

Information Hiding: A Generic Approach

Riad jabri, Boran Ibrahim and Hadi Al-Zoubi
Department of Computer Science, Faculty of Information Technology,
Philadelphia University, Amman, Jordan

Abstract: Problem statement: Privacy and security over communication channels are of primary concerns. Due to their complexity and diversity, there is a need for continuous improvements of the adopted solutions. In this study, we consider two of the adopted ones, namely, steganography and cryptography and propose a new information hiding system. **Approach:** The proposed system was based on a generic approach that incorporates text-based steganography and cryptography methods in a way that permits their combined or stand alone adoption. Thus, achieving message encryption incorporated with its concealing inside another unsuspecting one. Furthermore, two steganography methods (the inter-word spaces method and syntactic methods) had been combined with a hybrid text-encoding in a form of binary representation of terms rewriting systems. **Results:** An information hiding system had been implemented. The system offered encrypting and hiding dynamic and static text within a cover-text. The conducted experiments using static texts had shown a non-noticeable increase (0.02%) in the size of their respective stego-texts. For the dynamic texts, cover-texts with a size proportional to the length of the secret messages were needed. **Conclusion:** A generic model for information hiding with a respective implementation framework had been used as an effective tool to develop a hybrid and scalable steganography system that combined good features from the existing ones.

Key words: Information hiding, steganography, cryptography, genericity, rewriting systems, encoding, decoding

INTRODUCTION

Steganography is one of the information hiding techniques, defined as covered writing^[1]. It is the process of hiding data inside other data. For example, a text file could be hidden within an image or a sound file^[2]. For the purpose of our research (text-steganography), we consider setagography as a method of hiding a secret message in another message^[3]. Hence, steganography is about concealing the existence of the message. In contrast, cryptography is about concealing the contents of the message^[4]. The resulting product of steganography is called stego-text, while the resulting product of cryptography is called cipher text. Despite the covert and malicious uses of both, they also allow legitimate uses such as privacy and security over communication channels. Text-steganography proceeds according to the following scheme:

- A secret message (embedded, hidden data) is concealed in cover-text using an embedding algorithm to produce a stego-text

- The stego-text is then transmitted over a communication channel (Internet)
- Upon its delivery, the secret message is recovered using an extracting algorithm
- The embedding and the extracting algorithms are augmented by the so called a stego-key to encrypt and decrypt the hidden data respectively

The secret message is concealed using the following methods:

- Modification of the cover-text, such as insertion of spaces, misspelling, modifying the features (name, shape, position, color, size) of the individual characters^[5].
- Substitution, such as replacement of insignificant data within the cover text by hidden ones^[6].
- Generation, such as creation of a fake cover^[7].

The most recent efforts, techniques and tools are based on the presented scheme and make use of one or more of the above mentioned concealing methods. The ones related to our study are as follows:

Corresponding Author: Riad Jabri, Department of Computer Science, Faculty of Information Technology,
Philadelphia University, Amman, Jordan

- Por and Delina^[6] suggested an approach based on inter-word and inter-paragraph spacing to generate dynamic stego-text
- Bender *et al.*^[2] suggested a technique based on combining the following methods: Open space; syntactic (punctuation) and semantic encoding (synonym words)
- Kwan^[8] developed a tool, called SNOW, based on open space concealing method combined with compression and encryption
- Chapman and Davida^[9] suggested a technique based on natural language processing and using the sentence structures as a place for concealing data
- Bergmair^[10] investigated the different stegosystems that are based on natural language processing and proposed a linguistic coding scheme

Our proposed approach follows the presented scheme and defines a generic model for information hiding as the 5-tuple:

$$\text{GINM} = (\text{D1}, \text{D2}, \text{CO}, \text{SO}, \text{SD}, \text{CON}, \text{UCON})$$

Where:

- $\text{CO} = \text{CO1} \dots \text{CO}_n$ and $\text{SO} = \text{SO1} \dots \text{SO}_n$ represent the secret object and the cover object respectively. Such that: CO_i and SO_i are elements from a given domain D1
- $\text{CON}(\text{E}(\text{SO}), \text{CO}) \rightarrow \text{SD}$ is a concealing function. Such that SD is the stego-domain respective to embedding the encoded form of SO in CO
- $\text{E}(\text{SO}): \text{SO} \rightarrow \text{Sm}$ is a mapping function to encode SO into an object from the encoding domain D2
- $\text{UCON}(\text{SD}) \rightarrow \text{CO}$ is un concealing function that extracts the secret object from the respective stego-domain

