

Re: Checking a foolproof algorithm.

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2003-07/1806.html>

From: kurt wismer (kurtw_at_sympatico.ca)

Date: 07/26/03

Date: Sat, 26 Jul 2003 13:28:52 -0400

spymix wrote:

> *The scenario --*
>
> *I am an honest and trustworthy person and my friend is also an honest*
> *and trustworthy person.*
> *We have the same interest in encryption/decryption and programming.*
>
> *I believe I have discovered an algorithm along with the 2 prefixed*
> *keys that will create a foolproof coded message. I also created the*
> *decryption program.*
> *I send my friend the encrypted message along with the 2 prefixed keys.*
> *I also send him the same decrypted message.*
>
> *My friend by the way is an accomplished cryptologist and programmer.*
>
> *He studies the two (coded/plaintext) messages for as long as he needs*
> *to. He performs the entire test and they turn out inconclusive. He*
> *then attempts to write a decryption program to break the coded message*
> *but is having a lot of trouble finding out what the 2 key values do.*
> *Or at least that is what he thinks the problem is. He brings in other*
> *experts and they still have problems writing the decryption program.*
>
> *Given the information above, could this be a possible or impossible*
> *scenario?*

it's a possible situation, but it's not a very useful one...

> *If it is a possible scenario, then the encryption algorithm is*
> *foolproof.*

and this is not a valid conclusion...

you have made an error in your hypothetical situation... you've made your keys known to your attacker but not the algorithm... in any practical sense it's the converse of that which you need to be secure in... encryption/decryption algorithms cannot remain secret if they're used by the public, but keys can... and since the algorithm can't remain

sci.crypt: Re: Checking a foolproof algorithm.

secret, it can't be considered secure just because someone can't guess what it is while it still is secret...

breaking an encryption algorithm does not mean being able to reverse engineer the algorithm given all the inputs and outputs, it means being able to recover the plaintext (that which you encrypted so as to hide it from attackers) from the ciphertext (the encrypted data) in a usefully short amount of time... usually without being given the key(s)...

your example is like a door with the key hanging right on it, but with the lock hidden in an obscure and hard to reach area... it's security by obscurity...

--

```
"when surveys of all the world's countries are done,  
canada frequently rates number one.  
are we the best country? well we'll never know...  
there's nowhere else we can afford to go."
```