To appear in Proc. Information Hiding Wokrshop, Cambridge, UK, 30.5.-1.6.1996, LNCS, Springer-Verlag

Information Hiding Terminology

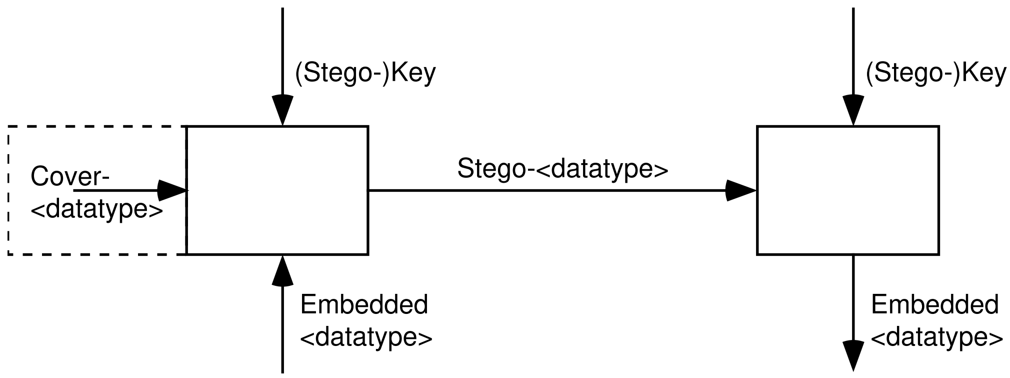
— Results of an informal plenary meeting and additional proposals —

Collected by Birgit Pfitzmann

The common opinion among the participants of this workshop was that all the topics that Ross Anderson had chosen, such as steganography and fingerprinting, had indeed much in common, and that we all profited by learning about the treatment of these topics in usually quite different communities and from different points of view. To facilitate this, we decided to agree on some common terminology in the final session of the workshop. I have tried to collect the results of this session, and added a few additional proposals.

The Terms Agreed Upon

We decided that most of the applications have the following common core:



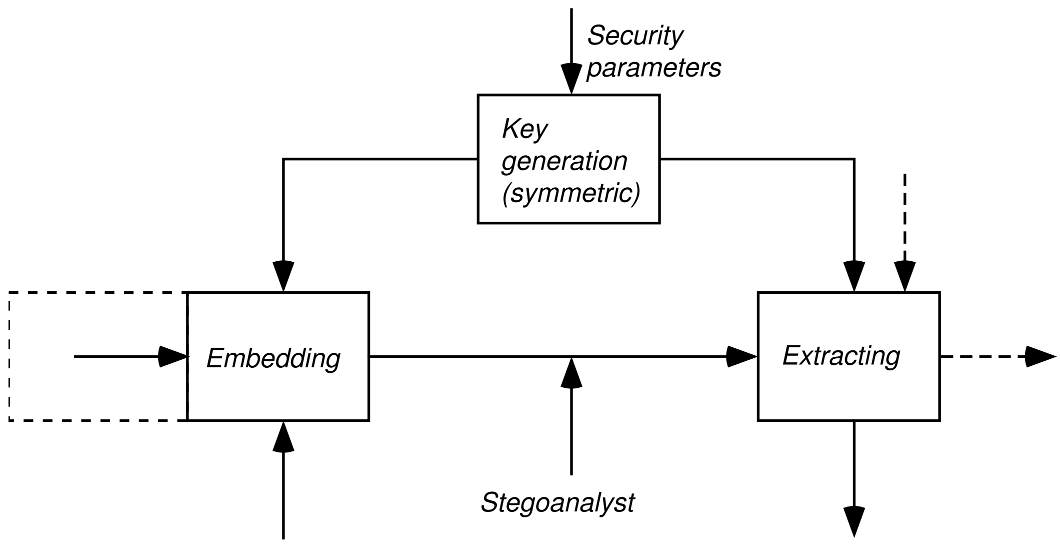
We decided upon the terms for the in- and outputs in the left-hand side of this diagram. We only fixed adjectives or qualifiers; everyone can prefix them to their favorite word for the datatype they use, i.e., <datatype> stands for “text”, “message”, “image”, or whatever seems appropriate. In the explanations, I use “message”.

1. Embedded <datatype>. Something to be hidden in something else.
2. Stego-<datatype>. The output of the hiding process; something that has the embedded message hidden in it.
3. Cover-<datatype>. An input with an “original” form of the stegomessage. In some applications, such a covermessage is given from the outside, in others, it can be chosen during the hiding process. The latter is represented by the dashed extension to the inner hiding process.
4. Stegokey or simply key. Additional secret data that may be needed in the hiding process. In particular, the same key (or a related one) is usually needed to extract the embedded message again.