Based on the GINM model, the construction of a steganography system is reduced to instantiating the generic functions from which the GINM is composed. For example, considering the cover and secret domains (D1 and D2) as alphabets from a given natural language, the secret and cover objects (SO and CO) are instantiated as a secret message and a cover text respectively. Where: $\text{CO1} \dots \text{CO}_n$ and $\text{SO} = \text{SO1} \dots \text{SO}_n$ are defined as characters from the language alphabet. A concealing function can then be defined based on different encoding and embedding methods. We borrow an example from^[5], where: The features of the individual characters (shape, position) are defined in a

form of the so-called codewords and are represented in a codebook, used by both an encoder and a decoder. Given a secret message SO, the concealing function $\text{CON}(\text{E}(\text{SO}), \text{CO})$ is then defined to substitute each $\text{SO}_i \in \text{SO}$ respective to $\text{CO}_i \in \text{CO}$ by watermarked one (Sm). Where Sm is produced by the function $\text{E}(\text{SO})$ as a mapping (codeword (codebook, CO)) from the codebook.

In this study, we have implemented text steganography system using the proposed approach and as described in the following sections. In addition to its efficiency and generalization, the proposed system is distinguished from similar ones by the following:

- The system permits its use as an encryption system
- The system is based on a generic approach and a generic implementation framework. Hence, it combines different encoding and embedding techniques

The system is a multi lingual. In addition, it accepts and generates both dynamic and static secret messages, as well as stego texts respectively.

MATERIALS AND METHODS

The main objective of this research is to develop an efficient and a generalized information hiding approach that contributes to the privacy and security of messages over communications channels. Based on such approach, a generic steganography system is defined based on instantiation of the proposed GINM model by the 5-tuple:

$$\text{GSTS} = (\text{L1}, \text{L2}, \text{CT}, \text{SM}, \text{ST}, \text{CON}, \text{UCON})$$

Where:

- CT and SM are a cover text and a secret message respectively, represented by characters from a given natural language L1
- L1 is a binary {0, 1} encoding language
- $\text{CON}(\text{E}(\text{SM}), \text{CT}) \rightarrow \text{ST}$ is a concealing function to embed the encoded, encrypted and compressed form of the Secret Message(SM) in the Cover Message (CT). As a result, a Stego Text (ST) is obtained
- $\text{E}(\text{SM}): \text{SM} \rightarrow \text{Sm}$ is a mapping function to encode, encrypt and compress SM into a binary string from the encoding language L2
- $\text{UCON}(\text{ST}) \rightarrow \text{SM}$ is un concealing function that extracts the secret message from the respective Stego-Text (ST)

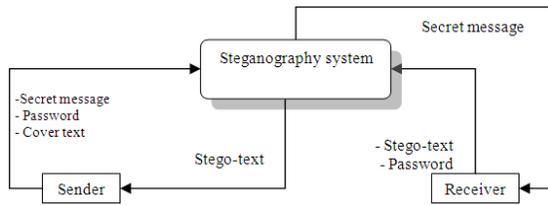


Fig. 1: Data flow diagram of the steganography system

The implementation of the steganography system GSTS is then reduced to the implementation of an interaction context and the functions: CON, E and UCON. They have been implemented according to the algorithms given below using C#.NET 2005 as a programming tool. As a result, a steganography system has been constructed with a data flow diagram as shown in Fig. 1 and the following functionality:

- The interaction context involves two users (a sender and a receiver) and the following activities. The sender interaction context facilitates: User authentication; browsing of the Secret Message (SM) and the Cover Text (CT) from their respective text files and initiation of the concealing process. In addition to authentication, the receiver interaction context facilitates: Browsing of the stego- text from its respective text file and initiation of the un concealing process
- The system responds to the sender's request by activating the function CON (E (SM), CT) to perform the respective: encoding; encryption; compression and embedding. As a result, the stego-text is displayed
- The system responds to the receiver's request by activating the function UNCON (ST) to perform the respective: Decoding; decryption; decompression and un embedding. As a result, the secret message is displayed

