle initialisation.

Simultaneous key loading is used to simplify control of the system but this consequently requires the buffer to accept a high number of front-end search outputs in parallel, which in turn necessitates a second smaller but deeper buffer to store outputs once expansion and decoding has taken place. The system diagram also shows outputs from comparison units being combined in pairs before being sent to the primary buffer. Although this requires memory and comparison unit duplication at the back-end of the system the speed and resource advantages of reduced buffer width more than compensate, particularly when the number of search units (n) and the number of plainstream registers (p) are large.

The key used in loading is split in three parts with first being a regularly changing variable provided by a system wide counter, the second being a short identifier that is unique to each cipher unit and the third being an externally controlled fixed register variable that is unique to the chip. By carefully selection of the register value loaded on completion of each counter cycle[1] multiple chips can accomplish a search of the 80bit key space. A small control unit (not shown in Figure 3) that controls loading of this register is also responsible for providing the desired degree of system flexibility in IV and plainstream memory values.

---

[1] Use of a LFSR based counter in the final system necessitates that one chip should be programmed differently, with the counter being held at zero and the fixed key being driven by a counter to mitigate the former's inability to reach an all zero state.
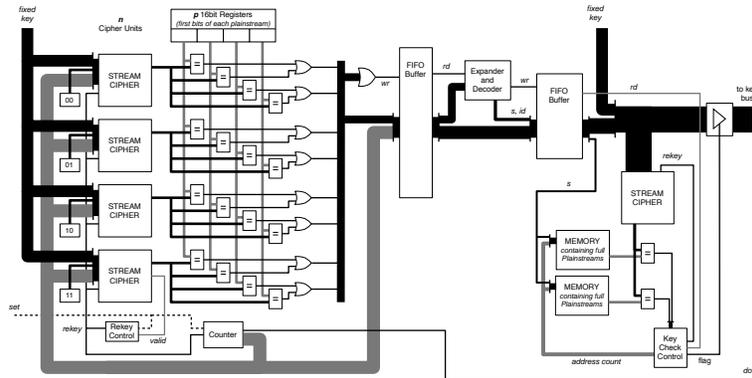
Fig. 3. System diagram of the full key search system

At the system output a relatively simple set-up is used with the key being tristated onto an 80bit bus when a match is found: the probability of two chips finding a match simultaneously is too low to be of concern. Further as a match may just be pure chance or more than one plainstream-key pair may be desired the final system caters for an acknowledgement signal to resume the paused chip. A pseudocode description of the full search system's operation is given in Appendix A.

2.5 System Testing

Initial simulation testing of a single chip core was carried out for the Altera Cyclone EP1C20F400C6 using Quartus II 5.0SP2 with results implying that the optimal brute force configuration on this chip is 16 search units each with a plainstream search space of 16. Such a configuration runs at 100MHz and uses 12018 logic elements[2], while the use of 4 back-end comparison units restrict memory requirements to just 19kbits. Resource usage is further detailed in Appendix B. Testing also indicated that although the use of a data element is relatively free for 8 plainstreams, beyond 16 its usefulness diminishes as the increased logic and routing complexity negate potential resource benefits.

3 Economics of 80bit Stream Cipher Brute Force

3.1 Introduction

If a well informed decision is to be made on the use of 80bit keys then performance figures must be translated into economic terms so a comparison to data value can be made. The potential of key search systems needs not only to be estimated for the current technological situation but also for those that may occur

---

[2] Excludes resources used to tristate the key output.

in the future. To obtain this information for systems such as those outlined it is first necessary to consider the past and future progress of FPGA technology.

3.2 FPGA History and Future Roadmap

After the conception of the FPGA in 1984 early chips were relatively small and novel devices: the 1985 Xilinx XC2064, designed for a 2μm process, made available just 64 logic blocks for logic configuration [10]. 21 years later FPGAs are commonplace devices that are found throughout electronic systems with modern high performance chips such as the Xilinx Virtex-4 XC4VLX200 designed on a 90nm process and containing advanced logic blocks that number in the hundreds of thousands. Looking at this substantial progress it is obvious that the field of FPGA development is one which is impressively fast moving. Furthermore, the performance gains to the designer are unlikely to slow in the near future as the simple structure of FPGA's repeated cells makes it likely the full benefit of future nanometer process nodes will be realised.

