Subject: OFFICIAL COMMENT: CHI
From: Douglas Held <dheld@fortify.com>
Date: Mon, 9 Feb 2009 23:15:05 +0000
To: hash-function@nist.gov
CC: hash-forum@nist.gov, Joy Forsythe <jforsythe@fortify.com>

Hello,

The CHI submission seems to be missing genKAT.c, so the Reference_Implementation cannot be built without creating a little bit of code.

Is this file available for review?

```
host10:CHI dougheld$ grep -r genKAT .
./Optimized_32_bit/Makefile:KAT_OBJS = genKAT.o
./Optimized_32_bit/optimized_32.vcproj:
RelativePath="..\Reference_Implementation\genKAT.c"
./Optimized_64_bit/Makefile:KAT_OBJS = genKAT.o
./Optimized_64_bit/optimized_64.vcproj:
RelativePath="..\Reference_Implementation\genKAT.c"
./Reference_Implementation/Makefile:KAT_OBJS = genKAT.o
./Reference_Implementation/reference.vcproj:                RelativePath=".\genKAT.c"

host10:CHI dougheld$ find . | grep genKAT
host10:CHI dougheld$
```

Kind Regards,
Douglas Held
Fortify Software

Subject: Re: OFFICIAL COMMENT: CHI
From: Larry Bassham <lbassham@nist.gov>
Date: Tue, 10 Feb 2009 11:03:43 -0500
To: Douglas Held <dheld@fortify.com>

Two things. First, a copy of genKAT.c can be found at http://csrc.nist.gov/groups/ST/hash/sha-3/documents/KAT1.zip
Second, please refrain from using OFFICIAL COMMENT to ask administrative questions. The OFFICIAL COMMENT are for comments regarding the appropriateness of an algorithm being selected as the SHA-3 standard.

If you have questions like this in the future, feel free to email myself (lbassham@nist.gov), Shu-jen Chang (shu-jen.chang@nist.gov), or Sara Caswell (sara@nist.gov).

Larry Bassham

On Feb 9, 2009, at 6:15 PM, Douglas Held wrote:

> Hello,
>
> The CHI submission seems to be missing genKAT.c, so the Reference_Implementation cannot be built without creating a little bit of code.
>
> Is this file available for review?
>
> ```
> host10:CHI dougheld$ grep -r genKAT .
> ./Optimized_32_bit/Makefile:KAT_OBJS = genKAT.o
> ./Optimized_32_bit/optimized_32.vcproj:
> RelativePath="..\Reference_Implementation\genKAT.c"
> ./Optimized_64_bit/Makefile:KAT_OBJS = genKAT.o
> ./Optimized_64_bit/optimized_64.vcproj:
> RelativePath="..\Reference_Implementation\genKAT.c"
> ./Reference_Implementation/Makefile:KAT_OBJS = genKAT.o
> ./Reference_Implementation/reference.vcproj:                RelativePath=".\genKAT.c"
>
> host10:CHI dougheld$ find . | grep genKAT
> host10:CHI dougheld$
> ```
>
> Kind Regards,
> Douglas Held
> Fortify Software

From: Tor.Bjorstad@ii.uib.no
Sent: Monday, May 11, 2009 6:56 AM
To: hash-function@nist.gov
Cc: hash-forum@nist.gov
Subject: OFFICIAL COMMENT: CHI

Follow Up Flag: Follow up
Flag Status: Red

Dear all,

We (myself, Jean-Philippe Aumasson, Willi Meier and Florian Mendel) have made an interesting observation on the PRE-MIXING step of CHI-256 and CHI-224.

Our observation can be used to obtain pseudo-second preimages and pseudo-collisions for CHI-256/224 with negligible effort. It applies to arbitrary chaining values and messages, and for an arbitrary number of rounds of the step function. However, we have not been able to extend our findings to attack the full hash function - this is an attack on the compression function only.

In CHI-256/224, the state of the round function consists of six 64-bit words, (A, B, C, D, E, F). The compression function is an unbalanced Feistel network which is clocked for 20 rounds; in each round, the words A, B, D and E are combined nonlinearly with two expanded message words to produce new values AA and DD, which are then xored with C and F. After one step, the updated state becomes (F ^ AA, A, B, C ^ DD, D, E), and so on.

Our observation is quite simple: in the first part of the step function (PRE-MIXING), the state words A, B, D and E are used to compute four temporary values preR, preS, preT and preU. The values A, B, D, E are not used again after this. Each of the temporary values depend on shuffled and rotated versions of exactly TWO state words.
Thus, if we flip all the bits of every state word (d = 0xFFFF FFFF FFFF FFFF), the differences in preR, preS, preT and preU are all 0.
All the subsequent (nonlinear) parts of the step function will remain inactive.

