Bit attacks

D. J. Bernstein
University of Illinois at Chicago

```
From: andr...@ise...
Date: 11 Feb 2009 14:48
Subject: Question

Running CubeHash8/1 with 64
bit output over 2 different
datasets give me the same
hash under Visual Studio.
Using the code from simple.c
and call it the following
way:
```

acks

Bernstein

sity of Illinois at Chicago

From: andr...@ise...

Date: 11 Feb 2009 14:48

Subject: Question

Running CubeHash8/1 with 64 bit output over 2 different datasets give me the same hash under Visual Studio. Using the code from simple.c and call it the following way:

```
memcpy
"AAAAA
,16);
Hash(6
for(i
printf
printf

memcpy
"AAAAA
,16);
Hash(6
for(i
printf
printf
```

ois at Chicago

From: andr...@ise...
Date: 11 Feb 2009 14:48
Subject: Question

Running CubeHash8/1 with 64 bit output over 2 different datasets give me the same hash under Visual Studio. Using the code from simple.c and call it the following way:

```
memcpy(data,
"AAAAAAAABBBB
,16);
Hash(64,data,
for(i = 0; i
printf("%02x"
printf("\n");

memcpy(data,
"AAAAAAAACBBB
,16);
Hash(64,data,
for(i = 0; i
printf("%02x"
printf("\n");
```

cago

From: andr...@ise...
Date: 11 Feb 2009 14:48
Subject: Question

Running CubeHash8/1 with 64 bit output over 2 different datasets give me the same hash under Visual Studio. Using the code from simple.c and call it the following way:

```
memcpy(data,
"AAAAAAAABBBB\0\0\0\0
,16);
Hash(64,data,16,hash)
for(i = 0; i < 8; i++
printf("%02x",0xff&ha
printf("\n");

memcpy(data,
"AAAAAAAACBBB\0\0\0\0
,16);
Hash(64,data,16,hash)
for(i = 0; i < 8; i++
printf("%02x",0xff&ha
printf("\n");
```

From: andr...@ise...
Date: 11 Feb 2009 14:48
Subject: Question

Running CubeHash8/1 with 64 bit output over 2 different datasets give me the same hash under Visual Studio. Using the code from simple.c and call it the following way:

```
memcpy(data,
"AAAAAAAABBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");

memcpy(data,
"AAAAAAAACBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");
```

andr...@ise...
11 Feb 2009 14:48
ct: Question

ng CubeHash8/1 with 64
utput over 2 different
ets give me the same
under Visual Studio.
the code from simple.c
all it the following

```
memcpy(data,
"AAAAAAAABBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");

memcpy(data,
"AAAAAAAACBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");
```

As you
minor
datase
with a
produc

379ec8
379ec8

Is this
of the

@ise...
2009 14:48
tion

ash8/1 with 64
er 2 different
me the same
sual Studio.
e from simple.c
he following

```
memcpy(data,
"AAAAAAAABBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");

memcpy(data,
"AAAAAAAACBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");
```

As you can see
minor differen
dataset (first
with a "C". Ru
produces:

```
379ec80069d7a7
379ec80069d7a7
```

Is this the winner
of the final Cube

48

ith 64
ferent
same
dio.
imple.c
wing

```
memcpy(data,
"AAAAAAAABBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");

memcpy(data,
"AAAAAAAACBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");
```

As you can see, there
minor difference in th
dataset (first "B" rep
with a "C". Running it
produces:

```
379ec80069d7a71b
379ec80069d7a71b
```

Is this the winner
of the final CubeHash prize

```
memcpy(data,
"AAAAAAAABBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");

memcpy(data,
"AAAAAAAACBBB\0\0\0\0"
,16);
Hash(64,data,16,hash);
for(i = 0; i < 8; i++)
printf("%02x",0xff&hash[i]);
printf("\n");
```

As you can see, there is a minor difference in the dataset (first "B" replaced with a "C". Running it produces:

```
379ec80069d7a71b
379ec80069d7a71b
```

Is this the winner
of the final CubeHash prize?

```
y(data,
AAAABBBB\0\0\0\0"

64,data,16,hash);
 = 0; i < 8; i++)
f("%02x",0xff&hash[i]);
f("\n");

y(data,
AAAACBBB\0\0\0\0"

64,data,16,hash);
 = 0; i < 8; i++)
f("%02x",0xff&hash[i]);
f("\n");
```

As you can see, there is a minor difference in the dataset (first "B" replaced with a "C". Running it produces:

```
379ec80069d7a71b
379ec80069d7a71b
```

Is this the winner
of the final CubeHash prize?