The presented key search designs make little use of the additional functionality offered by modern devices, therefore the most relevant measure of FPGA development is that of logic block count, speed and price. Estimating the long-term trend of these measures is complicated by the fact that logic block configurations are generally not consistent between manufacturers or even between device families. Fortunately Altera's recent low cost FPGA families have maintained a consistent logic block structure over a 6 year period of FPGA development and as a consequence, study of the block-count/price relationship focused attention on the "ACEX" (0.18μm), "Cyclone" (0.13μm) and "Cyclone II" (90nm) device families with their 4-input lookup table, flip-flop cell architecture. Choosing 250k device volumes and the chips in each family with lowest cost per logic element, Table 1 can be produced. These results indicate a 4 fold decrease in logic cost over a 5 year period.

Table 1. Progress of Altera's Low Cost Chip Families

Chip Family	Introduction	LE Count	Chip Cost	LE/$
Cyclone II	Feb 2005	33k	$22 [12]	1.5k
Cyclone	Sept 2002	20k	$20 [13]	1.0k
ACEX	Mar 2000	5k	$12 [14]	0.4k

This is only part of the picture as the falling transistor lengths also lead to faster logic and reduced delays while FPGA place and route tools have constantly evolved resulting in design resource reduction and further speed improvements. As by way of demonstration a number of designs were simulated for each device family using the software, Quartus II.

The results (Table 2) indicate a gradual speed improvement between generations with the case of the linear feedback shift register (LFSR) outlining the

effect of falling process sizes on register-to-register speeds: albeit the simplicity of the LFSR logic does not accurately represent the logic complexity of key search engines. The most representative design is clearly the key search engine that is the focus of this paper and the results for the "Cyclone" and "Cyclone II" devices relevantly reflect any performance improvement. However, comparison to the older "ACEX" device is problematic since the logic requirement is too much for these comparatively small devices. To provide a wider picture of FPGA progress the stream cipher Grain was chosen from the large key search system as reflective of overall system complexity but with resource needs small enough for it to be universally suitable.

Table 2. Progress in Performance of Altera Low Cost Chip Families

Function	ACEX EP1K100FC256-3 (Quartus II 3.0)	Cyclone EP1C20F324C8 (Quartus II 3.0)	Cyclone II EP2C35F484C8 (Quartus II 5.1)
LFSR	200MHz	275MHz	340MHz
Grain (t=16)	47MHz, 625LE	157MHz, 625LE	166MHz, 615LE
Key Search (n=p=16)	-	65MHz, 12094LE[a]	85MHz, 12434LE

Although the number of results shown in Table 2 is small the relevance of the designs mean that a reasonable evaluation of performance progress can be made. It is clear that for key search systems there was a large leap in performance between the "ACEX" and "Cyclone" series followed by a smaller performance improvement in the move to "Cyclone II". As cautious approach should be applied when considering brute force attacks it is reasonable to assume that the maximum 353% leap in clock speed observed for the stream cipher is representative of a long-term trend in hardware and design tool advances. This additional factor multiplies to that of falling logic cost and so yields the conclusion that FPGA based key search performance should be expected to almost quadruple with every chip generation. Thus the hypothesis, closely resembling Moore's law [11], that every 3 years the cost-performance ratio should increase by a factor of 4 is the pretence on which the calculations of future chip cost presented in Section 3.4 are made.

3.3 Extrapolating Results from Module Design

The EP1C20 simulation device was a very economical chip at its time of release, however it is a generation behind current low cost, 90nm FPGAs: Altera's "Cyclone II" and Xilinx's "Spartan III". This paper does not investigate the Xilinx offerings since the device families from the two manufactures are in direct competition and hence should produce comparable results, particularly as the

---

[a] Quartus II 5.1 used for design compilation

architectural differences provide little in the way of performance or resource advantages for the key search system. The most economical "Cyclone II" FPGA available is the EP2C35 with a 250k unit volume pricing of \$22 per chip [12]. Testing shows that the slowest speed-grade of this chip accommodates a n=32, p=16 key search system running at 86MHz.

Altera also offers a technology called "HardCopy" which is a form of Structured ASIC. In this a generic chip, with FPGA style cell structure, is mass manufactured without the top few metallisation layers, these layers are then added at a later stage to customise individual devices. Although such a manufacturing process sacrifices the FPGA's covert nature of design and programming flexibility it does allow lower unit costs and higher performance. This makes it attractive option for a key search unit and since it utilises a FPGA based design flow it also avoids the design complexity associated with conventional ASICs. The HC210W chip, currently the most economical of the high end "HardCopy II" devices, has a unit cost of \$15 when ordered in volumes of 100000 and a further non-recoverable engineering cost of \$225k [15]. Design simulations on Quartus II 5.0SP2 indicate that such a device allows a n=64, p=16 key search system running at a somewhat faster clock speed than a "Cyclone II" set-up. The exact speed is not easy tie down as the design process for "Hardcopy II" is based around the high performance "Stratix II" family which is on average half the speed [15] so the achieved 115MHz for a n=64, p=16 key search system on a EP2S60F484C4 leads to a estimated "Hardcopy II" clock speed of 230MHz. The larger EP2S90 is also compatible with the HC210W and preliminary investigations indicate a n=64, p=32 design would comfortably fit the chip.

