Proceedings of the 5th National Conference; INDIACom-2011
Computing For Nation Development, March 10 – 11, 2011
Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi

A Review of Py (Roo) Stream Cipher and its Variants

M. U. Bokhari[1], Shadab Alam[2] and **Faheem Syeed Masoodi[3]**
[1,2,3]Department of Computer Science, AMU, Aligarh
[1]mubokhari@gmail.com, [2]s4shadab@gmail.com and [3]masoodifahim@gmail.com

ABSTRACT
To overcome the deficiencies and non-standardization, and in search of a standard stream cipher that can provide more security and better performance than AES, ECRYPT (a consortium of European Research organizations) issued a call for stream cipher submission in the form of a project eSTREAM. The submissions were sought in two categories; Profile 1 (software based ciphers) and Profile 2 (hardware based ciphers). Py (Roo) was submitted by Eli Biham and Jeniffer Seberry in 2005 in software based cipher category was one of the candidates among total 34 submitted candidates. Py was very impressive in terms of efficiency and it was 2.5 times faster than RC4. But unfortunately it was found to be susceptible to some cryptanalytic attacks. An improved version of Py was submitted in form of Pypy but it was also not able to withstand the cryptanalytic attacks due to some errors in the design. To overcome the deficiencies in the previous versions of Py family, the designers of these ciphers proposed the tweaked versions of these ciphers in form of TPy, TPy6 and TPypy. Some attacks were also reported against these ciphers and possible solutions were also proposed in form of its variants RCR-32 and RCR-64 by Sekar, Paul & Preneel in 2007.
Our aim of this paper is to analyze various stream ciphers of Py (Roo) family, attacks against each of the ciphers and claims of the security of these ciphers for coming to a conclusion regarding security of these ciphers and future improvements.

KEYWORDS
Py (Roo), Py6, Pypy (Roopy), TPy, TPy6, TPypy, Rolling Arrays, RCR-32, RCR-64

1. INTRODUCTION

The hardware speed is increasing many folds with advancements in the technology. A general trend shows that speed of the systems doubles in every 18 months. The hardware cost is also decreasing drastically. It means that more and more powerful systems are available in hands of the hackers **[1]**. These developments require more and more secure ciphers and also at an efficient speed. Stream ciphers are an important class of symmetric encryption algorithm. Stream ciphers are used for encryption at fast speed. There were many stream ciphers proposed for NESSIE **[2]** project but it did not select any stream cipher for its portfolio due to shortcomings in all the candidates. RC4 was used as de-facto standard stream cipher till that time but it was also found to be prone to many attacks and slower speed [3, 4, 5…..11]. The block ciphers have been standardized by AES but there is not a standard for the stream ciphers.

To overcome all these problems and to come up with a standard, secure and efficient stream cipher, European Union has funded ECRYPT. The ECRYPT stream cipher project, eSTREAM is a multi year effort to identify promising new stream ciphers that can provide standard stream ciphers for wide adaptation. The call was made in two categories; Profile 1 (Stream cipher for software application) and Profile 2 (Stream Cipher for hardware application). This call for primitives resulted in total of 34 proposals in both the categories.

The stream cipher Py (Roo) analyzed in this paper was one of these candidates in profile 1 stream cipher category. After initial analysis of less than a year 28 ciphers were promoted in phase 2 including Py. But some attacks were reported against it and to improve these shortcomings a new version Pypy was introduced but that was also susceptible to some attacks. The eSTREAM committee eliminated Py from phase 3 but the report **[12]** says that "Py and its variants demonstrate a promising approach that might offer exceptional performance. Unfortunately, however, there is sufficient analysis **[13, 14, and 15]** to suggest that the submitted versions of the cipher demonstrate a weakness in the design."

To overcome the deficiencies in the previous variants of the Py, the designers had tweaked these ciphers and proposed three new ciphers TPy, TPypy, TPy6. Even though these tweaked variants have solved the problems of the previous known attacks against this family but new attacks were also reported against these new variants. In a new proposal Paul, Preneel & Sekar have introduced two new variants RCR-32 and RCR-64 and claimed it to be secure against any attack known for Py and its previous variants. But most of the claimed attacks against tweaked versions i.e. TPy, TPypy & TPy6 were claimed as non-attack by the designers of these ciphers.

