Translation-Based Steganography

Christian Grothoff, Krista Bennett, Ludmila Alkhutova, Ryan Stutsman and Mikhail Atallah

Why use translation as a cover?

Natural Language (NL) translation is an inherently noisy process.

- Translations, regardless of quality, allow for a wide range of outputs
- Variation of a translation does not necessarily constitute “damage”
- Ready availibility of low-quality translations makes certain alterations plausible and errors easy to mimic

Automated NL Translation

- Far from perfect
  - Most common translation engines are statistical engines which translate via pattern-matching and sets of syntactic rules
  - Most ignore context completely, translating word-for-word and often ignoring syntactic and semantic differences between source and target languages
- Even if it ever becomes perfect, our technique still works!

A rich space for hiding information

- Plausibility regardless of quality, e.g.
  - If poor quality:
    Translations are already “damaged” by decisions made during translation. Thus, more space is available for hiding information by both mimicking and correcting errors.
  - If good quality:
    The inherent lack of a one-to-one correspondence between languages means that minor alterations can be introduced without rousing suspicion.
- Synonyms can alter the output without damaging the meaning of a translation, giving more plausible translation possibilities
- The overlap in error types between various translators makes text sources difficult to infer

What is our approach?

Approach relies on creation of multiple translations of cover text to encode hidden messages

- Dynamically configurable combination of translation engines and post-processing options creates candidate translations
- Secret message is encoded by choice of translation

Existing machine translation (MT) systems are used in a pluggable manner

- A number of MT systems can be multiplexed in order to create varied translations
- Additional MT systems can be added in order to both enhance bitrate and provide more “safety in numbers”
- MT systems trained on custom corpora can be introduced; when this is done, the corpus becomes part of the shared secret between users

Post-processing increases variation in number and quality of translations

- Automated semantic substitution provides plausible variation without hand-crafting substitution lists
- Error insertion modules add combinations of commonly observed translation errors
- Error correction modules rectify commonly observed translation errors

Quality and bitrate

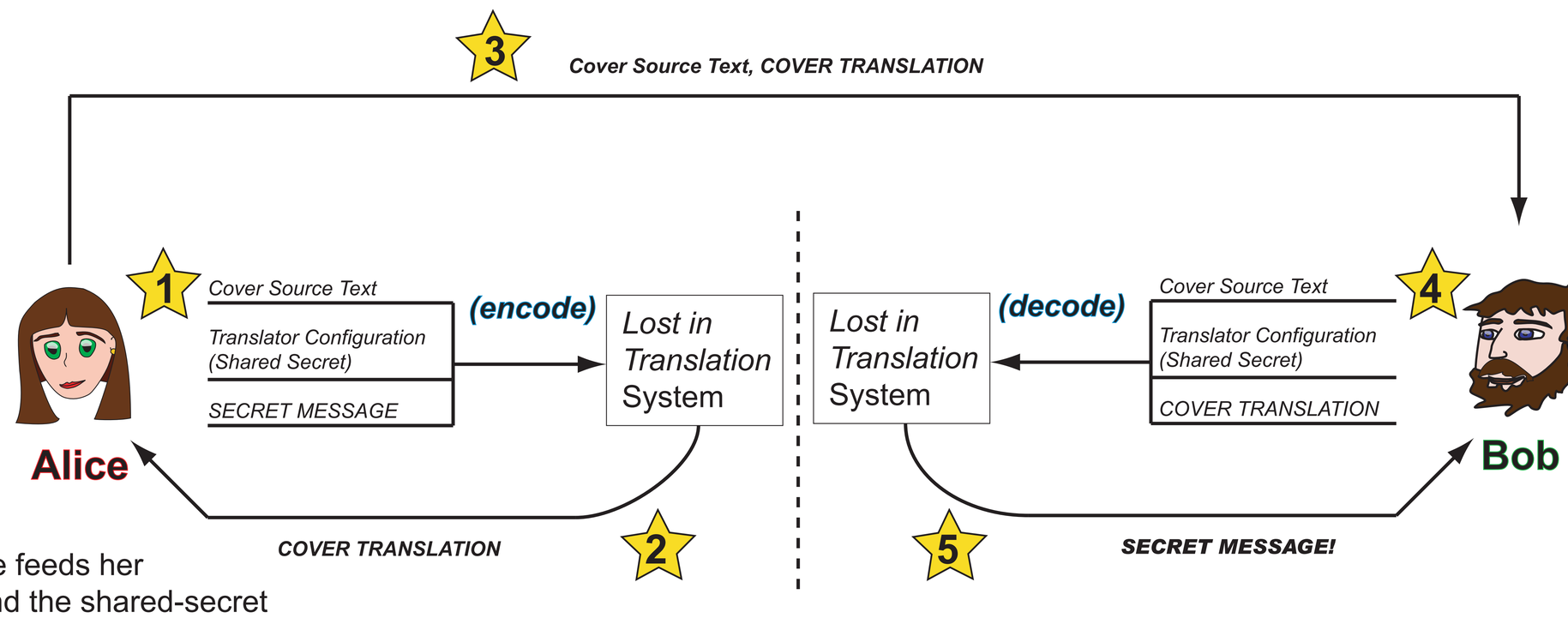
- Candidate translations are ranked according to quality estimates
- Huffman tree is constructed according to ranking in order to map bit sequences to sentences
- Lower bound can be placed on translation quality

