International standards for stream ciphers: A progress report

Chris J. Mitchell and Alexander W. Dent
Information Security Group, Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
{c.mitchell, a.dent}@rhul.ac.uk

Abstract

The main objective of this paper is to review the current status of stream cipher standardisation. The hope is that, by doing so, the algorithms and techniques that are likely to be standardised at some point during the next year or so will be subjected to rigorous scrutiny by the cryptographic community.

1 Introduction

The history of stream cipher standardisation is a short and not terribly distinguished one. Until 2001, when work on the international encryption standard ISO/IEC 18033 commenced, there were no general purpose international encryption standards. This was the result of an explicit decision by ISO/IEC not to produce standards on encryption algorithms. This decision arose from an abortive attempt in the mid 1980s to produce an international standard for DES, which failed for political reasons.

As a result, the only standards covering stream ciphers (apart from the CTR, OFB and CFB block cipher modes of operation — discussed below) were those produced for specific applications. Most notable amongst these are probably the stream ciphers A5/1, A5/2 and A5/3 designed for use with GSM [4, 32]; the RC4 stream cipher (see, for example, [30]), standardised for use with IEEE 802.11 wireless networks as part of a system called *Wired Equivalent Privacy* (WEP), and also widely used elsewhere; and the UMTS/3GPP $f8$ function [3, 10]. We next briefly review these application-specific stream ciphers.

The stream ciphers A5/1 and A5/2 were the two schemes originally included in the GSM standards [32]. The use of these two schemes was mandatory, although details of the algorithms were kept secret. During the 1990s the algorithms were reverse engineered (see, for example, [4]); subsequently both algorithms have been shown to possess serious security vulnerabilities (see [1] for the most effective known cryptanalytic technique). In fact, A5/2 would appear to be a deliberately weakened version of A5/1, which was itself designed to offer a limited amount of security for political and export control reasons. Given these constraints, and the fact that they were designed in the mid-1980s, the fact that the algorithms have successfully been attacked should not be a surprise.

As a result a replacement algorithm, known as A5/3, has been standardised [12]; A5/3 is based on a mode of operation of the KASUMI block cipher [11, 26], and is believed to be secure. Problems, however, remain with GSM encryption, since, for backwards compatibility reasons, it is possible to force handsets to use A5/1 or A5/2, which can be used to compromise keys for A5/3 [1].

RC4 was devised by Rivest in the 1980s, and initially the details of its operation were not made public. Full details were not disclosed until 1994, and, as described in [14], it is currently widely used, including as one of the encryption techniques for SSL. However, its adoption as part of the IEEE 802.11b WEP protocol has caused the most publicity. As has been widely documented (see, for example, [13, 14, 31]), the way in which RC4 is used in WEP is insecure. More precisely, Fluhrer, Manton and Shamir [13] have succeeded in demonstrating the presence of undesirable properties in the RC4 key scheduling in the case when part of the key is known; it is precisely this property that gives rise to the WEP problems.

We have already mentioned that A5/3 is based on the use of a block cipher in a special way, i.e. a so-called 'mode of use' (discussed in a little more detail in Section 6). The same is true for the $f8$ function used for stream cipher encryption in the UMTS/3GPP mobile communications system [3, 10], which, like A5/3, is based on the KASUMI block cipher.

For a more detailed discussion of stream cipher and encryption standardisation see Chapter 4 of [7].

2 The ISO register of algorithms

One result of the ISO decision to abandon work on encryption standardisation was the creation of the ISO 'Register of cryptographic algorithms'. This register is a means by which encryption algorithms can be given standardised names, a step somewhat short of standardising algorithms. The

register was set up in the early 1990s under the terms of ISO/IEC 9979, the second edition of which was published in 1999 [18].

Although the register lists algorithms, is does not achieve two important things which one might expect of a standard. Firstly, inclusion in the register does not in any way imply that ISO recommends the algorithm for use. Secondly, there is no requirement on the proposer of an algorithm to provide a full description of it — in fact, details of the operation of a number of algorithms listed in the register, of which there are a total of 24, remain confidential. The only functional information that the user has to provide are input/output characteristics, so that a user can implement the algorithm as a 'black box' within a system, and test vectors, so that a user can test that a black box implementation of the algorithm is functioning correctly.

The register is available online at `www.iso-register.com`. It is interesting to note that RC4 is entry 0007 in the register, having been added in October 1994. Test vectors are given, but no details of the algorithm. Other stream ciphers listed in the register, such as B-CRYPT, BARAS, FWZ1, FSAngo, and BLIC, do not seem to have been as widely used as RC4.

