Synchronous entropy generators

Synchronous entropy generators allow you to use a simple and fast XOR encryption algorithm and not show the connection between the sender and recipient of messages to strangers.

To generate the same set of numbers (the key), application instances must use some source of information with a stream of numbers. Ideally, this is a special server with a digital noise flow based on the thermal noise of the diode. And the stream is transmitted without processing - “as is”.

If there is no such source, then you can use digital streams of Internet radio or Internet TV. Such flows are not random - this is a minus. But a lot of users connect to such servers and you will be “one of hundreds of thousands” there. If the radio is a conversational type or a TV news channel (for example, CNN), then the stream will be satisfactory and numbers must be selected from it over a large interval and processed using non-reversible methods.

An Android Java project demonstrates the functionality of the method.

https://github.com/vallshmeleff/radiorandgentwo

How it's done. Some sequence of numbers N is given, which the application brings to life in the digital stream of Internet radio. The length of this sequence must be chosen so that matches occur, for example, once per hour. Then the applications of the message sender and the recipient synchronize the moment the key is received from the stream, without interacting with each other in any way.

When the sequence is found, the application, using some algorithm, begins to select numbers from the stream. As much as needed for XOR encryption.

The key (set of numbers) will be the same for all instances of the application. To ensure reliable decryption, you can write down several keys in a row.

This application works. If the source code is used in an Android Studio project, the application will listen to the Internet radio stream, select a key, encrypt the text from a variable, send it via SMS to itself, receive the SMS, decode the text.

Using an external thread to generate identical keys for multiple independent clients is effective, but inconvenient. The same level of "randomness" of the encryption key can be achieved without an external digital stream (unless it is a stream from a hardware generator).

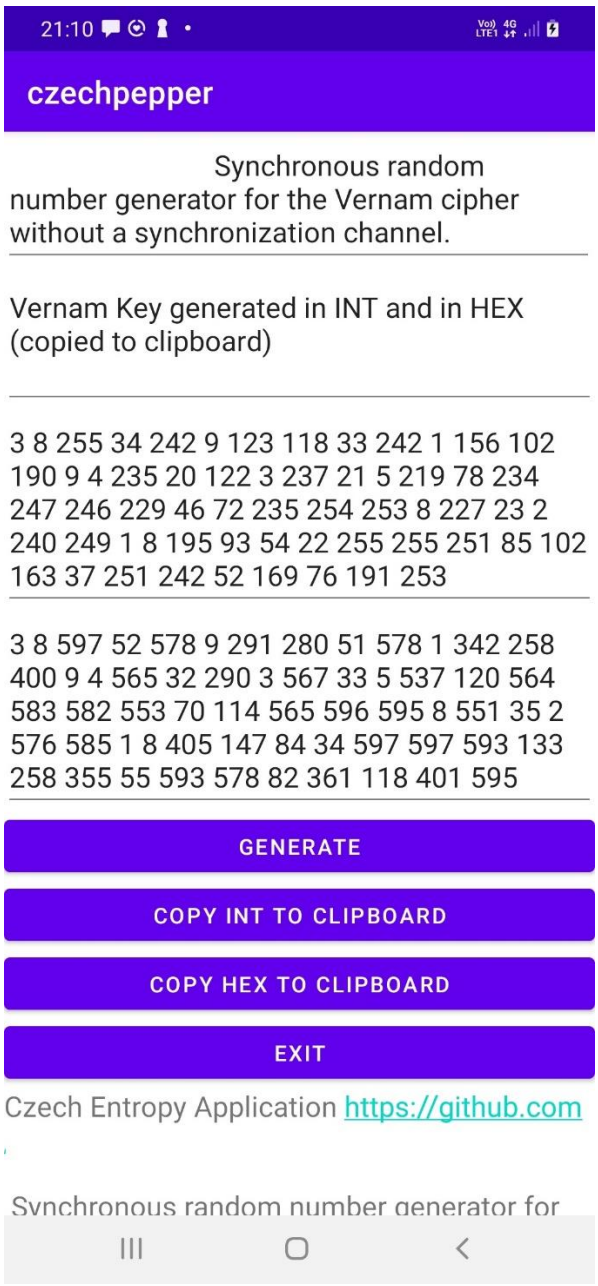
Czech Entropy

This application prototype takes long text as input (120-150 characters in the example), expands it by about 30,000 times and selects a “random” set of numbers. A formulaless non-reversible algorithm is used.

As text to start the application, the sender and recipient use, for example, the same top news story published by, for example, CNN. For example, at 12.00 Or the first two sentences from page 140 of the first book "The Lord of the Rings".

https://rescuewebcam.blogspot.com/

Application prototype:



The application DOES NOT ENCRYPTED data. The application only generates the same set of 60 numbers when you enter the same text.

For the application to work correctly, the phones must be identical and have the same build of the operating system.

The process of generating "random" numbers is very simple. There are many interesting technologies to upgrade it to a solid commercial level. For example, the algorithm and the number of used algorithms for generating “random” numbers can be changed dynamically during the operation of the application by the sender and the recipient, and they will not interact in any way. Those. Secretly. Hidden

Czech Entropy