Encoding function: The implementation of the encoding function is based on the following idea: The Secret Message (SM) is decomposed into a set of patterns in encoded form. Hence, SM is defined as $SM = SM_1 SM_2 \dots SM_n$, $SM_i \in SM$, $\forall i \in [1, 2, \dots, n]$. SM is then decomposed based on a generic matching criterion as follows: For a given text SM, a generic matching predicate at position i of SM is defined as $MP_i (SM) = \{0, 1\}$. The text matching criteria is defined as:

$$MC(SM) = \bigcup_i MC_i, \quad \forall i \in [1, 2, \dots, n]$$

such that $MP_i = 1$. The encoding function is then defined as:

$$E(SM) = ((MC_i, T_i(SM) \times rws_i) \cup (MC_j, T_j(SM) \times rws_j) \cup \dots \cup (MC_n, T_n(SM) \times rws_n)) \rightarrow (P_i \cup \dots \cup P_n) \quad (1)$$

Where:

- $T_i (SM) \in SM$ = A string of characters from SM up to the position (i)
- $rws_i^{[11]}$ = Term rewriting rules, defined based on the encoding strategy
- $P_i = (MC_i, T_i (SM) \times rws_i)$ = A pattern, obtained as a result of rewriting $T_i (SM)$ according to rws_i

Thus, the encoding function $E(SM)$ as defined by display (1) can be implemented using any of the linguistic encoding methods, either using the syntactic ones or using the semantic ones. Further more, $E(SM)$ can be used as a stand alone encrypting function. For example, the rewriting rules (rws_i) can be defined as substitution ones to replace characters, words and paragraphs from the secret text (SM) by a respective synonyms from the same language or from a different language. In addition patterns can be transmitted by the sender in agreed upon order, where then they are assembled by the receiver according to the same order.

For our research purpose, the function $E(SM)$ is implemented as summarized by the algorithm given in algorithm 1 and as discussed below:

- The decomposition of secret text (SM) is performed based on syntactic methods, where SM is considered as composed of multiple lines. Each line is decomposed into subsequences $T_i (SM)$ based on the number of white spaces within each subsequence. Hence, the matching predicate at position i of SM is defined as $MP_i (SM) = 1$, if the character at that position is blank (white space). The text Matching Criteria $MC(SM)$ is then defined as number of the blanks up to a given position within SM. Subsequently, SM is decomposed into the subsequences $T_i (SM)$, $T_j (SM), \dots, T_n (SM)$ based on such criteria. To simplify the implementation of the function $E(SM)$, the number of the subsequences is determined based on the context of the secret message. In our implementation, we have assumed a maximum of three subsequences per line. Hence, a line i of SM is represented as $T_{i1}(SM) T_{i2}(SM) T_{i3}(SM)$.

Therefore, the text is represented as $\bigcup_i T_i(SM)$
 $T_2(SM) T_3(SM)$. Where $i \in [1, 2, \dots, n]$ represents
the line number

- The individual subsequences ($T_i(SM)$) obtained from step 1 are then processed by the respective rewriting rules (rws_i), defined as the composite function:

$RSW_i: \text{Compress}(\text{Encrypt}(\text{Binary}(T_i(SM))) \rightarrow P_i$

Where:

- P_i is a pattern representing the subsequence $T_i(SM)$ in encoded form
- Binary is a function that converts $T_i(SM)$ into a binary according to two methods. The first method uses the UTF-8 encoding to facilitate dynamic secret messages and subsequently dynamic stego-text. The second method uses Huffman code^[12] with squeezing to facilitate static, but efficient encoding
- Encrypt is a function that encodes the stream of bytes as generated by the function Binary using the built-in C#.NET encryption tools. Where, such stream is "exclusive-ored" with a random key
- Compress is function that reduces the size of the encoded and the encrypted form of $T_i(SM)$ by eliminating the redundant bytes and dividing the resulting sequence by a constant n

Algorithm 1:

Encoding algorithm: The implementation algorithm of the function $E(SM)$:

Input: The secret Message SM.

Output: The encoded form of SM represented by the set of patterns P_i .