3 The ISO/IEC encryption standard

In 2001 the ISO/IEC joint technical committee decided that the political climate had changed sufficiently to enable work to commence on an international encryption standard. The result is ISO/IEC 18033: *IT security techniques — Encryption algorithms*. This standard has four parts, as follows:

- Part 1: *General*, [23], giving general definitions and background information for the other three parts,
- Part 2: *Asymmetric ciphers*, [20],
- Part 3: *Block ciphers*, [21], and
- Part 4: *Stream ciphers*, [22].

It is interesting to note that when work started on this multi-part standard, the numbering was slightly different: Part 2 was for block ciphers, Part 3 for stream ciphers, and Part 4 for asymmetric ciphers. However it soon became clear that there was a shortage of proposals for standard stream cipher functions, and hence there was initially some doubt about whether the stream cipher part could be completed. As a result it was decided to use Part 4 for the stream cipher part, so that cancellation of this part would not cause problems for the numbering of the other parts.

Currently, Part 1 is out for FDIS (final draft international standard) ballot. This is the final stage before publication, and Part 1 should be published in late 2004 or early 2005. The other three parts are all currently out for FCD (final committee draft) ballot. This is the stage before FDIS, with publication expected during 2005.

4 ISO/IEC 18033-4 model for stream ciphers

The stream cipher draft standard defines a stream cipher to consist of a combination of a *keystream generator* (KG) and an *output function*. The KG is a technique for generating a pseudo-random sequence of symbols (typically bits) from a secret key, a starting variable, and, possibly, other inputs. The output function defines how the output of the KG (the *keystream*) shall be combined with the *plaintext* to yield the *ciphertext*. The plaintext and ciphertext are assumed to be sequences of symbols (typically bits).

4.1 Keystream generators

Two general classes of KG are defined, namely *synchronous* KGs and *self-synchronising* KGs. A synchronous KG is a finite state machine; its internal state determines both the next symbol output by the KG and the next state. A technique is required to determine the initial state based on a secret key and a starting variable. A self-synchronising KG, by contrast, has no internal stored state; each output symbol is a function of the previous ciphertext (actually a fixed number (r) of the immediately previous ciphertext symbols) and a secret key. An additional function is required to determine the initial r 'dummy' ciphertext symbols as a function of the key and a starting variable.

A stream cipher using a synchronous KG is called a synchronous stream cipher, and a stream cipher based on a self-synchronising KG is a self-synchronising stream cipher. One advantage of the latter class is that they are able to recover (after a delay) from loss of synchronisation between encrypter and decrypter, e.g. caused by the loss or gain of one or more 'blocks' of ciphertext symbols, where the length of a block of ciphertext symbols is equal to the number of symbols produced by the KG in one iteration.

Four specific examples of synchronous KGs are specified in the current draft of ISO/IEC 18033-4, together with one example of a self-synchronising KG. Of the four self-synchronous KGs, two are based on the use of a block cipher, and are described in Section 6, and the other two are 'dedicated' KGs, considered in Section 7. The single example of a self-synchronising KG is again based on use of a block cipher — see Section 6.

4.2 Output functions

The distinction between the KG and the output function that is used to combine the keystream with the plaintext may seem rather artificial. This is especially true given that, in practice, almost all stream ciphers use the exclusive-or output function described below, and it is difficult to envisage the need for any alternative. However, the exclusive-or output function is designed only to protect the confidentiality of the plaintext, and offers little protection against active attacks. The distinction between the KG and the output function allows the introduction of alternative methods of combining keystream with plaintext to provide security services other than confidentiality, notably an integrity service.

As well as two general classes of KGs, two output functions are specified in ISO/IEC 18033-4. Both are only defined for the case where the plaintext and keystream are both sequences of bits. Note that, for any mode, it is necessary to define not only how ciphertext is obtained from plaintext, but also the converse, i.e. the process must be reversible.

The first standardised function is the 'standard' exclusive-or output function. That is, each bit of the ciphertext is obtained by taking the exclusive-or (modulo 2 sum) of one bit of plaintext and one bit of keystream. I.e. if the plaintext is $p_0, p_1, \ldots$, and the keystream is $k_0, k_1, \ldots$, the ciphertext is $c_0, c_1, \ldots$, where $c_i = p_i + k_i \bmod 2$.