There was no time to decide on names for further objects, but individual authors are strongly encouraged to introduce additional or deviating terminology by extending or modifying the figure above.

A Few Additional Proposals

I will venture to make a few more proposals, at least as a reference for other proposals:



A. The process of hiding the embedded message is called embedding.

B. Getting the embedded message out of the stegomessage again is called extracting.

C. The party from whom the embedded message is hidden is called the stegoanalyst.

D. The key has to be generated, often depending on one or more security parameters (e.g., one for the cryptographic security of a pseudorandom sequence and another for the collusion size). The standard case where the same key is used in embedding and extracting is called symmetric.

E. An entity or person that embeds and extracts is called an embeddor or an extractor, respectively.

Differences Between the Models

Noticing differences between models is often equally important as noticing similarities. Hence I also sketch some points where the various applications differ once we consider more than the core components. These differences are not all independent.

- Is there a given covermessage, and if yes, does it have to be used several times?
  For instance, in fingerprinting there is typically the original image or music to be sold. In contrast, in steganography, the covermessage can often be designed specifically for hiding the embedded message, or the stegomessage may be constructed immediately.
  If the same covermessage is used several times with different embedded messages, collusion attacks have to be considered. This is typical for fingerprinting.

- Does the covermessage play any role in extracting, too? In the second figure, two dashed arrows represent the cases where it is an input or an output, respectively.

  For instance, in fingerprinting, the covermessage can be an input in extracting, because the merchant is both the embeddor and the extractor. In steganography, the covermessage is usually neither an input nor an output, but some situations may be so restricted (e.g., war letters home) that the embeddor cannot afford to lose the legitimate bandwidth offered, and thus the covermessage must be extracted.

- What information should the stegoanalyst get legitimately?

  For instance, in fingerprinting, the stegoanalyst is the buyer and pays for the stegomessage. Thus the stegomessage has to be quite similar to the covermessage, even if the stegoanalyst does not know much about the covermessage and cannot notice the difference. Treaty verification and steganography have no such goal.

- What is the adversarial goal of the stegoanalyst:
  - to detect stegomessages;
  - to find out the embedded message;
  - to prove that a message has been embedded, or even what message, to third parties;
  - to remove the embedded message from the stegomessage without changing the stegomessage too much, e.g., retaining the similarity to the covermessage (as an illegal redistributor might) or even keeping the active interference unnoticed (as a censor might who wants to conceal his existence, or a telecommunication provider who claims to guarantee a transparent channel of a certain bandwidth);
  - or to jam, i.e., to remove all possible embedded messages without much care about the covermessage?

  The first three goals will usually be achieved by passive observation, the last two by active interference.

- What does the stegoanalyst know about the covermessage a priori? He might know it entirely, or a probability space that it is chosen from, or a family of probability spaces indexed by external events (e.g., that a letter “it is raining” is less likely in good weather), or some predicate about it. This knowledge will often be quite fuzzy.

More Examples

So far, the examples were mainly for fingerprinting and steganography. Here are a few more:

- Covert channels are similar to steganography, but the coverdata are the whole behaviour of a system. This fact also gives the stegoanalyst more freedom in being active, in particular in designing the probability space of the coverdata in the first place.

- Subliminal channels in authentication protocols are interesting in cases where the message to be authenticated is completely known to the stegoanalyst and the stegoanalyst may jam wrong messages. In these cases, the only available coverdata are the authentication data. Here the stegoanalyst has a precise criterion, valid authentication, and may change the stegodata as long as valid authentications remain valid.

- Traitor tracing for broadcast encryption is a kind of fingerprinting of a cryptographic key, i.e., the coverdata are a key needed to decrypt the ciphertexts that are broadcast. The stegodata must retain this property, and so must the data the stegoanalyst produces, i.e., the keys for pirate decoders.
- Avoiding traffic analysis means hiding locations or addresses. This is somewhat different from the other examples in that “location” is not a message. Some schemes do work by hiding messages in continuous dummy traffic, but even those are often multi-party protocols, i.e., many embeddors must send their “stegomessages” in a coordinated way. I propose not to use the terminology in this application.

One simple conclusion can also be drawn: If the covermessage can be chosen by the embeddor, and the stegoanalyst does not know it completely and is not allowed to jam, the embeddor and extractor can always defeat the stegoanalyst by encoding the embedded message within their legitimate choice among covermessages. Of course, this is not quite so clear if, in practice, none of the parties has a formal probabilistic model of the source. But it still shows that some steganography is possible whenever any normal communication is possible. More complicated statements are treated in the individual papers.