Hence, (d, d, d, d, d, d) is an iterative characteristic for the step function, and holds with probability 1. If this difference occurs in the chaining variable, it will persist for any number of rounds, and finally be cancelled by the (modified) Davies-Meyer feedforward.
Hence, given some arbitrary chaining value C and any message M, we have that CHI256-Compress(C, M) = CHI256-Compress(C', M), with C'
being the bitwise complement of C.

We emphasise that this weakness does not occur in CHI-512/384, because the PRE-MIXING step is computed differently. Furthermore, it does not say anything about the merits of the remaining parts of the
CHI-256/224 step function, since these are never activated. Finally, we have not been able to extend our result to attack the full hash, because the size of the chaining variable (384 bits) is sufficiently greater than the digest size.

Regardless, we believe that our result raises concerns about the security of the underlying "CHI-256/224 block cipher" implied by the specification, as well as the Merkle-Damgård security proof for the domain extender of CHI-256/224.

The CHI team has been notified, and has confirmed our findings.

Best regards,

Tor E. Bjørstad

--
Tor E. Bjørstad * PhD student, Dept. of Informatics, UiB, Norway
Email: tor.bjorstad@ii.uib.no (work) / torebj@gmail.com (private)
Phone: (+47) 97 08 77 22 (mobile) / (+47) 55 58 41 81 (office)
Web: http://www.ii.uib.no/~tor/ * Skype: tor.erling.bjorstad

From: hash-forum@NIST.GOV on behalf of Hawkes, Philip [phawkes@qualcomm.com]
Sent: Monday, May 25, 2009 5:33 AM
To: Multiple recipients of list
Subject: OFFICIAL COMMENT: CHI

Hi All,

This email is a response to the analysis of Tor E. Bjørstad et al that was submitted as an official comment recently. We are grateful to Tor, Jean-Philippe, Willi and Florian for their analysis. They graciously informed us before making a public comment in order to obtain our feedback.

We provide a summary of our response first. This summary is followed by a more detailed explanation.

Regards, Phil Hawkes (on behalf of the CHI team)

SUMMARY:

We concur that the differential exists. We agree that this is a bad weakness.

This attack highlights a problem in the PRE-MIXING of CHI-224/256 (which maps 4 inputs to 4 outputs). This mapping was intended to be one-to-one, but we neglected to notice that the current structure is not a one-to-one mapping. Thanks to the Bjørstad et al observation, we are now considering how to make the PRE-MIXING a one-to-one mapping. We plan to take some time considering the options. The attack does not involve any other part of the block cipher, and hence (we feel) the observation does not reflect on the security and soundness of the underlying CHI block cipher once the PRE-MIXING is made into a one-to-one mapping.

Also, is our opinion that the observation does not reflect on the security and soundness of the CHI domain extender. Note that with the differential obtained by the Bjørstad et al, the traditional Merkle-Damgård domain extender would result in the output hash states colliding. The CHI domain extender was intended as a minor change (to the traditional Merkle-Damgård domain extender) to make fixed points difficult to find. This property has not been violated by the Bjørstad et al attack.

Conclusion: we intend to fix the problem in the PRE-MIXING with a minor tweak, but we plan to leave the CHI domain extender as specified. We encourage further analysis of CHI, since the design principles have not been compromised.

DETAILS

NOTES ON THE FINDING.

We concur with their finding that a differential of probability exists where a difference in all bits of the input hash state results in the output hash state colliding when the corresponding messages are identical.

This attack highlights a bigger problem in the PRE-MIXING that we (the CHI team) should have identified in the design phase. We intended that the PRE-MIXING of CHI-224/256 (which maps 4 inputs to 4 outputs) would be a one-to-one mapping, but neglected to notice that combining two input bits for each output is not a one-to-one mapping. This is something that I knew (as a general principle) but I never thought about the impact on the PRE-MIXING we were designing - I feel embarrassed that I missed something so simple.

IMPLICATIONS OF THE ATTACK ON THE UNDERLYING BLOCK CIPHER

Bjørstad et al comment that their observation "... raises concerns about the security and soundness of the underlying ... block cipher" as currently specified. Our opinion is that the concerns are easily remedied.

The attack indicates that combining pairs of rotated values was a bad design choice for CHI. The attack uses this weakness to form a differential which avoids all the components providing diffusion and confusion (the rotations, the S-box and the addition operations). The attack does not reflect the strength of the components providing the diffusion and confusion.