In this paper we have tried to analyze the design and security specifications of Py cipher and its variants to assess the reliability of these ciphers and get a better understanding for the design of a stream cipher that is more secure as well as efficient. This paper studies the design specifications, implementation details and security attacks of Py cipher and its variants.

2. DESIGN SPECIFICATIONS

The main component in the design of Py (Roo) family of ciphers is rolling arrays. A rolling array is vector whose units are cyclically rotated and every rotation shifts its entries by one

 ISSN 0973-7529 ISBN 978-93-80544-00-7

location. Some additional basic operations are performed as part of the array rotation. A very important property of the rolling array is that if the same entry is accessed in two consecutive steps then we will get different content. This property is very useful for increasing the speed of mixing the internal state of the cipher. In the Py (Roo) stream cipher and its other variants two rolling arrays have been used and both affect each other by their operation. Permutation and swap operations are performed on one array and then the update operation accesses the other entries. One rolling array is a permutation P of all 256 byte values and other is an array Y of 260 words of 32 bits **[16]**. All the entries are rotated and oldest entries is updated. In this way indirect access is performed that add a large amount of complexity to the mixing process that make it complex for a cryptanalyst to follow.

3. Py (Roo)

Py (pronounced Roo) is a synchronous stream cipher designed in response to eSTREAM project call. The main component of this cipher design is rolling arrays. It also uses various other ideas from many other ciphers, like permutation and variable rotation. To some extent Py is similar to RC4 as it also uses the technique of random scuffle **[14]**. In Py all the array elements are also rotated in every round. The main strength of this cipher Py is its speed. It is 2.5 times faster than RC4. It takes less than 2.9 Cycles/byte on Pentium-III.

.Py is a stream cipher designed especially for very fast and secure encryption. It is intended for use with keys of upto 256 bits (32 bytes) and initial vector (IV) of upto 128 bits (16 bytes) but it can also be used with large keys of upto 256 bytes and IV sizes upto 64 bytes. The stream generated for a given pair of key and IV is restricted to length of upto 2^{64} bytes.

In the design of Py cipher two rolling arrays have been used. One array P is of 256 bytes that contains a permutation of all the values from 0…. to 255 and second array Y is an array of size 260 where each word is of 32 bit and are indexed as -3,… to 256. Both the arrays rotated in each step of cipher and two output words computed **[16]**. This word is updated by mixing two words of Y into it, where two words are indirectly selected from P and then variable rotation is performed on it. In this way the word is rotated by a number of bits that is calculated from another entry of P.

3.1 ATTACKS ON Py

Paul, Preneel and Sekar found a statistical bias in the distribution of the output words that can be used to construct a distinguisher that can work with $2^{83.2}$ random keys/IVs **[17]**. In that attack, the key stream can be distinguished from random with $2^{89.2}$ outputs. Later on Paul Crowley improved this attack by using hidden Markov model by a factor of about 2^{16} in the number of samples **[18]**. By using this model a distinguishing attack can be made against Py with 272 given bytes of output.

4. Py6

Py6 is a variant of Py with reduced internal state size. In this smaller variant of Py, the value of the Permutation P is reduced to a smaller size of 64 and Y to 68 entries. The word size of Y remains unchanged to 32 bits. This variant was proposed to achieve fast initialization with the speed remains the same as the speed of Py. The smaller size of internal states allows a much faster key setup and IV setup that is very attractive for encryption of short streams. This variant has smaller rolling arrays, thus its key setup and IV setup are much faster than of Py, and take 796 and 1464 cycles, respectively. The total number of cycles required by the key and IV setups is thus smaller than the key setup of RC4, and the stream generation is about 2.5 times faster than RC4.

Py6 differs from Py in the implementation that in this variant the difference free set had to be replaced by a difference free set modulo 64 and modulo 68. However, there cannot be 10 numbers in such a difference free set, thus in this variant the difference free set contains only the six values that are used as indices to the array P **[16]**. Also, in the generation of the permutation P the internal permutation cannot be used, thus it is removed from one location, and some rotations by eight bits are replaced by rotations by six bits in order to ensure full mixture of the data. As the indices in this variant are shorter, we restrict the length of the generated streams to 2^{40}.