The second is a much more complex function known as MULTI-S01, that is only designed for use with a synchronous KG. This function, discussed in Section 5, is designed to offer both confidentiality and integrity protection to the plaintext. It is thus analogous to the authenticated-encryption techniques designed for block ciphers, such as OCB [29], EAX [2] and CCM [28, 35]. These techniques are also the subject of ongoing international standardisation efforts — the first working draft of what is intended to become ISO/IEC 19772 on authenticated encryption was published early in 2004 [24].

5 The MULTI-S01 output function

The MULTI-S01 output function was originally proposed by Furuya et al. [15]. Its use requires the selection of a security parameter n (typically $n = 64$ or 128).

The output function processes plaintext in blocks of n bits, and hence 'padding' of plaintext is typically required. Two n-bit blocks are appended to the end of the padded plaintext, where the first is made up of a keystream block, and the second is a redundancy block, e.g. a fixed public value. Each

n-bit plaintext block is encrypted by first ex-oring a block of keystream, then finite field multiplying by a fixed block of keystream, and finally ex-oring the result with the output of the first step of the encryption of the previous block.

No formal security results are known for MULTI-S01; currently, the best known attack has a probability of success of at most $(m-1)/(2^n-1)$, where an attacker knows the plaintext corresponding to a ciphertext containing m blocks [15].

6 Keystream generators from block ciphers

There are three very well-known techniques for using a block cipher as a stream cipher KG, namely the CTR (Counter), OFB (Output Feedback) and CFB (Ciphertext Feedback) modes of operation. All three of these modes of operation are included in the latest version of the international modes of operation standard, ISO/IEC 10116 [19], as well as in other national standards such as US NIST FIPS Special Publication 800-38A [27].

CTR and OFB modes enable any block cipher to be used to produce a synchronous KG. Both operate in a very similar way. That is, a sequence of inputs is given to the block cipher, along with a secret key, and part or all of the block cipher outputs constitute the keystream. The CTR mode simply uses a counter to vary the inputs to the block cipher, whereas the OFB mode uses the output produced by the block cipher as the next input. Using the terminology of the ISO/IEC 18033-4 model, the state for the CTR mode is simply the counter value, and the state for the OFB mode is the previous block cipher output.

The CFB mode enables any block cipher to be used to produce a self-synchronising KG. As with the CTR and OFB modes, a sequence of inputs is provided to the block cipher, along with a secret key, and part or all of the block cipher outputs constitute the keystream. Here the input to the KG is a block of ciphertext bits of the appropriate length.

These three schemes for constructing KGs would all appear to be satisfactory from a security perspective. However, they are all relatively slow by comparison with KGs designed specifically for high-speed operation. Given that one major application for stream ciphers is very high speed data encryption, this means that block cipher-based KGs are not the complete solution. Standardised dedicated KGs are also required.

For further details on block cipher modes of operation, and standards for such modes, see Chapter 5 of [7].

7 Dedicated keystream generators

The current draft of ISO/IEC 18033-4 contains two dedicated designs for synchronous KGs, namely SNOW 2.0 [9] and MUGI [34], both first published in 2002. Both are developments of previous designs.

7.1 SNOW 2.0

SNOW 2.0 is a direct descendant of SNOW 1.0 [8], first published in 2000, and has been designed to resist the attacks on SNOW 1.0 proposed by Coppersmith, Halevi and Jutla [5] and Hawkes and Rose [17].

SNOW 2.0 uses either a 128-bit or a 256-bit key, and employs a state variable consisting of 18 32-bit blocks (i.e. a total of 576 bits). The design employs a 16-stage GF(2^{32})-linear feedback shift register; 16 of the 18 32-bit state blocks make up the state of this register. It also uses a finite state machine with a state made up of the other two 32-bit state blocks; the way in which this state machine operates also depends on the shift register state variables. The keystream, which is generated 32 bits at a time, is a non-linear combination of the output of the finite state machine and two of the state variables of the linear feedback shift register.

7.2 MUGI

MUGI is based on a KG called Panama, proposed by Daemen and Clapp in 1998 [6]. The main goal of the changes to Panama was to make the design more efficient for hardware implementation and to make security analysis of the design simpler [34].

MUGI uses a 128-bit key and employs a state variable consisting of 19 64-bit blocks (i.e. a total of 1216 bits of internal state). The 'next state' function is designed somewhat similarly to a block cipher round function, and combines triples and pairs of 64-bit blocks to generate new values for 64-bit blocks.