We are considering an appropriate change to the PRE-MIXING to make the PRE-MIXING one-to-one. This will prevent the Bjørstad et al attack. We intend to only consider options that have minimal effect on other existing analysis of CHI. We encourage ongoing analysis of current CHI since the results will likely still apply to the fixed CHI.

Our opinion is that, after we make appropriate changes to make the PRE-MIXING one-to-one, we will have addressed the Bjørstad et al concerns about the security and soundness of the underlying block cipher.

IMPLICATIONS OF THE ATTACK ON THE CHI DOMAIN EXTENDER

The attack results in a collision in the output hash state. This relies on final XOR operation cancelling the two sets of differences (a) the differences in the output of the CHI block cipher and (b) the differences in the input hash state after the rotations are applied.

Bjørstad et al comment that their observations "... raises concerns about the security and soundness of the ... domain extender ".
We firstly note that the CHI domain extender was intended as a minimal change to the traditional Merkle-Damgård domain extender for the purpose of making fixed points difficult to find. This property has not been violated by the attack.

We also note that using the traditional Merkle-Damgård domain extender would still result in the output hash states colliding. In modifying the traditional Merkle-Damgård domain extender, we considered feeding back a modified version of the input hash state, where the modification would not involve input from the message.

For any simple modification, there are choices of input hash state differences and block cipher output differences such that the input hash input cancels with the differences in the output from the block cipher If we assume that the input and output differences for an optimal differential through the block cipher are reasonably independent, then each choice for modifying the input hash state prior to feedback would be equally likely to result in the modified input hash state and block cipher output cancelling each others' differences. That is, any simple modification chosen was just as likely to be susceptible result in a collision in the output hash function.

In the case of CHI, we were "unlucky" in that the differential through the block cipher has input hash state differences block cipher output differences that cancel. There are modifications for which the differences would not have cancelled. However, once we remedy the problem with the PRE-MIXING, there is no reason to suspect that the current modification would be any worse than other modifications. We could apply a more complex modification to the input hash state before the feedback, but the advantages appear limited.

Consequently, we are content to continue using the current CHI domain extender.

CONCLUSION

We intend to fix the problem in the PRE-MIXING with a minor tweak that has minimal effect on other existing analysis of CHI. We encourage further analysis of CHI, since this attack does not reflect the potential strength of CHI.

We plan to leave the CHI domain extender as specified.

From: Hawkes, Philip [phawkes@qualcomm.com]
Sent: Tuesday, September 01, 2009 5:41 AM
To: hash-function@nist.gov
Cc: hash-forum@nist.gov
Subject: OFFICIAL COMMENT: CHI

Attachments: chi_tweak_short_20090817.pdf

chi_tweak_short_2 0090817.pdf (...

```
Attached is a description of the tweak for CHI. I know CHI is no longer in the
race, but I hope there is still *some* value in making the tweak public.

Best regards,
Phil Hawkes (on behalf of the CHI team)
```

Tweak to the CHI Submission to the SHA-3 Competition

Design Team: Cryptographic Hash Initiative, Qualcomm International

Phil Hawkes, Cameron McDonald, Harry Wiggins
phawkes@qualcomm.com, cameronm@qualcomm.com, hwiggins@qualcomm.com
Level 3, 77 King Street, Sydney NSW 200, Australia

August 17, 2009

Abstract

This document introduces a simple tweak to the Little CHI algorithm (CHI-224 and CHI-256) to fix a problem the *PRE-MIXING* phase that was noted by Bjørstad et al [1].

1 Summary

A weakness has been identified in Little CHI (CHI-224 and CHI-256) of the original submission. To prevent this weakness, Little CHI is to be tweaked - the resulting algorithm is Little CHI-v2. The only difference between Little CHI and Little CHIv2 is that the computation of $preR, preS, preT, preU$ is changed to

$$\begin{aligned}
preR &:= A \oplus DROTR32^{8,8}(B) \oplus DROTR32^{5,1}(SWAP32(D));\\
preS &:= DROTR32^{17,12}(A) \oplus DROTR32^{14,22}(D) \oplus DROTR32^{2,23}(SWAP32(E));\\
preT &:= DROTR32^{18,17}(SWAP32(D));\\
preU &:= DROTR32^{7,26}(SWAP32(A)).
\end{aligned}$$

The rotation amounts applied to each state variable have not changed - only the combinations of rotated words input to has been altered.

2 Background

The CHI algorithms are considered in pairs: Little CHI is the algorithm for CHI-224 and CHI-256; while Big CHI is the algorithm for CHI-384 and CHI-512. No weaknesses of Big CHI have been identified and the specification remains unchanged by this tweak. The domain extender is also not changed by our tweak.

