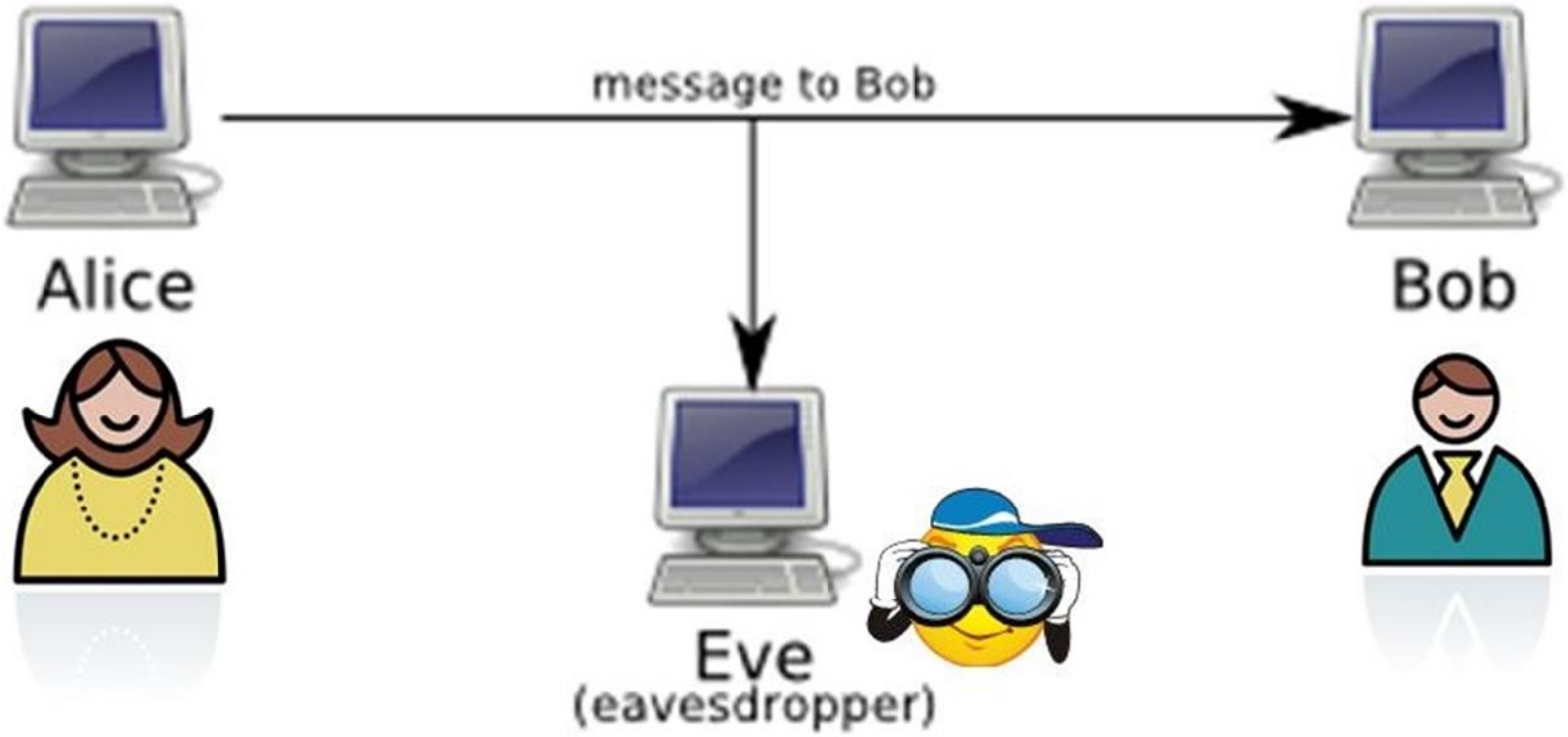
Steganographic channels

Ho Dac Hung

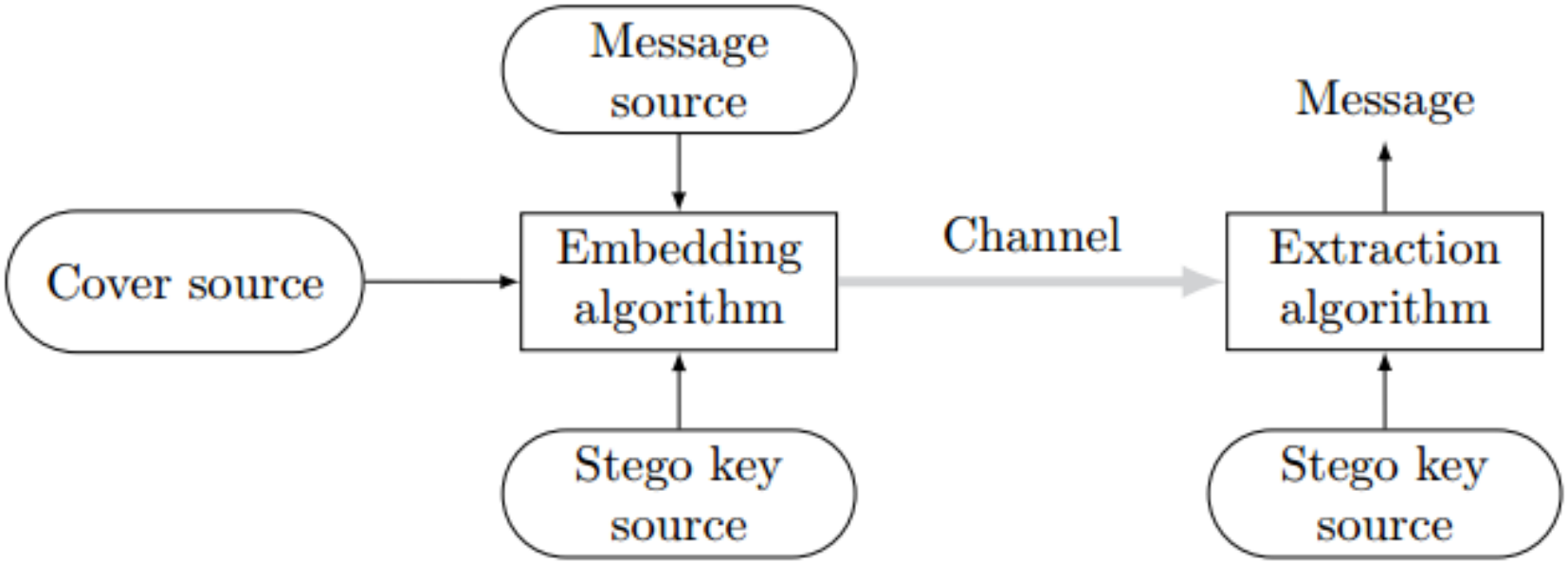
Contents

- Problem
- Steganography by cover selection
- Steganography by cover synthesis
- Steganography by cover modification

1. Problem



1. Problem


Message source
Message
Cover source
Embedding algorithm
Channel
Extraction algorithm
Stego key source
Stego key source

1. Problem

- Passive warden scenario
- Active warden scenario
- Malicious warden scenario

1. Problem

- The problem of steganography can thus be formulated as finding embedding and extraction algorithms for a given cover source that enable communication of reasonably large messages without introducing any embedding artifacts that could be detected by the warden. In other words, the goal is to embed secret messages undetectably.

2. Steganography by cover selection

- In steganography by cover selection, Alice has available a fixed database of images from which she selects one that communicates the desired message.

2. Steganography by cover selection

- The embedding algorithm can work simply by randomly drawing images from the database till an image is found that communicates the desired message.
- The stego key here is essentially the set of rules that tell Alice and Bob how to interpret the images.

