Distinguishing Attacks on T-Functions

Simon Künzli[1], Pascal Junod[2], and Willi Meier[1]

[1] FH Aargau, 5210 Windisch, Switzerland
{s.kuenzli, w.meier}@fh-aargau.ch
[2] Nagravision SA (Kudelski Group), 1033 Cheseaux, Switzerland
pascal.junod@nagra.com

Abstract. Klimov and Shamir proposed a new class of simple cryptographic primitives named T-functions. For two concrete proposals based on the squaring operation, a single word T-function and a previously unbroken multi-word T-function with a 256-bit state, we describe an efficient distinguishing attack having a 2^{32} data complexity. Furthermore, Hong *et al.* recently proposed two fully specified stream ciphers, consisting of multi-word T-functions with 128-bit states and filtering functions. We describe distinguishing attacks having a 2^{22} and a 2^{34} data complexity, respectively. The attacks have been implemented.

Keywords: Stream cipher, T-function, square mapping, distinguishing attack, statistical cryptanalysis

1 Introduction

Binary additive stream ciphers encrypt a plaintext stream by combining it with a key stream by means of an XOR operation (the decryption simply being the XOR of the key stream with the ciphertext stream). The key stream consists of a pseudo-random bit sequence usually generated by iteration of an *update function*, the latter being initialized with a secret state. One expects that the sequence generated by a cryptographically secure stream cipher is statistically indistinguishable from a truly random sequence (and this for any adversary with some limited computational power), and that there exists no key-recovery attack better than brute-force.

Recently, Klimov and Shamir [7, 8, 9, 10, 6] proposed a new framework for highly efficient mappings which could be used as primitives in stream ciphers and other cryptographic schemes. These primitives consist of *triangular functions* (T-functions) which are built with help of fast arithmetic and Boolean operations widely available on high-end microprocessors or on dedicated hardware implementations; these mappings come with provable properties such as invertibility and a single-cycle structure. As an example, the mapping TF-0 is proposed in [7], which is defined by $x \mapsto x + (x^2 \vee C) \bmod 2^n$ for an n-bit state x and with $C \equiv 5, 7 \pmod 8$. As the maximal length of a cycle may be too short for typical values of n (e.g. $n = 64$), and as state-recovery attacks have been described [2, 8], TF-0 is not meant to be directly used for cryptographic purposes.

E. Dawson and S. Vaudenay (Eds.): Mycrypt 2005, LNCS 3715, pp. 2–15, 2005.

Considering cryptographic applications, several efficient multi-word T-functions are proposed in [9]. Some of these proposals have been broken by Mitra and Sarkar [13] using time-memory tradeoffs. Based on the results of Klimov and Shamir, a new class of multi-word T-functions and two fully specified stream ciphers have been proposed by Hong *et al.* [3, 4]. Their schemes TSC-1 and TSC-2 have a transparent design and allow for some flexibility.

1.1 Contributions of This Paper

In this paper, we analyse several proposals of T-functions and exhibit substantial weaknesses in some of these constructions. The flaws are extended to dedicated attacks.

First we analyse the statistical properties of the pure square mapping, which allows us to find an efficient distinguisher (with an expected 2^{32} data complexity) on TF-0 as well as on a previously unbroken multi-word mapping described in [9] and labeled here as TF-0m, both based on the squaring operation. TF-0m operates on a 256-bit state and the output sequence consists of the 32 most significant bits.

Then, we cryptanalyse the TSC-family of stream ciphers [4], which operates on a 128-bit state and outputs 32 bits of the state using a filtering function. We find a very efficient distinguisher for TSC-1 with an expected 2^{22} data complexity; for TSC-2, we describe a different distinguishing attack with an expected 2^{34} data complexity.

To confirm our theoretical results, the distinguishing attacks have been implemented and run many times with success. Our distinguishers have a negligible error probability and a remarkably small time complexity.

1.2 Notational Conventions

We analyse cryptographic schemes consisting of an internal state $x \in \mathcal{X}$, an update function $\mathsf{f} : \mathcal{X} \to \mathcal{X}$ and an output function $\mathsf{g} : \mathcal{X} \to \mathcal{Y}$. In the case where time instants are relevant, we will denote x^t the state at time t (distinction of powers will be clear from the context). Hence, the iterative scheme maps the state x^t to $x^{t+1} = \mathsf{f}(x^t)$ and outputs $y^t = \mathsf{g}(x^t)$. The seed of the iteration is obtained from the secret key with help of a key scheduling process. The keystream K consists in the concatenation of successive outputs, namely $K = y^0||y^1||\cdots$.

We assume throughout this paper the threat model of a known-plaintext attack, i.e., we assume to know some part of the keystream K. Our purpose is then to distinguish K from a uniformly distributed random sequence, or to recover the state at any time.

In the case where the state is a vector formed by some words, we will denote a single word by x_j and the state as $x = (x_0, x_1, \ldots)$. Adopting the common notation, $[x]_i$ is the $(i+1)$-st least significant bit-slice of the state, $[x]_0$ denoting the rightmost bit-slice. Consequently, $[x_j]_i$ is the $(i+1)$-st least significant bit of word j. The operation $\mathrm{msb}_m(x)$ states for the m most significant bits of x. Arithmetic operations are performed modulo 2^n with typical word size $n = 32$ or 64 bit. Boolean operations are performed on all n bits in parallel and are