Method:

For each line L_i in SM

```

s = Compute spaces (Li);  Ti-length = s mod 3;
If (Tij-length < > 0)
    { max-length = s div 3;  max-length1= max-length +
    remainder;
    Ti1-Length= Ti2-Length= max-length;  Ti3-Length=
    max-length1;
    }Elseif { Ti1-Length= Ti2-Length = Ti3-Length = s
    div 3,
    {{Ti1(Li), Ti2(Li), Ti3(Li)} = Decompose( Li);
    {Pi1(Li), Pi2(Li), Pi3(Li)}= RSW ({Ti1(Li), Ti2(Li),
    Ti3(Li)});
    }
    
```

Example 1: Let $SM =$ "The implementation algorithm of the function $E(SM)$ ". Applying the encoding algorithm using Huffman code will produce the following subsequences and patterns:

$T_1(SM) =$ "The implementation" $\rightarrow P_1 = 000$

$T_2(SM) =$ "algorithm of" $\rightarrow P_2 = 0010$

$T_3(SM) =$ "the function $E(SM)$ " $\rightarrow P_3 = 0011$

Concealing function: A generic implementation for the concealing function CON is defined as:

$$CON(E(SM), CT) = ((EC_i, P_i \times CT|_i \times er_i) \cup \dots \cup (EC_n, P_n \times CT|_n \times er_n) \rightarrow ST_i \dots ST_n \quad (2)$$

Where:

- The embedding method is represented by embedding criteria $EC_i \dots EC_n$ and respective rewriting rules $er_i \dots er_n$
- P_i, \dots, P_n is the patterns generated by the encoding function $E(M)$
- $(CT|_i, \dots, CT|_n) \in CT$ represents strings of characters from the Cover Text (CT). These strings are selected from the text CT based on the embedding criteria as appropriate covers for embedding the patterns P_i, \dots, P_n respectively
- $ST_i = (P_i + CT|_i), \dots, (ST_n = P_n + CT|_n)$ represent the stego texts generated by the function $E(SM)$, as a result of embedding the patterns P_i, \dots, P_n within the covers $CT|_i, \dots, CT|_n$ according to the rules er_i, \dots, er_n respectively

Based on the definition as given by display (2), the implementation of the concealing function is reduced to its instantiation by a particular embedding method. We adopt a method that is similar to the one suggested by Por and Delina^[6]. But, with appropriate modifications. The modified method is a combination of the open space method and the syntactic method. Its implementation algorithm is given in algorithm 2.

Where:

- The compressed patterns P_i, \dots, P_n are rewritten as respective sequence of white spaces. Such that the digit "1" is rewritten as two spaces and the digit "0" is rewritten as one space
- The embedding criteria and the rewriting rules are defined based on the white spaces and the punctuations occurring within the cover text to meet the following objectives:
 - To select the covers $(CT|_i, \dots, CT|_n) \in CT$ that are suitable for embedding the corresponding

patterns P_1, \dots, P_n . Hence, the embedding criteria is reduced to degree of suitability in terms of the number of the white spaces needed by the individual patterns. Based on such criteria, a function (split) is defined to decompose the cover text into individual covers (CT_i), consisting of one or more cover lines

- To rewrite each cover $CT_i \in CT$ by inserting the white spaces respective to its corresponding pattern $P_i \in T(SM)$
- To distinguish between the white spaces as they occur within the text CT and the ones used for rewriting the individual patterns. Hence, punctuations are used as end markers for the individual patterns
- To contribute to the quality of information hiding in terms of its security and robustness. Hence, the embedding criteria and subsequently, the function split are extended by the requirement for a random allocation of the individual covers rather than a uniform one

Algorithm 2:

Embedding algorithm:

Input: The cover text CT and the set of patterns $\{P_1, \dots, P_n\}$

Output: The stego text represented by the set $\{ST_1, \dots, ST_n\}$.