4.1 ATTACKS ON Py6

As the design of Py6 is same as the Py the attacks which are applicable to Py are also applicable to Py6. Except those attacks that are applicable to Py, distinguishing attacks were reported against Py6 with 2^{68} data and comparable time by Paul and Preneel **[19].**

5. Pypy (Roopy)

Distinguishing attacks were reported against Py by exploiting some statistical bias in the design of Py. To overcome these shortcomings in the design of Py, a new improved version Pypy was proposed by the designers of Py. It takes every second word of the stream of Py (starting from the second word) and half of the outputs are discarded, i.e., the first output of the two outputs at each step is discarded **[20]**. Though, slower than Py, it is still about 1.5 times faster than RC4. Pypy follows the same structure, as of Py in terms of key setup, IV setup and implementation. The only difference being that the first output of the two outputs at each step is discarded.

5.1 ATTACKS ON Pypy

The IV setup of Py and Pypy are same. Wu and Preneel showed that there is serious flaw in the IV setup of Py and Pypy. In these ciphers, two key streams can be identical for every 2^{16} IV's for IV's with special difference. In this way key recovery attacks can be made against the ciphers Py, Pypy, Py6 with chosen IVs **[21].** This attack was subsequently improved by Isobe et al. They showed that 128 bit key can be recovered with a time complexity of 2^{48} **[22]**.

 ISSN 0973-7529 ISBN 978-93-80544-00-7

To overcome the deficiencies in the design of Py, Py6, and Pypy, the designers withdrew them and introduced three new modified or tweaked version of these ciphers that are TPy, Tpy6 & TPypy.

6. TPy, Tpy6 & TPypy

TPy, Tpy6 & TPypy came into existence as a result of major limitations of equivalent IV's found in all members of Py family. TPy, Tpy6 & TPypy are the enhanced versions of Py, Py6 & Pypy. Since the key setup & stream generation remain unaltered in the new design, both characteristics are same as of the previous version **[23]**. The tweaked IV setup may be very slightly slower than the original, as the modification is very small. The authors of Py claimed that there are no equivalents IVs in Py. When looking at the IV setup, and considering the two loops that mix the IV into EIV (on which the new attack is based), it becomes clear that the authors of Py, when tried to improve the IV setup, added the second loop in order to mix the IV better into the internal state. However, this extra mixing was a poor choice, as if only the first loop was proposed, the IV setup would not have the equivalent IVs problem. To overcome this problem the IV setup of all the three members of the Py family were tweaked. In the tweak of Py and Pypy, the second mixing loop of the IV were modified such that it will not lead to equivalent IVs, by ensuring that it is invertible (following the suggestion of **[17]**), in a way that prefer over removing this second loop. The final loop of the IV setup was changed slightly, also in order to ensure invertibility. In TPy6, unlike in Py6, the EIV rolling array is also doubled in size to ivsizeb*2. As a result, it limits the keys of TPy6 to be up to 64 bytes, and the IVs to be up to 32 bytes (both are larger than the minimal and recommended sizes). **[23]**

6.1 SECURITY ATTACKS

In **[24]** Sekar, Paul and Preneel published distinguishing attacks on Py, Pypy, TPy and TPypy with data complexities 2^{281} each**.** In **[25]** Sekar, Paul and Preneel showed new weaknesses in the stream ciphers TPy and Py. Exploiting these weaknesses distinguishing attacks on the ciphers are constructed where the best distinguisher requires $2^{268.6}$ data and comparable time. In **[26]** Sekar, Paul and Preneel mounted distinguishing attacks on TPy6 and Py6 with $2^{224.6}$ data and comparable time each. Further in **[27]** Sekar, Paul and Preneel detected related-key weaknesses in the Py family of ciphers including the strongest member TPypy. Under related keys they shown that a distinguishing attack on TPypy with data complexity $2^{193.7}$ which is lower than the previous best known attack on the cipher by a factor of 2^{88}. It was also shown in this paper that the above attack also works on the other members TPy, Pypy and Py. Later in **[28]**, the designers of Py Cipher have denied the attacks on TPy family especially on TPypy.

7. RCR-32 & RCR-64

In the process of getting a secure cipher Paul, Preneel & Sekar proposed two new ciphers RCR-32 and RCR-64 that are derived from TPypy & TPy respectively. The key and IV setup of the RCR-32 and RCR-64 are identical with TPy and TPypy. The only change in the design of these ciphers have been made that variable rotation of quantity s is replaced with constant rotation **[27]**.

7.1 SECURITY ATTACKS

These new variants were claimed to be secure from all attacks that have been made against previous version of the Py family. Any attack has not been reported against these ciphers till date.