3.4   Results and Evaluation

By combining the findings of Section 3.3 & 3.2 two tables may be constructed (Table 3 & 4) which evaluate the chip cost that an adversary would have to spend for the regular recovery of a key within the specified time-scale. It should be noted that in constructing and running a key search machine a substantial number of other costs would likely be incurred: PCB manufacture; cost of racks to hold the PCBs; cost of a computer to control the system; wiring; power regulators; programming chips (for FPGA based machines); air conditioning installation; cost of the building housing the machine; power consumption; logistical cost of handling high numbers of components. These costs may add a sizeable percentage to the overall expense of any brute force attempt but it is thought probable that FPGAs and Structured ASICs will still dominate a final bill of materials.

The figures outlined in Table 3 indicate that practical FPGA attacks which take under a month to obtain a key are probably 10 to 20 years away however the estimated costs for 1-year and 1-month searches in twenty years time appear be low enough to be applicable in many situations. The Structured ASIC results (Table 4) cause greater concern as even today \$8 million of devices would

---

[a] cost likely to be higher as below volume threshold

[b] non-recoverable engineering cost makes use of conventional FPGAs more cost effective

Table 3. Estimated Chip Cost of 80bit Key Brute Force using EP2C35 and future equivalents

Recovery Time	2005	2010	2015	2025
1 hour	$1.1 trillion	$270 billion	$68 billion	$1.1 billion
1 day	$45 billion	$11 billion	$2.8 billion	$44 million
1 month	$1.5 billion	$380 million	$94 million	$1.5 million[a]
1 year	$120 million	$31 million	$7.7 million[a]	$120k[a]

Table 4. Estimated Chip Cost of 80bit Key Brute Force using HC210 and future equivalents

Recovery Time	2005	2010	2015	2025
1 minute	$3.8 trillion	$960 billion	$240 billion	$3.8 billion
1 hour	$64 billion	$16 billion	$4.0 billion	$63 million
1 day	$2.7 billion	$670 million	$170 million	$2.8 million
1 month	$130 million	$33 million	$8.4 million	$350k[a]
1 year	$7.5 million	$2.1 million	$680k[a]	(480 chips)[b]

make a one year attack possible, this fact alone could immediately preclude such stream ciphers from a wide variety of commercial applications. By 2025 only data with very fast value/time roll-off appears suitable for use with the 80bit keys of eStream Profile II.

4 Conclusion

The key search machine described in this paper indicates that a single $15 chip would be capable of searching 39 billion key-plainstream pairs every second. Potential for further gains through unrolling of stream cipher initialisation or moving to a more memory based system might mean that even the slightly concerning figures presented for 80bit stream cipher keys, from a data protection perspective, are still be too optimistic. It is therefore apparent that in the medium-term such hardware orientated stream ciphers are not suited to the many applications where the data value for an adversary is still significant within a few days of the data's original production.

References

1. Preliminary Call for Stream Cipher Primitives. ECRYPT NoE, v1.0, 30th Nov 2004. http://www.ecrypt.eu.org/stream/call/
2. ECRYPT Yearly Report on Algorithms and Keysizes (2004). D.SPA.10 Rev 1.1, IST-2002-507932. ECRYPT, Mar 2005.
   http://www.ecrypt.eu.org/documents/D.SPA.10-1.1.pdf
3. Bernstein, D.J.: Understanding Brute Force. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/036 (2005). http://www.ecrypt.eu.org/stream