The *Lost in Translation* (LiT) System

The Protocol

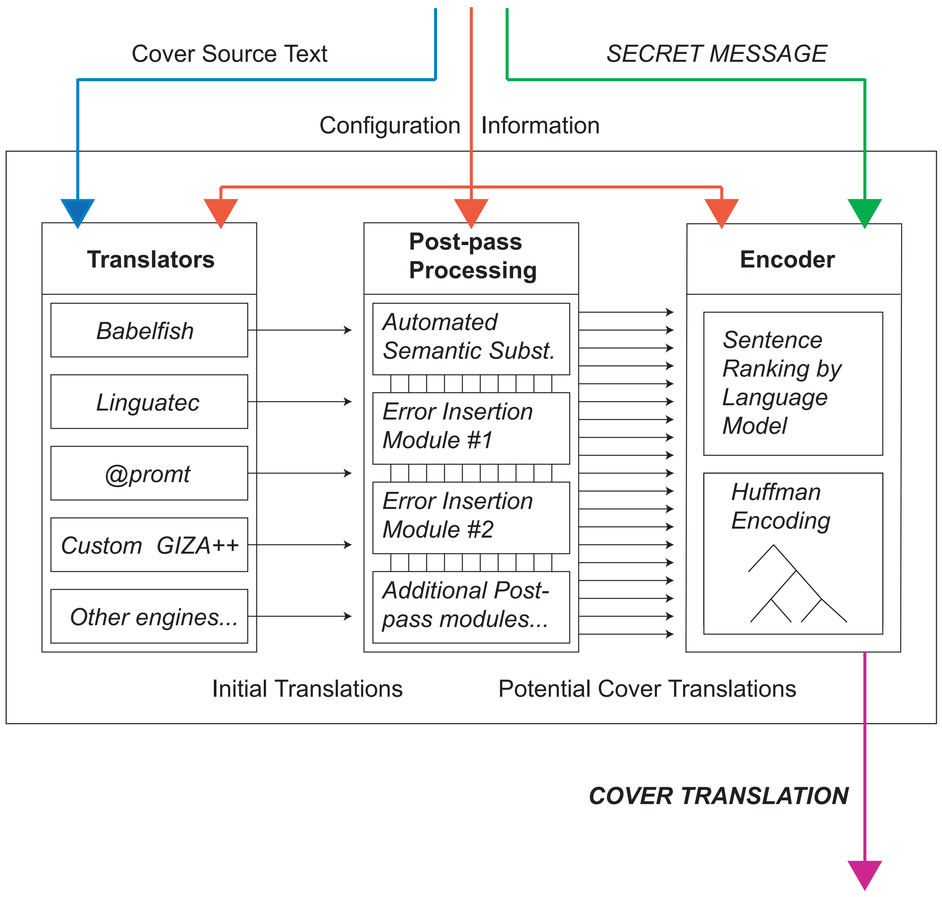
Alice wants to send a hidden message to Bob. (Previously, they securely shared the secret of their translator configurations with one another.)