Method:

For each pattern P_i

{ $CT_i = \text{Split}(CT)$;

For $j = 1$ to $P_i.\text{length}$

{ If ($pi[j] = "1"$)

{ $ST_i = ST_i + CT_i[j] + " "$ }

Elseif { $ST_i = ST_i + CT_i[j] + " "$ }

} $ST_i = ST_i + \text{"end marker"}$

$ST_i = ST_i + \text{Remaining}(CT_i)$; Return ST_i

Un concealing Function: A generic implementation for the un concealing function UNCON is defined as:

$$\text{UNCON}(ST) = ((DC_i, ST_i \times dr_n) \cup \dots \cup (DC_i, ST_i \times dr_n)) \rightarrow P_1 \dots P_n \rightarrow SM_1 \dots SM_n \quad (3)$$

Where:

- The decoding method is represented by respective criteria $DC_1 \dots DC_i$ and rewriting rules $dr_1 \dots dr_n$ with an objective to decode the embedded white spaces within the individual stego-texts ST_i into their respective patterns P_i

- The individual patterns P_i are then decompressed, decrypted and decoded into the respective parts SM_i of the secret message SM

Based on display 3, the implementation of the function UNCON is reduced to the instantiation of its generic definition by specific algorithms. For example, the decoding of the individual patterns P_i into their respective part SM_i of the secret message MS is give in algorithm 3.

Algorithm 3:

Decoding algorithm:

Input: The individual patterns P_i

Output: The respective part SM_i of the secret Message

Method:

$c = 0$

For $j = 0$ to $(P_i.\text{length} - 1)$

{ For $n = 7$ to 1

{ if ($P_i[c++] = "1"$)

{ $SM_i[j] = (SM_i[j] | (1 << n))$

} Return SM_i

RESULTS

Based on the proposed approach and its respective implementation methodology, a steganography system has been developed with an interaction context represented by two forms as given in Fig. 2 and 3 respectively. The first form is denoted by encoding facilitates interaction with the presented encoding and concealing functions. Furthermore, it is augmented with quality indicators such as the size of the secret message and the browsed cover text as well as the hiding ratio. The latter gives the utilization percentage of the cover text by the hidden message. The second form of the interaction context is denoted by decoding and facilitates interaction with the presented un concealing function.

Through its interaction contexts, the proposed steganography system has been tested using several multilingual texts (Arabic and English). The results are summarized as follows:

Results for static stego-texts: The static-stego texts are generated using Huffman code with a compression. We have tested texts with different size. Representative results are given in Table 1 in terms of: 1) the size of the Secret Message (SM); the cover text (CT) and the Stego Text (ST) and 2) the number of the patterns that are hidden in the stego- text as respective encoding of the secret message.

Table 1: Testing results for different secret messages

No. of Patterns	Size of SM (KB)	Size of CT (KB)	Size of ST (KB)
24	0.50	1.10	1.29
10	0.24	0.48	0.58

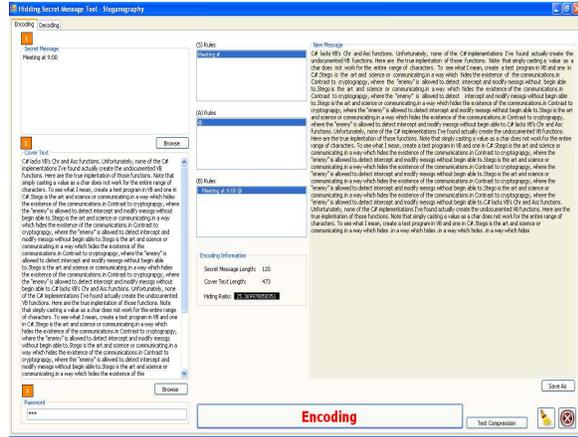


Fig. 2: The "encoding" interaction context

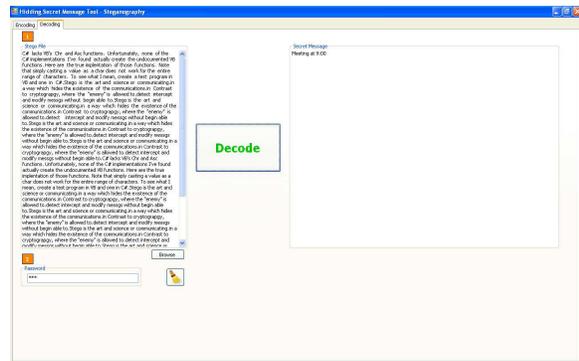


Fig. 3: The "decoding" interaction context

Results for dynamic stego-texts: The dynamic stego-texts are generated based on binary encoding. Hence, secret messages with changeable contents can be hidden within different cover texts. The experiments on different messages have shown 1:4 ratios between the sizes of the secret message and the cover text. Figure 2 shows a secret message and a cover text that has a 25% utilization percentage.

