

Re: Real-time sound cyphering algorithm

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2005-05/1160.html>

From: David Eather (*eather_at_tpg.com.au*)

Date: 05/28/05

Date: Sat, 28 May 2005 14:28:21 +1000

Bryan Olson wrote:

> *Unruh wrote:*

> *Bryan Olson wrote:*

> >> *The way we would approach this today is to digitize the sound,
> >> compress (with an audio-specific coder), encrypt, code for
> >> forward error correction, and modulate onto an audio carrier.
> >> This group should be able to help with the encrypt step, and
> >> people here might know something about the others, but each of
> >> the steps is now reasonably well studied on its own.*

> >

> > *The problem is how to remove it from the audio carrier (played
> over a > loudspeaker with its largely random frequency dependent
> amplitude and phase
> changes) and still have it decoded to the original. Ie, you need
> huge error correction redundancy to hope to resurrect the original
> signal.*

>

> *Things have gotten much better.*

>

> *Trellis Coded Modulation (TCM), introduced by Gottfried
> Ungerboeck in 1982, is great for modulating (and demodulating)
> discrete signals over noisy analog lines. For error correction,
> Turbo Codes, introduced by Berrou, Glavieux, and Thitimajshima
> in 1993, are significantly more efficient than the error
> correction used in CD's, DVD's and HDTV. Turbo codes require
> some buffering; I think the stated half-second window should be
> fine.*

>

> *There are also various multi-media packetizing mechanisms
> designed to allow the stream to continue even while
> communication is lost for short periods. We'd probably want to
> use one, and make sure our crypto doesn't break it.*

>

>

> *I have now written most of what I know on the subject, and, I
> hope, not too much more.*

sci.crypt: Re: Real-time sound cyphering algorithm

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

I think this is the hard way to do this – a good way that will get a perfect result, but still a hard way. The OP wanted to decrypt and identify the sound by ear, so it doesn't have to be bit perfect.

Encode the sound stream by adding (not xoring) the stream cipher output. Decode is just receive the sound (normalise it first) and subtract the same stream cipher output. The result will probably have bit errors in most bytes but still have the identifiable sound.

(I skipped over the necessity of keeping both stream ciphers in sync – –If you can make sure there is no similar frequency in the ciphered sound stream I suggest periodically adding a few bytes of high frequency as sync bytes . Or if there is enough computational power superimpose the whole waveform on a low frequency carrier and use the zero crossings for sync – because of the errors in transmission this will take a bit of trial and error but you will know when you have the correct result – a FFT will show the result is no longer white noise).

-----BEGIN PGP SIGNATURE-----

Version: PGP 8.1

iQA+AwUBQpfzUJS9Fk5okqe7EQLqUACgmd+313eDHaU+gjuLWC5W8scXopsAmMor
t6ZgpyP8vWIwL11fcn0uomo=
=es3n

-----END PGP SIGNATURE-----