6.857 Homework	**Problem Set 1**	# 1-3 - The SHA-3 Competition
Anonymous		March 12, 2009

1 Skein

The Skein Hash Function Family, proposed by Ferguson et Al., uses the ThreeFish block cipher as its compression function and the UBI (unique block iteration) chaining mode. The authors introduce a key element that they call a *tweak*. For each block of the plaintext, the tweak contains three fields: two flags that indicate if the current block is the first or the last one of the plaintext and a value indicating the number of bits compressed so far. The tweak is meant to protect against cut-and-paste attacks. Each ThreeFish invocation, thus, receives as input the cyphertext from the previous compression block in the chain xor-ed with the plaintext for that block, the tweak and the plaintext for this block.

Although the authors claim security for the ThreeFish cipher, only the compression function claims are substantiated with proofs. The authors have compiled the proofs in a separate paper [1] (see second-to-last paragraph on page 30) and just provided discussions or sketches of proofs in this report. For each of the following security claims, the authors state that they provided proofs in [1].

- Indifferentiability from a random oracle (top of page 32),
- Pseudorandomness of the keyed Skein (page 31)
- Collision resistance (bottom of page 30)
- Preimage resistance (page 28 and bottom of 29)

The authors also prove the following

- Based on the assumption that ThreeFish is PRP, Skein in the keyed mode for a MAC is not subject to length extension attack (page 31)
- If the compression function is PRF, so is Skein in HMAC mode (page 32).
- When used as a stream cipher, Skein is indistinguishable from random and when used as a PRNG it is forward secure, if ThreeFish is PRP (top of 33).
- Resistance against r-collisions (page 28).

It seems that the authors are proving resistance to computing near misses, that is finding two messages whose hashes have small Hamming distance (they claim resistance on the top of 29 and state it is backed by proofs on bottom of 29).

Although the authors present empirical results, they do not prove the following

- Preimage resistance and collision resistance of the ThreeFish compression function. It is usual to assume the security of the atomic primitive when proving results about the overall scheme.

- Resistance to differential and linear cryptanalysis. They provide an attack example in Section 9.3.4 (starting on page 54) to show how unlikely linear and differential attacks are to be succesful by explaining that Skein maximizes the amount of diffusion which has been a cause of previous such attacks on old tools.

[1] M. Bellare, T. Kohno, S. Lucks, N. Ferguson, B. Schneier, D. Whiting, J. Callas, and J. Walker "Provable Security Support for the Skein Hash Family"

2 ECHO

ECHO is a hash function that is built off of AES. The paper mostly proves properties regarding its modes of operation:

- It is built using the Merkle-Damgard construction, which means that it has collision-resistance and preimage-resistance provided that the compression function has these properties. [p. 40]
- It uses the HAIFA framework, which makes the construction resistant to length-extension attacks when used for a MAC [p. 40].
- It uses a "double-pipe" strategy (the chaining value is twice as long as the output value). They claim that this makes the scheme resistant to multi-collision attacks (assuming the compression function is sufficiently secure), as well as herding attacks. [p. 41] (They do not prove these claims, although they reference the literature about double-pipes.)

The paper proves some claims about its resistance to certain forms of cryptanalysis [pp. 28-32], though not for the algorithm as a whole.

The paper offers empirical rather than mathematical arguments for resistance to linear cryptanalysis and differential crytanalysis [p25]. They base these claims on the general agreement that AES is a good source of confusion and diffusion, so that AES is resistant to differential and linear cryptanalysis, and thus that the hash function (which is based on AES) is also resistant.

3 Dynamic SHA2

The main principle for Dynamic SHA2 is "When the message is changed, the calculation will be different" [pg 2]. Overall, the author uses parts of SHA-2 and introduces this design principle in order to strengthen the design. Some of the main features are

- Dynamic SHA2 uses the G, R and ROTR functions from SHA2
- Dynamic SHA2 does not use any constants. The author claims that this is because it is already "secure enough"[pg 11]. Yet later the author claims that Dynamic SHA2 would be more secure with the use of a constant(s).[pg 19]
- Dynamic SHA2 is preimage resistant since the probability of an attacker being able to guess the individual parts and parameters used is the same as the probability of the attacker being able to randomly guess the preimage. [section 3.3 pg 12]

- Dynamic SHA2 is second preimage resistant. [section 3.4 pg 12]
- Dynamic SHA2 is collision resistant. Even an birthday attack will be a workload $O(2^{112})$ for the 224-bit version. [section 3.5 pg 14]
- It is resistant to multi-block collision attacks and generic length extension attacks. [pg 34, no concrete proof]
- The author demonstrates a much better cipher by way of the paper itself, since it is impossible to understand the paper because of many English and logic errors. [all pages]
  Example: "So the probability of find out a message that has the given given output is $2^{-512} \times 2^{128} = 2^{384}$." [pg 19]

4 ESSENCE

ESSENCE is a hybrid cryptographic hashing algorithm based on Merkle hash trees combined with Merkle-Damgard iterative hashing structures. It is designed to be flexible and attempts to balance ease of implementation and security concerns.

The ESSENCE algorithm's cryptographic security receives a fairly complete treatment in the NIST submission, concentrating mostly on the security of its compression functions. ESSENCE claims the following security properties:

- In order to prevent cache-timing attacks, The ESSENCE algorithm can be implemented with constant execution times (at a cost of performance).
- To protect from side-channel attacks such as memory leaks or fault analysis, ESSENCE compression functions can also be implemented entirely within the register file of a processor.
- ESSENCE uses a so-called "G" Function as its compression function. It consists of two applications of the "E" permutation with an "F" feedback function. The two are designed to maximize resistance to correlation attacks and linear analysis. They were designed with a machine learning algorithm that uses resistance to linear and differential cryptanalysis as a candidate feedback function. The resulting function is shown to be resistant to differential and linear analysis if the algorithm is stepped at least $N = 24$ times. Although this analysis shows resistance to such attacks, this does not prove it is secure.
- The submission provides a proof for resistance against length extension attacks. ESSENCE achieves this by using two-layer hashing–a lower level hashed in Merkel-Damgard fashion, and an upper layer that uses a separate "running hash."
- The work required to find a first pre-image is estimated to be on the order of a brute force search (2^n for an n-bit hash), but this is provided without proof. Similarly, the authors "expects" the scheme to be collision resistant if the compression function is, but does not provide any proof.

5 Our Recommendation

We believe that Skein has the largest amount of provable security. We argue by elimination. Essence seems the weakest scheme. It does not prove any properties regarding preimage resistance and collision resistance. The author says that he “expects” the scheme to have these properties given that the compression function has these properties, but he does not discuss or reference any proofs. Both ECHO and Dynamic SHA-2 prove preimage and collision resistance and do not prove indistinguishability from a random oracle and pseudorandomness, thus having a strict subset of the provable properties of Skein.