

Re: hash function

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2005-02/4049.html>

leslie_hern78_at_hotmail.com

Date: 02/23/05

Date: 22 Feb 2005 17:54:50 -0800

David Wagner wrote:

> *Navin Kumar wrote:*

> > *To clarify, you want to obtain a construction for $h()$ satisfying the*

> > *properties: collision-resistant and preserves secrecy of x given b*
> *and*

> > *$h(x||b)$ and prevents finding a collision $h(x'||b) = h(x||b)$ given*
> *the*

> > *same b and $h(x||b)$?*

>

> *Ahh, is this the requirement? You want to avoid key-collisions?*

He seemed to have put up these two requirements in the very first post. It was you who had notational assumptions leading him to go astray.

> *I overlooked this last time. Ok, so you a simple PRF will not suffice,*

> *because the PRF security condition does not guarantee security against*

> *key-collisions. Conjecturally, SHA1-HMAC probably meets this additional*

> *"no key-collisions" condition -- but this is an unusual requirement, so*

> *it probably has not been studied as much.*

It appeared that he has given up the collision resistance requirement at some point.

>

> *Out of curiosity, what is the application that requires no key-collisions?*

>

> > *To ensure the suffix b doesn't affect the contribution x makes, a*

> > *trivial solution exists:*

> > *$h = h1(x) XOR h2(b)$*

>

> *This works, if the only criterion are the ones you described.*

sci.crypt: Re: hash function

- > *Indeed, if those are the only criteria you have, then something much simpler suffices. For instance, $h(x||b) = x$ will meet the two conditions above. So will $h(x||b) = h(x)$.*

Don't see how $h(x||b) = x$ could work. Could ye explain?

- >
- > *However, I'm worried that those are not really the only criteria.*
- > *I frequently see protocol designers who are not very clear on what security properties they need from their primitives. For instance,*
- > *one common thing is to hear someone say that they are looking for a hash for their protocol, and the hash needs to be is collision-resistant*
- > *and one-way. Sometimes this means that the requestor has proven their*
- > *protocol secure assuming only that these two conditions are met (so one*
- > *can be sure there are no other properties needed). However, all too often, this is not the case. More commonly, the implementor notices*
- > *that there is an attack if anyone can find collisions, so they figure that the hash had better be collision-resistant. That's fine as far*
- > *as it goes, but the fallacy is that defending against just the attacks*
- > *you know about is usually not good enough -- we also need to defend against attacks that might not have been anticipated. This confusion*
- > *is particularly likely when the protocol designer/implementor is not familiar with standard primitives and notions in cryptography, like PRFs, PRPs, random oracles, and the like.*

It sounds like having a proof can defend unanticipated attacks. Having security reduction to these primitives is better but still not guarantee un-anticipated attacks like the side-channel attack. Indeed, you might not guard against attacks due to adversary interactions not anticipated or specified in your reduction proof. In this sense, the defence of any provable secure protocols is also a bit ad hoc despite the better security against unanticipated attacks due to anticipated interaction.

- >
- > *Due to this frequent confusion, when I hear someone ask for a collision-resistant hash function without describing their application*
- > *in detail, I often try to suggest primitives that achieve more than what*
- > *they asked for, in case their applications needs more than just what they asked for. I have no idea whether that kind of reasoning applies*
- > *here, but it would be a good idea to ask.*

But you seem to neglect the requirement at first and give something achieving less at the first moment. Don't you feel ashamed to make

Re: hash function

sci.crypt: Re: hash function

assumption that everyone except you is so suck that you need to give them an over-designed solution? Dude, assume = ass-u-me!

>

> *For instance, with your construction, if I am given $h1(x)$ xor $h2(b)$ for*

> *any b , I can thereafter deduce the value $h1(x)$ xor $h2(b')$ for all other*

> *values b' as well. Is this a problem in Chan's application setting?*

> *I don't know, but I imagine it could well be.*