4. Cracking DES Secrets of Encryption Research, Wiretap Politics and Chip Design. Electronic Frontier Foundation, 1998. ISBN: 1-56592-520-3
5. Jean-Jacques Quisquater, Francois-Xavier Standaert: Exhaustive Key Search of the DES: Updates and Refinements. SHARCS 2005.
6. Hell, M., Johansson, T., Meier, W.: Grain A Stream Cipher for Constrained Environments. http://www.ecrypt.eu.org/stream/
7. Good, T., Chelton, W., Benaissa, M.: Review of stream cipher candidates from a low resource hardware perspective. SASC 2006, Stream Ciphers Revisited. http://www.ecrypt.eu.org/stvl/sasc2006/
8. Berbain, C., Gilbert, H., Maximov, A.: Cryptanalysis of Grain. SASC 2006, Stream Ciphers Revisited.
9. Hell, M., Johansson, T., Meier, W.: Grain - A Stream Cipher for Constrained Environments. Special Issue on Security of Computer Network and Mobile System. International Journal of Wireless and Mobile Computing, 2005.
10. Xilinx Staff: Celebrating 20 Years of Innovation. Xcell Journal. Xilinx, Issue 48 (Spring 2004). http://www.xilinx.com/publications/xcellonline/
11. Moore, G.E.: Cramming more components onto integrated circuits. Electronics. Volume 30, No 8, April 19, 1965. http://www.intel.com/technology/mooreslaw/
12. Altera: Altera's New Cyclone II FPGAs Offer 30 Percent Lower Costs Than Previous Generation. Altera Press Release, 28 June 2004.
Available at http://www.altera.com/corporate/news_room/releases/releases_archive/nr-releases_archive.html
13. Altera: Altera Completes First Generation Cyclone Device Family Rollout. Altera Press Release, 10 September 2003.
http://www.altera.com/corporate/news_room/releases/releases_archive/nr-releases_archive.html
14. Altera: Altera Introduces First Devices in Low-Cost ACEX Product Initiative. Altera Press Release, 13 March 2000.
15. Altera: Hardcopy 2 Backgrounder. Altera Virtual Press Kit Release 28 June 2004. Available at http://www.altera.com/corporate/news_room/presskit/hardcopyii/nr-hardcopyii.html

A Pseudocode

The following code describes the principle operations of the key search system outlined in Figure 3.

Part 1 - FRONT-END SEARCH

```
REPEAT UNTIL Set
SET Done to 0
SET IV to value stored externally

FOR each value of Counter
  FOR each cipher
    SET Key to {system FixedKey, unique CipherIdentifier, Counter}
    CALL rekeyCipher with Key and IV
  ENDFOR
  REPEAT UNTIL cipher initialisation period complete
  FOR each cipher
    FOR each pair of stored partial plainstreams
      SET Comparison to 0
      FOR each stored partial plainstream
        IF keystream same as plainstream THEN
          SET Comparison to 1
      ENDFOR
      SET searchFlag for pair to Comparison
    ENDFOR
  ENDFOR
  IF any searchFlag non-zero THEN
    STORE all searchFlags and Counter value in Primary FIFO
  ENDIF
ENDFOR

SET Done to 1
```

Part 2 - TRANSLATION UNIT

```
REPEAT UNTIL Primary FIFO not empty
READ searchFlags and Counter value from Primary FIFO
WHILE any searchFlag non-zero
  TRANSLATE most significant non-zero searchFlag into
            CipherIdentifier and PlainstreamSelect
  STORE CipherIdentifier, PlainstreamSelect and Counter value
        in Secondary FIFO
  SET most significant non-zero searchFlag to 0
WEND
```

Part 3 - BACK-END FULL TEST

```
REPEAT UNTIL Secondary FIFO not empty
READ CipherIdentifier, PlainstreamSelect and Counter value from
      Secondary FIFO
SET testKey to {system FixedKey, CipherIdentifier, Counter value}
CALL rekeyCipher with testKey and IV
REPEAT UNTIL cipher initialisation period complete
STORE first 80bits of keystream
FOR each memory
   IF keystream same as value at address PlainstreamSelect THEN
     SET memFlag for memory to 1
   ELSE
     SET memFlag for memory to 0
ENDFOR
IF any memFlag is non-zero THEN
   SET Bus Output to testKey
   REPEAT UNTIL Acknowledge
ENDIF
```

B   Resource Usage

In Table 5 resource usage is summarised for the main sub-systems of a brute force search design optimally configured for a Cyclone EP1C20F400 FPGA.

Table 5. Key Search System Resource Usage figures

Sub-System	Memory Bits	Registers	Logic Elements	No.
Counter	0	64	66	1
Stream Cipher	0	160-166	496-505	16
Comparison Unit	0	0	103-160	16
Plainstream Register	0	16	16	16
Primary FIFO	12160	46	71	1
Front-End (n=p=16)	12160	3189	10995	-
Expander-Decoder	0	73	204	1
Secondary FIFO	4736	46	64	1
Translation Unit	4736	119	269	-
Stream Cipher	0	166	493	1
Plainstream Memory	512	0	0	4
Compare	0	2	11	4
Check Control	0	20	26	1
Back-End Test	2048	212	581	-
Key-IV Register	0	109	109	1
Loading Control	0	0	20	1
Total	18944	3681	12018	-