CIPHER	SPEED
Py & TPy	2.80
Py6 & TPy6	2.80
Pypy & TPypy	4.58
RCR-32	4.45
RCR-64	2.70
RC4	7.30

Table 1: Speed of Py and its variant Ciphers in cycles/bytes on Intel Pentium-III processor

Attacks	Py6	Py	Pypy	TPy6	TPy	TPypy
Paul et al. [17]	X	$2^{89.2}$	X	X	$2^{89.2}$	X
Crowley [18]	X	2^{72}	X	X	2^{72}	X
Preneel-Paul [19]	$2^{68.6}$	X	X	$2^{68.6}$	X	X
Isobe et al. [22]	X	2^{24}	2^{24}	X	X	X
Sekar et al. [24]	X	2^{281}	2^{281}	X	2^{281}	2^{281}
Sekar et al. [25]	X	$2^{268.6}$	X	X	$2^{268.6}$	X
Sekar et al. [26]	$2^{224.6}$	X	X	$2^{224.6}$	X	X
Sekar et al. [27]	X	$2^{193.7}$	$2^{193.7}$	X	$2^{193.7}$	$2^{193.7}$
Preneel - Wu [29]	X	2^{24}	2^{24}	X	X	X

Table 2: Attacks on the Py-family of stream Ciphers
(It shows the complexity of attacks and X denotes that the attack does not work

8. CONCLUSION:

Though Py and its variants used a promising approach that can deliver high performance, analysis have proved susceptibility of these ciphers towards different attacks like distinguishing attack, key recovery attack and chosen IV attack. TPypy is considered to be strongest candidate over the entire family. Even though security claims have been made by Preneel et al

and Sekar et al. against TPypy but the designers have already specified that this cipher can only be assumed secure for 2^{64} bytes of key stream and hence counter claim by designers to prove it secure is also satisfactory. No other attack which is more powerful than these attacks and of less complexity has been reported against TPypy till date. Therefore the TPypy can be treated as secure cipher till any other claim or cryptanalysis comes against it.

REFERENCES

[1]. M.U. Bokhari, Shadab Alam, and Faheem Syed Masoodi " Coparative analysis of Py (Roo) family of stream ciphers" ICRITO-2010, November 2010, Faidabad (India) pp 667-671

[2]. NESSIE: New European Schemes for Signature, Integrity and Encryption, http://www.nessie.eu.org/nessie/.

[3]. Hal Finney, An RC4 Cycle that Can't Happen, Usenet newsgroup sci.crypt, September 1994.

[4]. S. R. Fluhrer, I. Mantin, and A. Shamir, \Weaknesses in the key scheduling algorithm of RC4." in Proceedings of Selected Areas in Cryptography {SAC'01 (S. Vaudenay and A. M. Youssef, eds.), Lecture Notes in ComputerScience, Vol. 2259, pp. 1{24, Springer-Verlag, Berlin, 2001.

[5]. S. R. Fluhrer and D. A. McGrew, \Statistical analysis of the alleged RC4 stream cipher." in Proceedings of Fast Software Encryption {FSE'00 (B. Schneier, ed.), Lecture Notes in Computer Science, Vol. 1978, pp. 19{30, Springer-Verlag, Berlin, 2000.

[6]. J.D. Golic, \Linear statistical weakness of alleged RC4 keystream generator." in Proceedings of Eurocrypt'97 (W. Fumy, ed.), Lecture Notes in Computer Science, Vol. 1233, pp. 226{238, Springer-Verlag, Berlin, 1997.

[7]. L.R. Knudsen, W. Meier, B. Preneel, V. Rijmen, and S. Verdoolaege, Analysis methods for (alleged) RC4." in Proceedings of Asiacrypt'98 (K. Ohta and D. Pei, eds.), Lecture Notes in Computer Science, Vol. 1514, pp. 327{341, Springer-Verlag, Berlin, 1998.

[8]. I. Mantin, Analysis of the Stream Cipher RC4, M.Sc. thesis, TheWeizmann Institute of Science, 2001. Available at http://www.wisdom.weizmann.ac.il/_itsik/RC4/rc4.html.

[9]. I. Mantin and A. Shamir, \A practical attack on broadcast RC4." in Proceedings of Fast Software Encryption { FSE'01 (M. Matsui, ed.), Lecture Notes in Computer Science, Vol. 2355, pp. 152{164, Springer-Verlag, Berlin, 2001.

