Software Development : Security : Cryptography : Onetime pad

Software Development : Security : Cryptography

Contents [hide]

The Proof

The proof that shows that the one-time pad is provably unbreakable is based on the analysis of the encryption of a single clear-text bit.

Given a table of

$XOR(cleartext,key) = cryptotext$

cleartext	key	cryptotext
0	1	1
0	0	0
1	1	0
1	0	1

it can be shown that if the probability of the key bit being 1 is exactly $1 \over 2$, then exactly 50% of all of the cryptotext bits equal 1, regardless of the cleartext bits. If the distribution of cryptotext bits does not depend on the clear-text bits, regardless of the length and value of the cleartext bit-stream, then the encryption algorithm is considered to be provably unbreakable.

One possible way to show that exactly 50% of cryptotext bits will equal 1 whenever exactly 50% of the key bits equals 1, regardless of the value of the cleartext bits, is to show that it holds for one 4-bit long key bitstream, for example, $1 1 0 0$, and then mention that due to the fact that the original contemplation, the one about the $1 1 0 0$, covered all possible 4-bit long cleartexts, the contemplation holds regardless of how the 4-bit long key bitstream is shuffled (key bitstream bits rearranged relative to eachother).

4 sequential cleartext bits can represent $2^4 = 16$ different, 4-bit long, bitstreams.

number of 1-s at cryptotext rows	8	8	8	8
key bitstream	1	1	0	0
----------	----	----	----	----
cleartext bitstream 1	1	1	1	1
cryptotext bitstream 1	0	0	1	1
----------	----	----	----	----
cleartext bitstream 2	1	1	1	0
cryptotext bitstream 2	0	0	1	0
----------	----	----	----	----
cleartext bitstream 3	1	1	0	1
cryptotext bitstream 3	0	0	0	1
----------	----	----	----	----

cleartext bitstream 4	1	1	0	0
cryptotext bitstream 4	0	0	0	0
----------	----	----	----	----
cleartext bitstream 5	1	0	1	1
cryptotext bitstream 5	0	1	1	1
----------	----	----	----	----
cleartext bitstream 6	1	0	1	0
cryptotext bitstream 6	0	1	1	0
----------	----	----	----	----
cleartext bitstream 7	1	0	0	1
cryptotext bitstream 7	0	1	0	1
----------	----	----	----	----
cleartext bitstream 8	1	0	0	0
cryptotext bitstream 8	0	1	0	0
----------	----	----	----	----
cleartext bitstream 9	0	1	1	1
cryptotext bitstream 9	1	0	1	1
----------	----	----	----	----
cleartext bitstream 10	0	1	1	0
cryptotext bitstream 10	1	0	1	0
----------	----	----	----	----
cleartext bitstream 11	0	1	0	1
cryptotext bitstream 11	1	0	0	1
----------	----	----	----	----
cleartext bitstream 12	0	1	0	0
cryptotext bitstream 12	1	0	0	0
----------	----	----	----	----
cleartext bitstream 13	0	0	1	1
cryptotext bitstream 13	1	1	1	1
----------	----	----	----	----
cleartext bitstream 14	0	0	1	0
cryptotext bitstream 14	1	1	1	0
----------	----	----	----	----
cleartext bitstream 15	0	0	0	1
cryptotext bitstream 15	1	1	0	1
----------	----	----	----	----
cleartext bitstream 16	0	0	0	0
cryptotext bitstream 16	1	1	0	0
----------	----	----	----	----

To be more explicit: the table covers all 4-bit long key-streams, where 50% of its bits equal 1, because the first row that contains the number 8 stays the same regardless of how the non-legend columns are ordered. The 8 is 50% of the 16.

References and Notes About the Proof

I(martin.vahi@softf1.com) learned the proof from the slides of Prof. Dr. Çetin Kaya Koç *(archival copy)*. As of 02.2013 I was not able to find the slides from his home page, i.e. I used a downloaded copy of the slides. His version of the proof is more intelligent than mine, but I often like proofs to be more explicit than mathematicians tend to prefer.

The one-time pad proof uses a version, where digits have only 2 values, the 0 and the 1, but it can be generalized to a version, where there can be 2 or more values. The general version has been used in practical applications at least as early as 2. World War, if not earlier.

Practical Attacks

The previously described proof only says that it is not possible to INFER THE ACTUAL clear-text from the cryptotext. The proof does not rule out a case, where multiple probabilistic clear-text candidates might be derived from the cryptotext. For example, if the probability of a clear-text candidate to match with the actual clear-text is about 80%

or 80% of the characters at the clear-text candidate match with the actual clear-text then that can still be very revealing. For example, if the clear-text candidate hints that a location of something interesting is "LondoF" and there is no place called "LondoF", but there is a place called "London", then from practical point of view the message can be considered to be cracked. The same with choices: if the clear-text candidates offer about 10 choices for names or places, then that number might be small enough for checking them all out in real life. That's why **it is important to apply proper salting before using the one-time-pad**.

Various Links

- TXOR