Alice chooses a cover text; it could be a text from a public source, or she might openly send something to Bob.
It does not have to be a secret.

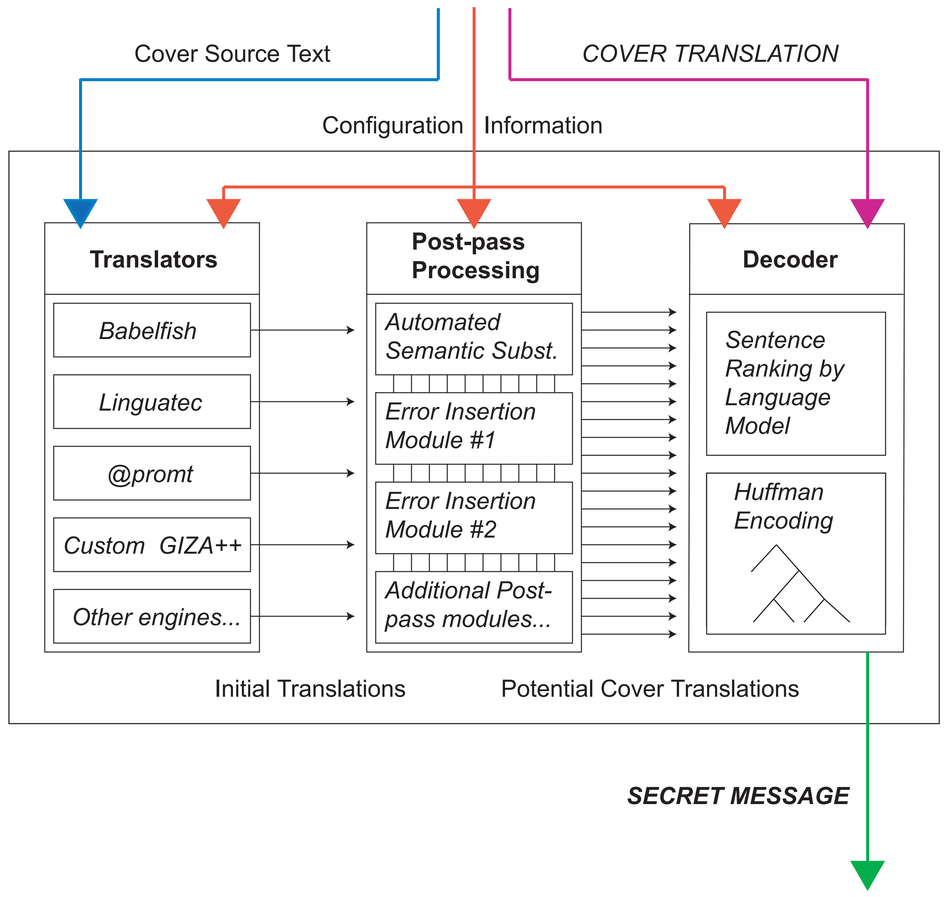


To send her message to Bob, Alice feeds her cover text, her secret message, and the shared-secret of the translator configuration into the system (1). The system then encodes the secret message within the translation, and returns a cover translation (2). Alice sends the cover translation, along with the cover source (or a reference to it) to Bob (3). Bob, who already has the shared-secret configuration, feeds the configuration, the cover source text, and the cover translation into his system (4). The system matches the cover text with the appropriate translation values and decodes the secret message (5).

Encoding



Decoding



Encoding and decoding of the secret message are nearly identical processes. The secret configuration information is fed into the system prior to processing, indicating which translation engines, post-pass processing modules, and language model to use. Once the system is configured, the cover source text is given to the translation engines. These then send their results on for post-pass processing. Post-pass modifications are made, and the candidate cover translations are sent to the encoder/decoder. Probabilities are assigned to each potential cover translation by the language model, and these probabilities are used to generate a Huffman tree of potential cover translations. This tree is then used to either encode the secret data, or to retrieve the encoded value of the cover translation.

REFERENCE IMPLEMENTATION of LiT AVAILABLE AT http://www.cs.purdue.edu/homes/rstutsma/stego/