X-Sieve: CMU Sieve 2.2
Date: Fri, 13 Apr 2007 15:43:28 +0300
From: Carmi Gressel <carmi@fortressgb.com>
Subject: Hash Algorithm Requirements and Evaluation Criteria
To: william.burr@nist.gov, hash-function@nist.gov
X-Mailer: Microsoft Office Outlook 11
Thread-index: Acd9yVWtBzgH0vW4TdWUrGoTdRVVxQ==
X-Proofpoint-Virus-Version: vendor=fsecure engine=4.65.5502:2.3.11,1.2.37,4.0.164
definitions=2007-04-13_04:2007-04-11,2007-04-13,2007-04-13 signatures=0
X-PP-SpamDetails: rule=spampolicy2_notspam policy=spampolicy2 score=0 spamscore=0
ipscore=0 phishscore=0 adultscore=0 classifier=spam adjust=0 reason=mlx engine=3.1.0-
0703060001 definitions=main-0704130007
X-PP-SpamScore: 0
X-NIST-MailScanner: Found to be clean
X-NIST-MailScanner-From: carmi@fortressgb.com

Dear Bill and Shu-jen:

Attached is a letter and an explanatory document, which had I posted,
would have ended in a dustbin. The Israeli Post Office was on a slowdown-
until yesterday now it's a full fledged strike- and Fedex doesn't visit us
regularly on a Friday.

Bill, Orr did his last review on my article- I cleaned up the explicit
drawings to show principles and Orr and I cleaned up a lot of
the text. It will be much easier to skim over, if there's interest.

I'm not trying to jump the gun- but I think that you would
note that a good Stream Cipher construct is extremely efficient,
especially when it adopts a Hash configuration for efficient key/IV loads.
(We're not the only ones).

Sincerely,

Carmi

Carmi Gressel
Fortress GB Ltd
Omer Industrial Park 8B
Omer 84965, ISRAEL
Mb +972-54-7776 059 Hm +972-8-9920518
Fx +972-8-6466 729 Skype @FGB - Carmi.Gressel
FGB -Tel IL +972-8-6909 727 UK +44-207-8747 595

FortressGB

April 13, 2007
London, UK
Omer, Israel

Attn: Hash Algorithm Requirements and Evaluation Criteria
William Burr
Mgr, Security Technology Group
100 Bureau Dr. Stop 8930
Gaithersburg, MD 20899

Dear Colleagues:

We are pleased to offer opinions relating to the requirements (and criteria) for accepting, evaluating and judging contesting hash algorithms. The suggestions are based on our experience with Common Criteria (security), our intimate knowledge of hardware and software/firmware security, and our experience transferring cryptographic technology to vendors and major silicon fabs.

We endorse and suggest expanding NIST's goal of having more approved algorithms. We stress our belief that algorithms biased toward efficient hardware are particularly attractive. They are a basis for stronger permutations, massive diffusion, lowest energy per encoded bit, message modification resistance and highest throughput. We know from our own experience, that embedding a single algorithm on an existing chip can add less than 2¢ to the cost per compact algorithm with present technologies. Putting 3 or 4 algorithms based on diverse principles ("look-alikes" to the user) may be the best assurance for a long life security strategy. In the light of attractive hardware and software options, we suggest replacing the criterion "simplicity" with "S/W-H/W compatibility" and "simplicity of implementation", as we also recognize the need for flexible cost effective software biased implementations for many applications.

Section A.1, as stated, grants NIST the greatest latitude to find the best and most cost effective hash functions. We agree with Bill Burr that this should not be changed. We look forward to submitting our ZK-Crypt hash algorithm implementations.

Sincerely,

Carmi Gressel
CTO, FortressGB

Suggestions Relating to new Hash Functions (Augments and revisions to FIPS 180-2)

The following relates to the preamble, to the candidate's submission and to the final evaluation.

1) "NIST has decided to develop one or more additional hash function"-
We suggest "to develop additional hash functions, preferably using more than one design principle". The advantages surpass the cost and will enable easy transitions between diverse purpose or throughput algorithms. This might be equated to the practice of implementing a hardware DH, RSA and Elliptic Curve module on silicon. Legacy devices in transition (working against servers) could choose an optimized firmware implementation before moving to a faster, lower cost high throughput hardware oriented algorithm. Raising the security of a particular function or switching to what may turn out to be a more secure algorithm should be seamless, without further investment.

2) Speed/throughput is of utmost importance for present and future applications; e.g., transparent safe booting, rapid broadband downloading, and authenticating large memory structures. With current functions, users find it difficult to afford the time required for regular integrity checks of sensitive memory systems. Highest throughput can only be achieved with optimized hardware. Optimized software and hardware algorithms should have reasonably compliant implementations.

3) Total energy consumption per hashed Message bit is a primary concern for all platforms in many applications. The energy consumption per processed bit should be a benchmarking factor for both hardware and software implemented functions [Mbits/mWatt sec]. Energy consumption varies between different compression systems, and can be as high as one hundred to one; e.g., block ciphers to equivalent "block configured" stream ciphers.

4) Message modification is probably the hacker's first choice for a rogue attack; therefore, methods and rationale for precluding reconciliation of the internal state resulting from such attack should be mentioned explicitly, not implied as "any cryptographic attacks" [B1] or "resistance to generic attacks" [C1].

B. Comments on the Proposed Draft Submission Requirements

B.1 In addition to parameters which are mentioned, speed, and energy consumption per processed bit, should be included in the design rationale.

Hardware design applications should relate to silicon layout constraints.

B.2 For hardware implementations a hi-level language hardware gate equivalent and speed estimate is a necessary equivalent to an "optimized [ANSI C] implementation" and should be required.
NIST should specify/provide one or more ANSI C randomality test benches for evaluating submissions.

C. Augmenting the Proposed Draft Evaluation Criteria

- Explicit defense mechanisms against message modification; e.g., the ability to prevent a message change without detection.
- First order diffusion of message bits (documenting the number of internal binary variables potentially affected by a single message bit at a single step).
- Replace "Simplicity" with "Compatibility of hardware and software implementations" and "Simplicity of implementation" (functions in a software version should be limited to accelerated word manipulation functions; e.g., Rotate, Complement, XOR, AND and OR).
- Per encoded bit energy consumption (measured in [Mbit/mWatt sec]).
- Speed/throughput (increasingly important for almost all applications measured in [Gbits/sec]).
- Minimum acceptable benchmarked randomality results (randomality tests are important for identifying distinguishing features).