8 Conclusions and challenges

A brief (partial) history of stream cipher standardisation has been provided. It should be clear that, although the draft international stream cipher standard is now approaching completion, the schemes it contains (apart from those based on block ciphers) are relatively new.

In particular, it would be enormously beneficial if three of the functions defined in the draft standard could be subjected to further careful scrutiny

now, rather than when the schemes are in more widespread use. These are as follows.

- *MULTI-S01.* The MULTI-S01 output function appears to be of possible practical benefit but has received very little attention in the literature.
- *SNOW 2.0* and *MUGI.* Both the dedicated synchronous KGs are of relatively recent design, and appear to have received much less attention than recently designed block ciphers such as AES. Papers have recently appeared giving some analysis of SNOW 2.0 [33] and MUGI [16], but there is little other discussion of these schemes in the published literature.

A further challenge for the cryptanalyst is posed by the combination of $f8$ and KASUMI, as used in 3GPP/UMTS [3, 10]. Whilst the primitive $f8$ has been proven secure in an extension of the ideal cipher model [25], this does not necessarily mean that the combination of $f8$ and KASUMI is secure.

Over and above the schemes in the standard, there are also some obvious gaps. First and foremost, apart from CFB mode there are no self-synchronising KGs; in particular there are no dedicated KGs of this type. Second, it is not clear whether SNOW 2.0 or MUGI are suitable for very high-speed hardware implementation, as would be required to bulk-encrypt ultra-high bandwidth communications channels. New proposals are therefore also needed for future standardisation.

References

[1] E. Barkan, E. Biham, and N. Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communications. In D. Boneh, editor, *Advances in Cryptology — CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 600–616. Springer-Verlag, Berlin, 2003.

[2] M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In B. Roy and W. Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer-Verlag, Berlin, 2004.

[3] K. Boman, G. Horn, P. Howard, and V. Niemi. UMTS security. In C. J. Mitchell, editor, *Security for mobility*, pages 99–121. IEE, London, 2004.

[4] M. Briceno, I. Goldberg, and D. Wagner. A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encryption algorithms. Available from `http://cryptome.org/gsm-a512.htm`, 1999.

[5] D. Coppersmith, S. Halevi, and C. Jutla. Cryptanalysis of stream ciphers with linear masking. In M. Yung, editor, *Advances in Cryptology — CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 515–532. Springer-Verlag, Berlin, 2002.

[6] J. Daemen and C. S. K. Clapp. Fast hashing and stream encryption with PANAMA. In S. Vaudenay, editor, *Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998, Proceedings*, volume 1372 of *Lecture Notes in Computer Science*, pages 60–74. Springer-Verlag, Berlin, 1998.

[7] A. W. Dent and C. J. Mitchell. *User's guide to cryptography and standards*. Artech House, 2004. To appear.

[8] P. Ekdahl and T. Johansson. SNOW — a new stream cipher. In *Proceedings of First Open NESSIE Workshop, KU-Leuven*, 2000.

[9] P. Ekdahl and T. Johansson. A new version of the stream cipher SNOW. In K. Nyberg and H. Heys, editors, *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002, Revised Papers*, volume 2595 of *Lecture Notes in Computer Science*, pages 47–61. Springer-Verlag, Berlin, 2003.

[10] European Telecommunications Standards Institute (ETSI). *3GPP TS 35.201, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: $f8$ and $f9$ Specification*, June 2002.

[11] European Telecommunications Standards Institute (ETSI). *3GPP TS 35.202, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification*, June 2002.

[12] European Telecommunications Standards Institute (ETSI). *3GPP TS 55.216, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the A5/3 Encryption Algorithm for GSM and ECSD, and the GEA3 Encryption Algorithm for GPRS; Document 1: A5/3 and GEA3 Specifications*, September 2003.

[13] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In S. Vaudenay and A. M. Youssef, editors, *Selected Areas in Cryptography, 8th Annual International Workshop, SAC 2001, Toronto, Ontario, Canada, August 16-17, 2001, Revised Papers*, volume 2259 of *Lecture Notes in Computer Science*, pages 1–24. Springer-Verlag, Berlin, 2001.

[14] S. Fluhrer, I. Mantin, and A. Shamir. Attacks on RC4 and WEP. *Cryptobytes*, 5:26–34, Summer/Fall 2002.

