Subject: OFFICIAL COMMENT: Blender
From: v.klima@volny.cz
Date: Fri, 19 Dec 2008 13:08:21 +0100 (CET)
To: hash-function@nist.gov
CC: hash-forum@nist.gov

```
Hi all,
in the enclosed paper we describe a near-collision attack (225/256 bits)on
BLENDER-256. It is also available on
http://cryptography.hyperlink.cz/BMW/near_collision_blender.pdf

Best regards,
Vlastimil Klima
```

near_collision_blender.pdf	**Content-Type:** application/pdf **Content-Encoding:** base64

Subject: OFFICIAL COMMENT: Blender
From: "Colin Bradbury" <cbradbury@luxtera.com>
Date: Mon, 22 Dec 2008 08:52:07 -0800
To: <hash-function@nist.gov>
CC: <hash-forum@nist.gov>

Security Update:

The security strength against a 4-way collision is (n+3)/2 bits and against a 16-way collision is (n+4)/2 bits.

Dr. Colin Bradbury (author)

Subject: OFFICIAL COMMENT: Blender
From: v.klima@volny.cz
Date: Fri, 02 Jan 2009 15:17:38 +0100 (CET)
To: hash-function@nist.gov
CC: hash-forum@nist.gov

```
Huge multicollisions and multipreimages of hash functions BLENDER-n

In the enclosed paper we present a 2^{2n}- multicollision and multipreimage
attack on the hash function Blender-n for all output sizes n = 224,
256, 384 and 512. The complexity and memory requirement for finding
2^{2n} multipreimages (multicollisions) is roughly 10 times more than
finding a collision for n/2-bit random hash function, i.e. 10*2^{n/4}.
The paper is also available on
http://cryptography.hyperlink.cz/BMW/collisions_and_preimages_blender.pdf

Best regards,
Vlastimil Klima
```

collisions_and_preimages_blender.pdf	**Content-Type:**	application/pdf
	Content-Encoding:	base64

Subject: OFFICIAL COMMENT: Blender
From: "Janet Bradbury" <colinb@colinb.cts.com>
Date: Sun, 22 Mar 2009 10:07:43 -0700
To: <hash-function@nist.gov>
CC: <hash-forum@nist.gov>

The HALL of SHAME has a special place for those who believe the fallacies in theoretical attacks. The attacks to date don't work with effort less than brute force. Identifying the fallacies is left as an exercise for the students.

The reference implementation has a typographical error detected by J. Forsythe at Fortify. The line

```
++ss.sourceDataLength2 [3];  // and the next one, etc.
```

should read

```
++ss.sourceDataLength2 [ 2 ];  // and the next one, etc.
```

With a view to deploying the algorithm in embedded systems the field, we have conducted an optimization exercise on the 8-bit implementation. The first step was to construct a "typical" implementation in C and then replace selected sections of code with the assembly language equivalent. This yielded the following numbers for hashing a 1000 bit message of random data.

Digest size	Reference (clock cycles)	Typical C	Typical C + Assembler	% gain over Reference
512	589296	309744	96600	83.61
384	534132	276336	87852	83.55
256	382224	233088	53652	85.96
224	370416	219768	50160	86.46
160	358608	206712	46308	87.09

Similar gains are expected by repeating the exercise with the 32-bit and 64-bit implementations. Naturally, the hardware performance is one word per clock at whatever clock speed the technology supports; all software numbers are a price/performance tradeoff.

The last step of the optimization exercise was to determine the cost of cross-linking the checksums in a manner similar to the main algorithm (speed versus security). The basic concept is that whenever a checksum addition produces a carry out, rotate the other checksum right by seven bits. The pseudo-code for this is as follows:

```
Checksum1 = Checksum2 = 0;
For (i=0; i<numWords; ++i) {
    [carry, Checksum1] = Checksum1 + W [i];
    if (carry != 0)    Checksum2 = ROTR7 (Checksum2);
    [carry, Checksum2] = Checksum2 + (~W [i]);
    if (carry != 0)    Checksum1 = ROTR7 (Checksum1);
}
```

Statistics from the 256-bit shortMsgKAT show that the first rotation occurs 48% of the time (in round numbers), the second rotation occurs 50% of the time, both rotations together occur 16% of the time, and neither rotation occurs 18% of the time. The performance hit for doing this is less than three percent of the optimized assembler timing for all cases. NIST can, of course, substitute their own checksum routines for a slightly different cost; NIST has reserved the right "to select a different security/performance tradeoff than originally specified by the submitter".

The more secure variant outlined above has now completed integration and beta testing and will be available for customer evaluation in the near future. KATs are available on request.

Colin Bradbury, MBA, Ph.D.
Blender Author.

From: Colin Bradbury [cbradbury@luxtera.com]

Sent: Tuesday, June 30, 2009 9:10 PM

To: hash-function@nist.gov

Cc: hash-forum@nist.gov

Subject: OFFICIAL COMMENT: Blender

Scorecard: Blender 3, attackers 0

A paraphrase of the attack by Vlastimil Klima is given below for anyone who missed the fallacy in the original paper. When the real costs are carefully evaluated, the other attacks have time*memory metrics greater than 2^n so those are disguised brute-force attacks. There are now over a thousand units deployed in the field using the stronger checksums as detailed in the previous Official Comment. I hereby grant everyone permission to use these algorithms free of charge. Thank you all for playing.

Regards, Colin Bradbury, MBA, Ph.D.
Blender Author.

HUGE MULTICOLLISIONS ON I-40

Interstate Highway 40 contains 2^12 kilometers of roadway. Assume that each kilometer is delimited by “milestones”. The highway is also subdivided longitudinally into a number of traffic lanes. We have found that a vehicle can change from one traffic lane to the next in ten decameters of travel along the highway and then change back into the previous traffic lane in the next ten decameters. If this maneuver is started in the #1traffic lane at one of the milestones and repeated fifty times, the vehicle ends up back in the #1 traffic lane at the next milestone. We then repeat this whole procedure until the vehicle has passed one of the milestones twice. By the birthday paradox, we expect this to happen within 2^6 kilometers. The milestone that has been passed twice is a FIXED POINT on the highway. Repeatedly traveling along the path between the first passing of that milestone and the second (without leaving the highway) results in huge multi-collisions. We will now show that ….