2. Steganography by cover selection

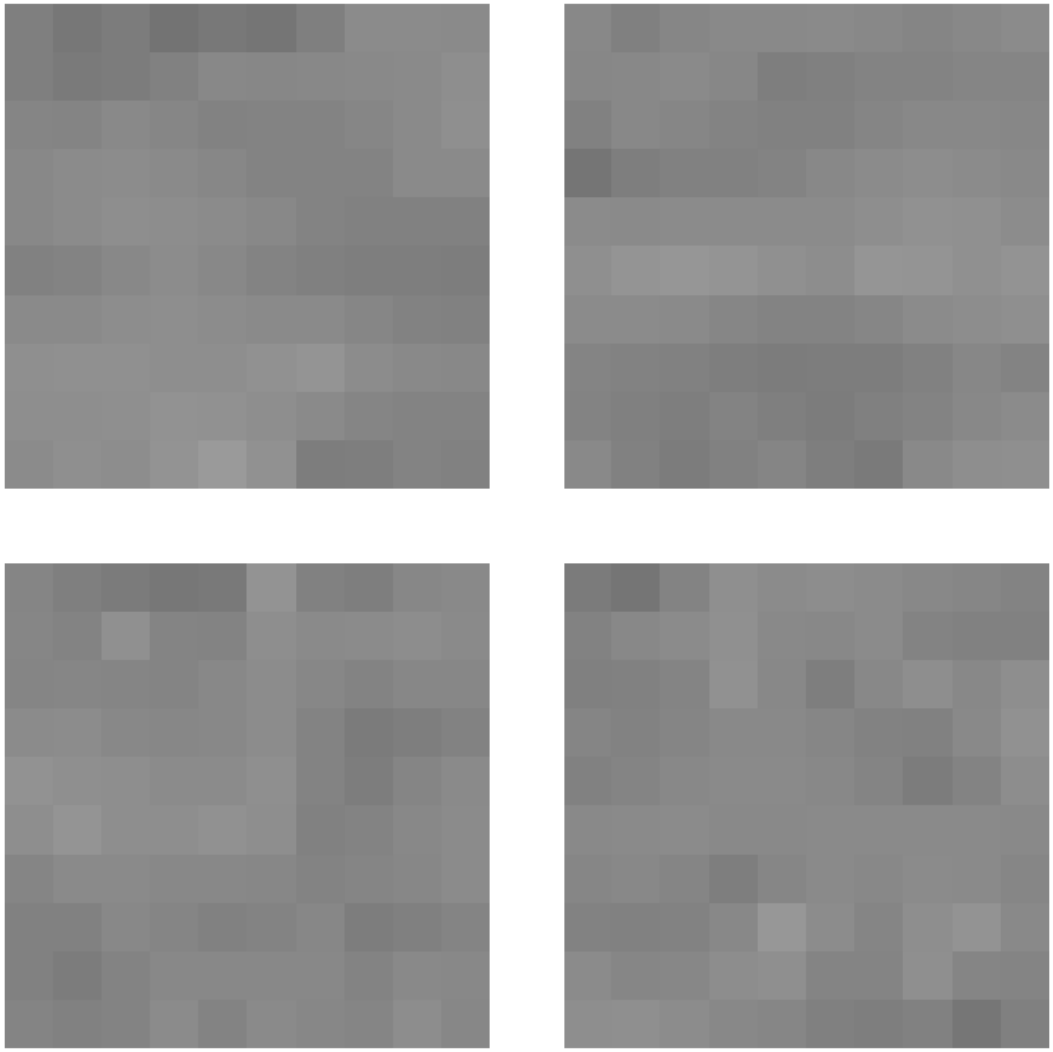
- An important case of steganography by cover selection involves message-digest (hash) functions.

$$h(x) = \{x[1] \bmod 2, x[2] \bmod 2, x[3] \bmod 2\}$$

3. Steganography by cover synthesis

- In steganography by cover synthesis, Alice creates the cover so that it conveys the desired message.
- Steganography by cover synthesis could be combined with steganography by cover selection to alleviate the exponential complexity of embedding by hashing.

3. Steganography by cover synthesis



3. Steganography by cover synthesis

- Let us assume that the images are 8-bit grayscale with $x_j[i]$ standing for the intensity of the ith pixel in the jth image, i = 1, . . . , n; j = 1, . . . , K.
- Alice will use a cryptographic hash function modified to return 4 bits when applied to 16 pixels.