[15] S. Furuya, D. Watanabe, Y. Seto, and K. Takaragi. Integrity-aware mode of stream cipher. *IEICE Trans. Fundamentals*, E85-A:58–65, 2002.

[16] J. D. Golic. A weakness of the linear part of stream cipher MUGI. In B. Roy and W. Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 178–192. Springer-Verlag, Berlin, 2004.

[17] P. Hawkes and G. G. Rose. Guess-and-determine attacks on SNOW. In K. Nyberg and H. Heys, editors, *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002, Revised Papers*, volume 2595 of *Lecture Notes in Computer Science*, pages 37–46. Springer-Verlag, Berlin, 2003.

[18] International Organization for Standardization, Genève, Switzerland. *ISO/IEC 9979: 1999, Information technology — Security techniques — Procedures for the registration of cryptographic algorithms*, 2nd edition, 1999.

[19] International Organization for Standardization, Genève, Switzerland. *ISO/IEC FCD 10116, Information technology — Security techniques — Modes of operation for an n-bit block cipher*, 3rd edition, 2004.

[20] International Organization for Standardization, Genève, Switzerland. *ISO/IEC FCD 18033–2, IT Security techniques — Encryption Algorithms — Part 2: Asymmetric Ciphers*, 2004.

[21] International Organization for Standardization, Genève, Switzerland. *ISO/IEC FCD 18033–3, IT Security techniques — Encryption Algorithms — Part 3: Block Ciphers*, 2004.

[22] International Organization for Standardization, Genève, Switzerland. *ISO/IEC FCD 18033–4, IT Security techniques — Encryption Algorithms — Part 4: Stream Ciphers*, 2004.

[23] International Organization for Standardization, Genève, Switzerland. *ISO/IEC FDIS 18033–1, IT Security techniques — Encryption Algorithms — Part 1: General*, 2004.

[24] International Organization for Standardization, Genève, Switzerland. *ISO/IEC WD 19772: 2004, Information technology — Security techniques — Authenticated encryption mechanisms*, 2004.

[25] T. Iwata and T. Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In B. Roy and W. Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 427–445. Springer-Verlag, Berlin, 2004.

[26] T. Iwata, I. Mantin, and A. Shamir. On the pseudorandomness of KASUMI type permutations. In R. Safavi-Naini and J. Seberry, editors, *Information Security and Privacy, 8th Australasian Conference, ACISP 2003, Wollongong, Australia, July 9-11, 2003, Proceedings*, volume 2727 of *Lecture Notes in Computer Science*, pages 130–141. Springer-Verlag, Berlin, 2003.

[27] National Institute of Standards and Technology (NIST). *NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques*, December 2001.

[28] National Institute of Standards and Technology (NIST). *NIST Special Publication 800-38C, Draft Recommendation for Block Cipher Modes of Operation: The CCM Mode For Authentication and Confidentiality*, September 2003.

[29] P. Rogaway, M. Bellare, and J. Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security*, 6:365–403, 2003.

[30] B. Schneier. *Appled cryptography: Protocols, algorithms and source code in C*. John Wiley & Sons Inc., New York, 2nd edition, 1996.

[31] A. Stubblefield, J. Ioannidis, and A. D. Rubin. A key recovery attack on the 802.11b Wired Equivalent Privacy protocol (WEP). *ACM Transactions on Information and System Security*, 7:319–332, 2004.

[32] M. Walker and T. Wright. Security. In F. Hillebrand, editor, *GSM and UMTS: The creation of global mobile communication*, pages 385–406. John Wiley & Sons, 2002.

[33] D. Watanabe, A. Biryukov, and C. De Cannière. A distinguishing attack of SNOW 2.0 with linear masking method. In M. Matsui and

R. Zuccherato, editors, *Selected Areas in Cryptography, 10th Annual International Workshop, SAC 2003, Ottawa, Canada, August 14-15, 2003, Revised Papers*, volume 3006 of *Lecture Notes in Computer Science*, pages 222–233. Springer-Verlag, Berlin, 2004.

[34] D. Watanabe, S. Furuya, H. Yoshida, K. Takaragi, and B. Preneel. A new keystream generator MUGI. In J. Daemen and V. Rijmen, editors, *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, volume 2365 of *Lecture Notes in Computer Science*, pages 179–194. Springer-Verlag, Berlin, 2002.

[35] D. Whiting, R. Housley, and N. Ferguson. *RFC 3610, Counter with CBC-MAC (CCM)*. Internet Engineering Task Force, September 2003.