Assignment 2 - One-Time-Pad

<Simon> <Rommer>, <1225253>

November 18, 2016

You are visiting Bikini Bottom and staying with your friend Spongebob. Nowadays, Spongebob and Patrick are secret agents and practice excessively to encrypt and decrypt messages. For you it is deadly dull in Bikini Bottom, because they do it all day. Since you have some background in crypto, you become interested in what they use for encryption. Without Spongebob being aware of it (which wasn't so difficult because it is Spongebob...), you copy 10 encrypted messages. You could also cast a glance at some code that was only partly hidden at the desk.

You immediately get the suspicion that SpongeBob and Patrick could be using the "unbreakable" One-Time-Pad. But, you suspect a serious flaw in their execution and decide to break the next secret message (target ciphertext) and confront your friends with the result.

Please document your findings and your solution by filling out this template and upload the resulting PDF to TUWEL.

1 One-Time-Pad

Assignment

Downloaded the ten ciphertexts and one target ciphertext from the website linked to in TUWEL. All the cyphertexts were encoded in HEX. This was ovious from the php-snipet Spongebob and Patrick used. The HEX-numbers also don't have leading "$0x$" es since according to the "*sprintf*" functione[1][2]. The goal was to break the encryption and obtain the original plaintexts.

One-Time-Pad Encryption

When encrypting with a One-Time-Pad you need a key to xor your message. The key hast to be at least as long as the message. If it's shorter, you could let the key repeat itself.

[1] `https://stackoverflow.com/questions/11070183/how-do-i-print-a-hexadecimal-number-with-leading-0-to-have-`
[2] Link is too long, but can still be clicked.

OTP's are virtually unbreakable if you use every key just once. Drawbacks are, that you have to securely transmit the key to the receiver (for example with Diffie–Hellman) and true random numbers for the key are hard to generate since the computers is per design a deterministic system, also there is no message authentication method.

Breaking One-Time-Pad

The cyphertexts can be decrypted since as soon the same key is used more than once and you have two or more messages with the same key, you can xor those messages and can get rid of the key this way. You now have a xor'ed text of the two messages where you can start crib dragging[3]. Crib dragging is basically guessing for common words to try on one message,and extend the guess on readable passages in the opposite message. For searching words I used an online service[4]. Also on how to guess smart , I've found good advice at stackexchange [5] and on Wikipedia[6]. Somehow I did not manage to guess the right words. Even though I've found some combinations that made sense and gave me specific words, it was not within my ability to be able to go on from that point. Therefor I regret that even after several days of trying the most common words and shifting them through the message pairs I was not able to find out about the article and decode the target message.

Results

Key Unfortunately I was not able to recover the key.

I wrote a little python script that gave me a list of xored messages in hex, here a link to my (rarely used) github account: `https://github.com/Acrasy/OTP-decode`

[3] `https://travisdazell.blogspot.co.at/2012/11/many-time-pad-attack-crib-drag.html`
[4] `http://www.morewords.com/`
[5] `https://crypto.stackexchange.com/questions/6020/many-time-pad-attack`
[6] `https://en.wikipedia.org/wiki/Most_common_words_in_English`