[10]. I. Mironov, \(Not so) random shu_es of RC4." in Proceedings of Crypto'02 (M. Yung, ed.), Lecture Notes in Computer Science, Vol. 2442, pp. 304{319, Springer-Verlag, Berlin, 2002.

[11]. Souradyuti Paul and Bart Preneel, A new weakness in the RC4 keystream generator and an approach to improve the security of the cipher.", Proceedings of Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, Lecture Notes in Computer Science, Vol. 3017, pp. 245{259, Springer-Verlag, Berlin, 2004.

[12]. eSTREAM committee's "Short Report on the End of the Second Phase" available at http://www.ecrypt.eu.org/stream/

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

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

[15]. H. Wu and B. Preneel "Key Recovery Attack on Py and Pypy with Chosen IVs." available at http://www.ecrypt.eu.org/stream/papersdir/2006/052.pdf

[16]. E. Biham, J. Seberry, "Py (Roo): A Fast and Secure Stream Cipher Using Rolling Arrays." available at http://www.ecrypt.eu.org/stream/ciphers/py/py.ps, ecrypt submission 2005

[17]. S. Paul, B. Preneel, G. Sekar, "Distinguishing Attacks on the Stream Cipher Py," Fast Software Encryption-FSE 2006 (M. Robshaw, ed.), vol. 4047 of LNCS, pp. 405-421, Springer Verlag, 2006.

[18]. P. Crowley, "Improved Cryptanalysis of Py," Workshop Record of SASC 2006 - Stream Ciphers Revisited, ECRYPT Network of Excellence in Cryptology, February 2006, Leuven (Belgium), pp. 52-60 also available at http://www.ecrypt.eu.org/stream/papersdir/2006/010.pdf

[19]. S. Paul, B. Preneel "On the (In) security of Stream Ciphers Based on Arrays and Modular Addition,"Asiacrypt 2006 (X. Lai and K. Chen, eds.), vol. 4284 of LNCS, pp. 69-83, Springer-Verlag, 2006.

[20]. E. Biham, J. Seberry, "Pypy (Roopy): Another version of Py", ecrypt submission 2006

[21]. Hongjun Wu, Bart Preneel, "Key Recovery Attack on Py and Pypy with Chosen IVs", eSTREAM report 2006/052 (2006) URL: http://www.ecrypt.eu.org/stream/papers.html

[22]. T. Isobe, T. Ohigashi, H. Kuwakado M. Morii, "How to Break Py and Pypy by a Chosen-IV Attack," eSTREAM, ECRYPT Stream Cipher Project, Report 2006/060

[23]. Eli Biham, Jennifer Seberry, "Tweaking the IV Setup of the Py Family of Ciphers The Ciphers TPy, TPypy, and TPy6" , Published on the author's webpage at http://www.cs.technion.ac.il/_biham/, January 25, 2007.

[24]. Gautham Sekar, Souradyuti Paul, and Bart Preneel, "Weaknesses in the Pseudorandom Bit Generation Algorithms of the Stream Ciphers TPypy and TPy", available at http://eprint.iacr.org/2007/075.pdf 2,3,11

[25]. Gautham Sekar, Souradyuti Paul, and Bart Preneel "New Weaknesses in the Keystream Generation Algorithms of the Stream Ciphers TPy and Py" available at http://eprint.iacr.org/2007/230.pdf 2, 3,11

[26]. Gautham Sekar, Souradyuti Paul, and Bart Preneel "Attacks on the Stream Ciphers TPy6 and Py6 and

 ISSN 0973-7529 ISBN 978-93-80544-00-7

Design of New Ciphers TPy6-A and TPy6-B" available at http://eprint.iacr.org/2007/436.pdf. 2, 3

[27]. G. Sekar, S. Paul, B. Preneel, "Related-key Attacks on the Py-family of Ciphers and an Approach to Repair the Weaknesses," Indocrypt 2007

[28]. E. Biham, J. Seberry, "The Truth on TPy", February 2008, available at http://www.ecrypt.eu.org/stream/papersdir/2008/014.pdf

[29]. H.Wu and B.Preneel "Differential Cryptanalysis of Stream Ciphers Py, Py6 and Pypy", Eurocrypt 2007 (Moni Naor ed) Vol. 4515 of LNCS, pp 276-290, Springer-Verlag 2007. 3,11