Let's lo

Progra
a string

Classic
“input
Okay:

```
\0\0\0\0"

16,hash);
< 8; i++)
,0xff&hash[i]);



\0\0\0\0"

16,hash);
< 8; i++)
,0xff&hash[i]);
```

As you can see, there is a minor difference in the dataset (first "B" replaced with a "C". Running it produces:

```
379ec80069d7a71b
379ec80069d7a71b
```

Is this the winner
of the final CubeHash prize?

Let's look at wha

Programmer wan
a string `s` with n

Classic MD5 API
“`input` has inpu

Okay: `input` = s
`inputlen`

```
"


;
)
sh[i]);




"


;
)
sh[i]);
```

```
As you can see, there is a
minor difference in the
dataset (first "B" replaced
with a "C". Running it
produces:

379ec80069d7a71b
379ec80069d7a71b
```

Is this the winner
of the final CubeHash prize?

Let's look at what happene

Programmer wants to hash
a string `s` with `n` bytes.

Classic MD5 API:
"`input` has `inputlen` byte

Okay: `input` = `s`;
`inputlen` = `n`

As you can see, there is a minor difference in the dataset (first "B" replaced with a "C". Running it produces:

```
379ec80069d7a71b
379ec80069d7a71b
```

Is this the winner
of the final CubeHash prize?

Let's look at what happened.

Programmer wants to hash
a string `s` with `n` bytes.

Classic MD5 API:
"`input` has `inputlen` bytes."
Okay: $\texttt{input} = \texttt{s}$;
$\texttt{inputlen} = \texttt{n}$

As you can see, there is a minor difference in the dataset (first "B" replaced with a "C". Running it produces:

```
379ec80069d7a71b
379ec80069d7a71b
```

Is this the winner
of the final CubeHash prize?

Let's look at what happened.

Programmer wants to hash
a string $\mathtt{s}$ with $\mathtt{n}$ bytes.

Classic MD5 API:
“$\mathtt{input}$ has $\mathtt{inputlen}$ bytes.”
Okay: $\mathtt{input} = \mathtt{s}$;
$\quad\quad\mathtt{inputlen} = \mathtt{n}$

NIST SHA-3 API:
“$\mathtt{data}$ has $\mathtt{databitlen}$ bits.”
Okay: $\mathtt{data} = \mathtt{s}$;
$\quad\quad\mathtt{databitlen} = 8 * \mathtt{n}$

```
u can see, there is a
 difference in the
et (first "B" replaced
a "C". Running it
ces:

30069d7a71b
30069d7a71b
```

---

the winner
final CubeHash prize?

Let's look at what happened.

Programmer wants to hash
a string $\texttt{s}$ with $\texttt{n}$ bytes.

Classic MD5 API:
"$\texttt{input}$ has $\texttt{inputlen}$ bytes."
Okay: $\texttt{input} = \texttt{s}$;
$\texttt{inputlen} = \texttt{n}$

NIST SHA-3 API:
"$\texttt{data}$ has $\texttt{databitlen}$ bits."
Okay: $\texttt{data} = \texttt{s}$;
$\texttt{databitlen} = 8 * \texttt{n}$

e.g. da
to hash

AAAAA
AAAAA

e, there is a
nce in the
t "B" replaced
unning it

71b

71b

---

r
Hash prize?

Let's look at what happened.

Programmer wants to hash a string s with n bytes.

Classic MD5 API:
"input has inputlen bytes."
Okay: input = s;
inputlen = n

NIST SHA-3 API:
"data has databitlen bits."
Okay: data = s;
databitlen = $8 * \texttt{n}$

e.g. databitler
to hash 16 bytes

AAAAAAAABBBBO
AAAAAAAACBBBO

is a
he
placed
t

e?

Let's look at what happened.

Programmer wants to hash
a string s with n bytes.

Classic MD5 API:
"input has inputlen bytes."
Okay: input = s;
inputlen = n

NIST SHA-3 API:
"data has databitlen bits."
Okay: data = s;
databitlen = 8 * n

e.g. databitlen = 128
to hash 16 bytes:

AAAAAAAABBBB0000
AAAAAAAACBBB0000

Let's look at what happened.

Programmer wants to hash
a string s with n bytes.

Classic MD5 API:
"input has inputlen bytes."
Okay: input = s;
inputlen = n

NIST SHA-3 API:
"data has databitlen bits."
Okay: data = s;
databitlen = $8 * \texttt{n}$

e.g. databitlen = 128
to hash 16 bytes:

AAAAAAAABBBB0000
AAAAAAAACBBB0000

Let's look at what happened.