3. Steganography by cover synthesis

- Alice divides every image into disjoint blocks of 4 × 4 pixels and assembles a new image in a block-by-block fashion so that each 4 × 4 block conveys 4 message bits.
- To embed the first 4 bits in the first 4 × 4 block of pixels, Alice searches through the hashes $h(x_j[1], \ldots, x_j[16])$, $j \in \{1, 2, \ldots, K\}$ till she finds a match between the hash of the first 16 pixels and the message, which will happen for image number j1.

3. Steganography by cover synthesis

- Then, she moves to the next block and finds j2 $\in \{1, \ldots, K\}$ so that $h(x_j[17], \ldots, x_j[32])$ matches the next 4 message bits, etc.
- The final stego image y will be a mosaic assembled from blocks from different images $y = (x_{j1}[1], \ldots, x_{j1}[16], x_{j2}[17], \ldots, x_{j2}[32], x_{j3}[33], \ldots)$.

3. Steganography by cover synthesis

- The probability of finding a match in one particular block among all K images is $1 - (1 - 1/16)^K$.
- The probability of being able to embed the whole message, which consists of n/16 groups of 4 bits is thus $(1 - (1 - 1/16)^K)^{n/16}$.

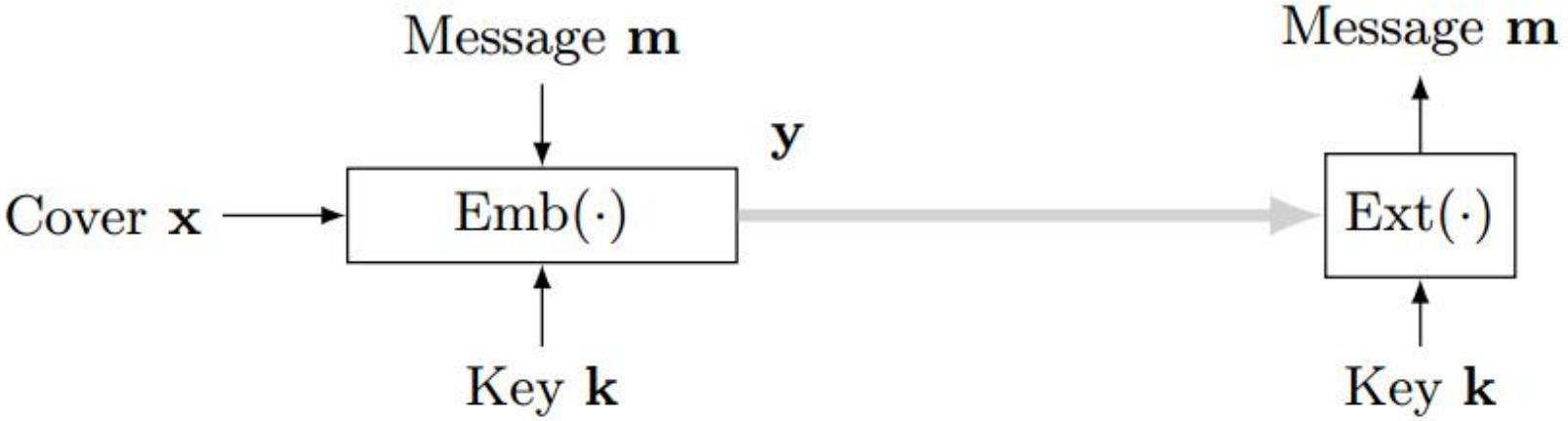
4. Steganography by cover modification

- Alice starts with a cover image and makes modifications to it in order to embed secret data.
- Alice and Bob work with the set of all possible covers and the sets of keys and messages that may, in the most general case, depend on each cover.

4. Steganography by cover modification

- C: set of cover objects x Є C
- K(x): set of all stego keys for x
- M(x): set of all messages that can be communicated in x
- Emb: C x K x M → C
- Ext: C x K → M

4. Steganography by cover modification



4. Steganography by cover modification

- we define the embedding capacity (payload) of cover x in bits as $\log_2|M(x)|$ and the relative embedding capacity is $\frac{\log_2|M(x)|}{n}$ where n is the number of elements in x.