

Re: Minimal crypto OTP by dummie

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2004-04/1320.html>

From: Giorgio (giorgio_at_bignami.zzn.com)

Date: 04/20/04

Date: 20 Apr 2004 02:42:59 -0700

> Using random numbers without some sort of pre-processing, wouldn't that be
> risky? Wouldn't it be necessary to generate a new random sequence if the
> first one somehow reveals fragments of the message?

...

> If the 'enemy' already has some info, they can compare the existing info to
> the fragment and see if it makes 'sense'. If it does, they have yet another
> piece of info they can use to trace the sender and perhaps catch him.

I hope my explanation could be useful: xoring a message M with a private random key K of same length L, element by element, result in a ciphertext C where each element can be deciphered in any of the possible states S of the elements with the same probability (each bit can be deciphered with equal probabilities in 0 or 1, each byte can be deciphered with equal probability in any of the 256 possible byte's values).

So M can be deciphered with the same probability in any message of length L, S^L possible results. This is equal to state that, without knowing K, M is unrelated with C and vice versa, so C is not decipherable unless brute force attack trying S^L results (possibly $S^{(L-n)}$ where n are known bytes of M). However this brute force attack will be equal to create the M from nothing (or from the known bytes) and obviously noone can prevent a monkey to hit a billion times a keyboard and write Window's source code ;)

This is why randomness of K is important, another good reason is that knowing C and some bytes of M is equal to know the related bytes of K, since xor is an easily invertible function, so it's important that K is really random or the knowledge of some K bytes could be used to cast other K values and compromise the cryptogram.

Why shouldn't be obvious K sequences be discharged?

If you prevent the creation of those sequences or if you simply discard those results and make your RNG create other sequences unless they are good, all the previous schema is compromised since possible output of encryption are less than S^L .

Why? Because you have two ways to check if the sequence in K is "not good".

Firstly you can control if the value in $K[i]$ reveal $M[i]$ (in example $K[i]$ is 0), but in this way you have $(S-x)^L$ possible output state and

this establish a relation between M and C and this knowledge can be in some ways used by the attacker (i.e. I know "e" in M will never be "e" in C, that's less shure to not knowing non relations between C and M). So the security in non correlation between M and C is lost.

Second if you check if sequences in K are trivial (i.e. 1234567, or 111 or 000...) you establish a relationship between elements of K; i.e. very simply $K[i]$ must be $\neq K[i-1]$, or $K[i]-K[i-1]$ must be different from $K[i-1]-K[i-2]$, resulting in $(S^L)-x$ possible outputs, x related to the checking algorithm used.

So knowing some M elements you can know related K elements and try to cast other elements of K, loosing the second good feature of OTP, the non correlation between key's elements.

Those two kind of correlation can be not easy to use to attack the cryptogram, but obviously are weakenings compared to a totally unrelated system.

So "bad" K does'n weaken the OTP, not only they are practically very improbable, but most important the message will always be teorically unrelated with the key so "saddam is here!!!!" can be "Saddam is not here" with the same probability and the point is that since the probability of a revealing sequence in K is the same of casually create the same M sequence from nothing no C sequence can be ***anyway*** proved to result from a trivial K sequence, so simply bad sequences doesn't exist.

The only check to perform is that the RNG device is working properly, in this case IMHO is better to analyze operational parameters of the device, not the apparent quality of the output.