Programmer wants to hash
a string s with n bytes.

Classic MD5 API:
"input has inputlen bytes."
Okay: input = s;
inputlen = n

NIST SHA-3 API:
"data has databitlen bits."
Okay: data = s;
databitlen = 8 * n

e.g. databitlen = 128
to hash 16 bytes:

AAAAAAAABBBB0000
AAAAAAAACBBB0000

What if the programmer
forgets to multiply by 8?

databitlen = 16:

AA	AAAAAABBBB0000
AA	AAAAAACBBB0000

ook at what happened.

mmer wants to hash

g `s` with `n` bytes.

MD5 API:

t has `inputlen` bytes.”

`input` = `s`;

`inputlen` = `n`

SHA-3 API:

has `databitlen` bits.”

`data` = `s`;

`databitlen` = 8 * `n`

e.g. `databitlen` = 128

to hash 16 bytes:

```
AAAAAAAABBBB0000
AAAAAAAACBBB0000
```

What if the programmer forgets to multiply by 8?

`databitlen` = 16:

```
AA AAAAAABBBB0000
AA AAAAAACBBB0000
```

```
From:
Date:
Subjec

Respo
here.
was my
with t
datale
numbe
```

at happened.

nts to hash
bytes.

l:

utlen bytes.”

s;

. = n

l:

bitlen bits.”

;

len = 8 * n

e.g. databitlen = 128
to hash 16 bytes:

AAAAAAAABBBB0000
AAAAAAAACBBB0000

What if the programmer forgets to multiply by 8?

databitlen = 16:

AA	AAAAAABBBB0000
AA	AAAAAACBBB0000

From: andr...(
Date: 11 Feb 2
Subject: RE: (

Responding to
here. Found th
was my mistake
with the numbe
datalength, in
number of bits

ed.

n

es.”

ts.”

n

e.g. databitlen = 128
to hash 16 bytes:

AAAAAAAABBBB0000
AAAAAAAACBBB0000

What if the programmer forgets to multiply by 8?

databitlen = 16:

AA	AAAAAABBBB0000
AA	AAAAAACBBB0000

From: andr...@ise...
Date: 11 Feb 2009 15:
Subject: RE: Question

Responding to my own
here. Found the bug a
was my mistake. I cal
with the number of by
datalength, instead o
number of bits.

e.g. `databitlen` = 128
to hash 16 bytes:

```
AAAAAAAABBBB0000
AAAAAAAACBBB0000
```

What if the programmer forgets to multiply by 8?

`databitlen` = 16:

```
AA AAAAAABBBB0000
AA AAAAAACBBB0000
```

```
From: andr...@ise...
Date: 11 Feb 2009 15:40
Subject: RE: Question

Responding to my own message
here. Found the bug and it
was my mistake. I call Hash
with the number of bytes for
datalength, instead of the
number of bits.
```

atabitlen = 128

n 16 bytes:

AAABBBB0000
AAACBBB0000

f the programmer

to multiply by 8?

itlen = 16:

AAAABBBB0000

AAAACBBB0000

```
From: andr...@ise...
Date: 11 Feb 2009 15:40
Subject: RE: Question

Responding to my own message
here. Found the bug and it
was my mistake. I call Hash
with the number of bytes for
datalength, instead of the
number of bits.
```

What f

will for

Let's sa

Surely

> 100

Expect

of serv

forgett

Will th

interop

n = 128

:

000
000

rammer

ly by 8?

6:

0000

0000

```
From: andr...@ise...
Date: 11 Feb 2009 15:40
Subject: RE: Question

Responding to my own message
here. Found the bug and it
was my mistake. I call Hash
with the number of bytes for
datalength, instead of the
number of bits.
```

What fraction of

will forget to mu

Let's say fraction

Surely SHA-3 wi

$>$ 1000 network

Expect $>$ 1000/

of server program

forgetting to mul

Will this bug be

interoperability t

```
From: andr...@ise...
Date: 11 Feb 2009 15:40
Subject: RE: Question

Responding to my own message
here. Found the bug and it
was my mistake. I call Hash
with the number of bytes for
datalength, instead of the
number of bits.
```

What fraction of programm
will forget to multiply by 8
Let's say fraction is $1/F$.

Surely SHA-3 will be used
> 1000 network protocols.

Expect $> 1000/F$ cases
of server programmer
forgetting to multiply by 8.

