RC4-Hash: A New Hash Function Based on RC4
(Extended Abstract)

Donghoon Chang[1], Kishan Chand Gupta[2], and Mridul Nandi[3]

[1] Center for Information Security Technologies(CIST), Korea University, Korea
dhchang@cist.korea.ac.kr
[2] Department of Combinatorics and Optimization, University of Waterloo, Canada
kgupta@math.uwaterloo.ca
[3] David R. Cheriton School of Computer Science, University of Waterloo, Canada
m2nandi@cs.uwaterloo.ca

Abstract. In this paper, we propose a new hash function based on RC4 and we call it RC4-Hash. This proposed hash function produces variable length hash output from 16 bytes to 64 bytes. Our RC4-Hash has several advantages over many popularly known hash functions. Its efficiency is comparable with widely used known hash function (e.g., SHA-1). Seen in the light of recent attacks on MD4, MD5, SHA-0, SHA-1 and on RIPEMD, there is a serious need to consider other hash function design strategies. We present a concrete hash function design with completely new internal structure. The security analysis of RC4-Hash can be made in the view of the security analysis of RC4 (which is well studied) as well as the attacks on different hash functions. Our hash function is very simple and rules out all possible generic attacks. To the best of our knowledge, the design criteria of our hash function is different from all previously known hash functions. We believe our hash function to be secure and will appreciate security analysis and any other comments.

Keywords: Hash Function, RC4, Collision Attack, Preimage Attack.

1 Introduction

Hash functions are of fundamental importance in cryptographic protocols. They compress a string of arbitrary length to a string of fixed length. We know that digital signatures are very important in information security. The security of digital signatures depends on the cryptographic strength of the underlying hash functions. Other applications of hash functions in cryptography are data integrity, time stamping, password verification, digital watermarking, group signature, e-cash and in many other cryptographic protocols.

Hash functions are usually designed from scratch or made out of a block cipher in a black box manner. Some of the well studied hash functions constructed from scratch are SHA-family [31,9], MD4 [26], MD5 [27], RIPEMD [25], Tiger [1], HAVAL [39] etc. Whereas PGV hash function [24], MDC2 [6] etc. are designed in a black box manner.

Since among SHA-family SHA-0 [31], SHA-1 [9] were broken by Wang *et al.* [35,36], we can not be confident about the security of other algorithms in the

R. Barua and T. Lange (Eds.): INDOCRYPT 2006, LNCS 4329, pp. 80–94, 2006.

SHA-family because their design principles are similar. Likewise MD4, MD5, RIPEMD and HAVAL were also broken [33,34,37,38]. So, we need to design new, variable length hash algorithms with different internal structures keeping security and efficiency in mind.

In response to the SHA-1 vulnerability [36] that was announced in Feb. 2005, NIST held a Cryptographic Hash Workshop on 2005 to solicit public input on its cryptographic hash function policy and standards. NIST continues to recommend a transition from SHA-1 to the larger approved hash functions (SHA-224, SHA-256, SHA-384, and SHA-512). In response to the workshop, NIST has also decided that it would be prudent in the long-term to develop an additional hash function through a public competition, similar to the development process for the block cipher in the Advanced Encryption Standard (AES).

It will be useful and interesting to propose some robust hash functions which are based on some well studied and structurally different from the broken class. In this direction we propose a hash function (RC4-Hash) whose basic structure is based on RC4. It also has the desirable advantage of variable length hash output. In fact our design provides hash output from 16 bytes to 64 bytes with little or no modification in the actual algorithm. It provides a wide range of security depending on the applications. In this context it may be noted that there are very few hash families providing variable size hash output. We provide security analysis against meaningful known attacks. We take care of the weakness of RC4 in a manner such that it will not affect the security of the Hash function. Many results on RC4 can be used to show the security of RC4-Hash against known attacks and importantly resistances against attacks by Wang *et al.* and Kelsey-Schneier second preimage attack [16]. Its efficiency is also comparable with SHA-1.

The rest of the paper is organized as follows. In Section 2 we give a simple description and some of the security analysis of RC4. We also give a short note on hash functions. RC4 based hash function is analyzed in Section 3 followed by a security/performance analysis of RC4-Hash in Section 4. We conclude in Section 5.

2 Preliminaries

We first describe the RC4 algorithm and its known security analysis which are relevant to this paper. Then we give a short note on hash functions. RC4 was designed by Ron Rivest in 1987 and kept as a trade secret until it leaked out in 1994. It consists of a table of all the 256 possible 8-bit words and two 8-bit pointers. Thus it has a huge internal state of $log_2(2^8! \times (2^8)^2) \approx 1700$ bits. For a detailed discussion on RC4 see Master's thesis of Itsik Mantin [18].

2.1 RC4 Algorithm

Let $[N] := [0, N-1] := \{0, 1, \cdots, N-1\}$ and $\mathrm{Perm}(A)$ be the set of all permutations on A. In this paper, we will be interested on $\mathsf{Perm}([N])$ (or we write Perm), where $N = 256 = 2^8$. For $S \in \mathsf{Perm}$, we denote $S[i]$ to the value of the permutation S at the position $i \in [N]$. In this paper, the addition modulo N is denoted