The compression function uses a block cipher in a modified Davies-Meyer mode. The round functions used in the block cipher has five phases: *PRE-MIXING*, *DATA-INPUT*, *NONLINEARITY*, *POST-MIXING* and *FEEDBACK*. Tor E. Bjørstad et al [1] published an attack on Little CHI that exploits a property of the *PRE-MIXING* of Little CHI. This *PRE-MIXING* is a mapping from four 64-bit words to four 64-bit words, and should be invertible. However, the mapping is not-invertible.

To solve this problem, we propose a minimal tweak to which makes the *PRE-MIXING* invertible, thus eliminating the attack. The rotations applied to the inputs are not changed - only the combinations of rotated words are changed. This means that existing analysis of the rotation amounts still applies to the tweaked version of Little CHI. The implementation performance is not expected to be negatively affected.

Note: at the time of publishing, the CHI team was already aware that CHI had not proceeded to Round 2 of the SHA-3 competition. This fix is provided primarily for those with an academic interest in CHI.

3 The Problem to Be Fixed

The original specification for the computation of $preR, preS, preT, preU$ is as follows:

$$\begin{aligned}
preR &:= DROTR32^{8,8}(B) \oplus DROTR32^{5,1}(SWAP32(D));\\
preS &:= A \oplus DROTR32^{18,17}(SWAP32(D));\\
preT &:= DROTR32^{7,26}(SWAP32(A)) \oplus DROTR32^{14,22}(D);\\
preU &:= DROTR32^{17,12}(A) \oplus DROTR32^{2,23}(SWAP32(E)).
\end{aligned}$$

Bjørstad et al [1] made the following observation:

```
...for some arbitrary chaining value (A,B,C,D,E,F), we flip ALL the state
bits (xor difference 0xFFFFFFFFFFFFFFFF). Because each of the four values
preR, preS, preT and preU which are computed in the pre-mixing step depend
on (rotated and shuffled versions of) exactly TWO chaining variables, this
difference will cancel and remain 0 throughout the step function.

Because of the Feistel structure of CHI, the difference will remain unaffected
in (A,B,C,D,E,F) for any number of steps, and is canceled at the very end
of the compression function by the (modified) Davies- Meyer feedforward.  Hence,
given any (M, C), we trivially obtain (M,C') such that CHICompress(M, C) =
CHICompress (M, C').
```

This observation led the submitters to notice that the *PRE-MIXING* is not invertible. The strength of the round function relies on the *PRE-MIXING* being invertible. Making the *PRE-MIXING* invertible would not only prevent the Bjørstad et al attack, but many other potential problems.

4 The Limitations

The primary role of the *PRE-MIXING* is diffusion. Other submissions use Galois field matrix multiplication for diffusion, which is a good approach due to links with coding theory and the flexibility in combining all input to compute an output. The original CHI algorithms have sought to use rotations as the basis for diffusion for simplicity sake. The submitters decided to continue this approach to keep in with the CHI ”philosophy”.

In the design of CHI, significant effort had been invested in choosing good rotation amounts for the *PRE-MIXING* in combination with the rotation amounts for the *POST-MIXING* and *DATA INPUT*. The submitters were loath to waste this investment. Furthermore, any changes to the rotation amounts would likely render any outside analysis irrelevant. Hence, the submitters were reluctant to change the rotation amounts.

Taking these limitations into account, the solution should use the same rotations applied to the same state variables. One approach would be have a two copies of the same rotated word contributing to two outputs (that is, the same word rotated the same amount). However, this would allow a mechanism for introducing two-bit differences to the same 4-in 3-out S-box. The CHI S-box was designed so that a single bit difference in the inputs always results in at least one bit difference in the output. This property does not hold when the inputs have differences in two bits. Consequently, the submitters decided not to have two copies of the same rotated word contributing to two outputs.

5 The Tweak

This is the new specification for the computation of $preR, preS, preT, preU$ for Little CHIv2:

$$\begin{aligned}
preR &:= A \oplus DROTR32^{8,8}(B) \oplus DROTR32^{5,1}(SWAP32(D));\\
preS &:= DROTR32^{17,12}(A) \oplus DROTR32^{14,22}(D) \oplus DROTR32^{2,23}(SWAP32(E));\\
preT &:= DROTR32^{18,17}(SWAP32(D));\\
preU &:= DROTR32^{7,26}(SWAP32(A)).
\end{aligned}$$

This operation is invertible (one-to-one), and uses the same rotation amounts applied to state words as for the original Little CHI.

References

[1] Tor E. Bjørstad, Jean-Philippe Aumasson, Willi Meier, and Florian Mendel. Official comment: Chi. NIST's hash-forum email list: hash-forum@nist.gov, May 2009. See `http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/documents/CHI_Comments.pdf`.