DISCUSSION

The experimental results have demonstrated the efficiency and flexibility of the proposed text steganography system. Furthermore, the system is based on an approach that is formalized in a generic

way. This enables different methods to be combined and adopted for its implementation and contributes to its improvements robustness and scalability. For examples:

- Two encoding methods have been adopted, the binary and the Huffman encoding. Such adoption permits comparing their advantages and disadvantages. It was found that first method is characterized by its flexibility. It enables hiding dynamic secret texts. However, it requires cover texts with a considerable larger size. On the other hand, the second method is associated with the extra overhead, in terms of time and space, needed for encoding and decoding
- The adopted embedding criteria combine suitability and randomness to ensure robustness of the stego-text

Compared to similar systems such as the one suggested by Por and Delina^[6], the proposed system has a comparable results in terms of ratios between the sizes of the secret messages and the cover texts. Our approach has achieved 1:4 ratio while in^[6] a cover text with a size <16 kb is required for a secret message with a length <4 kb. However, our system has better quality indicators such as less utilization percentage of the cover text and its robustness. Furthermore, it can be adopted for dynamic and static secret texts as well as cover texts.

CONCLUSION

In this research, a generic information hiding model has been suggested. Based on such model, a text steganography system has been implemented. The system is characterized by its generality, scalability and flexibility. Although the proposed system has better quality indicators than the ones for similar system, more improvements are needed for such indicators. Mainly, the capacity of the cover text and the robustness of the stego text. Hence, efforts in this direction constitute a future research.

REFERENCES

1. Bennett, K., 2004. Linguistic steganography: Survey, analysis and robustness concerns for information hiding in text, Purdue University, CERIAS Technical Report. https://www.cerias.purdue.edu/apps/reports_and_papers/view/2697

2. Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. Techniques for data hiding. *IBM Syst. J.*, 35: 313-336. <http://portal.acm.org/citation.cfm?id=243519.243522>
3. Cachin, C., 1998. An information-theoretic model for steganography. *Lecture Notes Comput. Sci.*, 1225: 306-318. <http://portal.acm.org/citation.cfm?id=731689>
4. Denning, D., 1982. *Cryptography and Data Security*. Addison-Wesley Publishing Company, USA, ISBN: 0-0201-10150-5, p. 414. <http://portal.acm.org/citation.cfm?id=SERIES11430.539308>
5. Eggers, J.J., J. Su and B. Girod, 2000. A Blind Watermarking Scheme Based On Structured Codebooks. *Proceeding of the IEE Seminar on Secure Images and Image Authentication*, Apr. 10, IEE Colloquium, London, UK, pp: 4/1-4/21. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=848144
6. Por, L. and B. Delina, 2008. Information hiding: A new approach in text steganography. *Proceeding of the 7th WSEAS International Conference on Applied Computer and Applied Computational Science*, Apr. 6-8, Hangzhou, Chiana, pp: 689-695. <http://www.wseas.us/e-library/conferences/2008/hangzhou/acacos/116586-634.pdf>
7. Wayner, P., 1996. *Disappearing Cryptography: Being and Nothingness on the Net*, Morgan Kaufman Publishers, ISBN: 13: 978-01273867 , p. 295
8. Kwan, M., 1998. The SNOW homepage. <http://www.darkside.com.au/snow/>
9. Chapman, M. and G. Davida, 1997. Hiding the hidden: A software system for concealing ciphertext as innocuous text. *Lecture Notes Comput. Sci.*, 1334: 335-345. DOI 10.1007/BFb0028489
10. Bergmair, R., 2007. A Comprehensive Bibliography of Linguistic Steganography. In: *Security, Steganography and Watermarking of Multimedia Contents IX*, Delp, E.J. (Ed.). ISBN: 13: 978-0819452092, pp: 880.
11. Bertolissi, C. and M. Fernandez, 2008. A rewriting framework for the composition of access control policies. *Proceedings of the 10th International ACM Conference on Principles and Practice of Declarative Programming*, July 15-17, ACM Press, USA., pp: 217-225. <http://portal.acm.org/citation.cfm?id=1389449.1389476>
12. Larmore, L.L., 1995. Constructing Huffman Trees in parallel. *SIAM J. Comp.*, 24: 1163-1169 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.3927>