Will this bug be caught by
interoperability tests?

```
From: andr...@ise...
Date: 11 Feb 2009 15:40
Subject: RE: Question

Responding to my own message
here. Found the bug and it
was my mistake. I call Hash
with the number of bytes for
datalength, instead of the
number of bits.
```

What fraction of programmers will forget to multiply by 8? Let's say fraction is $1/F$.

Surely SHA-3 will be used in > 1000 network protocols.

Expect $> 1000/F$ cases of server programmer forgetting to multiply by 8.

Will this bug be caught by interoperability tests?

```
 andr...@ise...
 11 Feb 2009 15:40
ct: RE: Question

nding to my own message
 Found the bug and it
y mistake. I call Hash
the number of bytes for
ength, instead of the
r of bits.
```

What fraction of programmers will forget to multiply by 8? Let's say fraction is $1/F$.

Surely SHA-3 will be used in > 1000 network protocols.

Expect $> 1000/F$ cases of server programmer forgetting to multiply by 8.

Will this bug be caught by interoperability tests?

Standa
require
client i

Still ex
of clien
indepen
forgett

@ise...
2009 15:40
Question

my own message
ne bug and it
e. I call Hash
er of bytes for
nstead of the
s.

What fraction of programmers will forget to multiply by 8? Let's say fraction is $1/F$.

Surely SHA-3 will be used in > 1000 network protocols.

Expect $> 1000/F$ cases of server programmer forgetting to multiply by 8.

Will this bug be caught by interoperability tests?

Standardizing a p
requires an indep
client implementa

Still expect > 10
of client program
independent serv
forgetting to mul

40

message
nd it
l Hash
tes for
f the

What fraction of programmers will forget to multiply by 8? Let's say fraction is $1/F$.

Surely SHA-3 will be used in > 1000 network protocols.

Expect $> 1000/F$ cases of server programmer forgetting to multiply by 8.

Will this bug be caught by interoperability tests?

Standardizing a protocol requires an independent client implementation.

Still expect $> 1000/F^2$ cas
of client programmer *and*
independent server progra
forgetting to multiply by 8

What fraction of programmers will forget to multiply by 8? Let's say fraction is $1/F$.

Surely SHA-3 will be used in > 1000 network protocols.

Expect $> 1000/F$ cases of server programmer forgetting to multiply by 8.

Will this bug be caught by interoperability tests?

Standardizing a protocol requires an independent client implementation.

Still expect $> 1000/F^2$ cases of client programmer *and* independent server programmer forgetting to multiply by 8.

What fraction of programmers will forget to multiply by 8? Let's say fraction is $1/F$.

Surely SHA-3 will be used in > 1000 network protocols.

Expect $> 1000/F$ cases of server programmer forgetting to multiply by 8.

Will this bug be caught by interoperability tests?

Standardizing a protocol requires an independent client implementation.

Still expect $> 1000/F^2$ cases of client programmer *and* independent server programmer forgetting to multiply by 8.

Typical tests will be passed. Protocol will be deployable. Last 7/8th of message will be trivially modifiable.

Security disaster!

fraction of programmers
get to multiply by 8?
ay fraction is $1/F$.

SHA-3 will be used in
0 network protocols.

$> 1000/F$ cases
er programmer
ing to multiply by 8.

is bug be caught by
erability tests?

Standardizing a protocol
requires an independent
client implementation.

Still expect $> 1000/F^2$ cases
of client programmer *and*
independent server programmer
forgetting to multiply by 8.

Typical tests will be passed.
Protocol will be deployable.
Last 7/8th of message
will be trivially modifiable.

Security disaster!

programmers
ltiply by 8?
is $1/F$.

ll be used in
protocols.

F cases
nmer
ltiply by 8.

caught by
ests?

Standardizing a protocol
requires an independent
client implementation.

Still expect $> 1000/F^2$ cases
of client programmer *and*
independent server programmer
forgetting to multiply by 8.

Typical tests will be passed.
Protocol will be deployable.
Last $7/8$th of message
will be trivially modifiable.

Security disaster!

ners
?

in

.

Standardizing a protocol requires an independent client implementation.

Still expect $> 1000/F^2$ cases of client programmer *and* independent server programmer forgetting to multiply by 8.

Typical tests will be passed.
Protocol will be deployable.
Last 7/8th of message
will be trivially modifiable.

Security disaster!

Standardizing a protocol requires an independent client implementation.

Still expect $> 1000/F^2$ cases of client programmer *and* independent server programmer forgetting to multiply by 8.

Typical tests will be passed.
Protocol will be deployable.
Last 7/8th of message will be trivially modifiable.

Security disaster!