The Journal of Systems and Software 86 (2013) 324–334

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A covert communication method via spreadsheets by secret sharing with a self-authentication capability☆

Che-Wei Lee[a,1], Wen-Hsiang Tsai[a,b,*]

[a] Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan
[b] Department of Information Communication, Asia University, Taichung 41354, Taiwan

ARTICLE INFO

Article history:
Received 3 March 2012
Received in revised form 18 July 2012
Accepted 18 August 2012
Available online 30 August 2012

Keywords:
Covert communication
Secret sharing
Information hiding
Self-authentication
Spreadsheet

ABSTRACT

A new covert communication method with a self-authentication capability for secret data hiding in spreadsheets using the information sharing technique is proposed. At the sender site, a secret message is transformed into shares by Shamir's (k, n)-threshold secret sharing scheme with $n = k + 1$, and the generated $k + 1$ shares are embedded into the number items in a spreadsheet as if they are part of the spreadsheet content. And at the receiver site, every k shares among the $k + 1$ ones then are extracted from the stego-spreadsheet to recover $k + 1$ copies of the secret, and the consistency of the $k + 1$ copies in value is checked to determine whether the embedded shares are intact or not, achieving a new type of blind self-authentication of the embedded secret. By dividing the secret message into segments and applying to each segment the secret sharing scheme, the integrity and fidelity of the hidden secret message can be verified, achieving a covert communication process with the double functions of information hiding and self-authentication. Experimental results and discussions on data embedding capacity, authentication precision, and steganalysis issues are also included to show the feasibility of the proposed method.

1. Introduction

Covert communication is a technique of concealing secret information into a *cover medium* in an imperceptible way or with a camouflage effect such that only a sender and an intended receiver know the existence of the hidden data in the resulting *stego-medium*. In the literature, emphases were put on the use of multimedia like images, videos, and audios (Wu et al., 1999; Gopalan et al., 2003; Chae and Manjunath, 1999; Cheddad et al., 2010) because these media in general provide larger embeddable spaces and cause less suspicion due to their wide distributions. And weaknesses existing in human beings' visual capabilities are often exploited to design effective covert communication methods. For example, the methods proposed in Bender et al. (1996), Wu and Tsai (2003), and Yang et al. (2008) replace the least-significant bits of pixels in cover images to embed information, and that of Fridrich and Du (2000) uses the parities of palette colors, composed by similar colors, to represent hidden message bits.

In addition to methods developed for multimedia, several others (Brassil and Maxemchuk, 1999; Lee and Tsai, 2010a,b; Zhong et al., 2007; Liu and Tsai, 2007) used cover media of text, PDF, or Word documents for covert communication. In Brassil and Maxemchuk (1999), data are embedded by slightly adjusting the lines, tabs, or characters in text files. Lee and Tsai (2010a,b) used special ASCII codes in PDF files to embed data between characters. Liu and Tsai (2007) made use of the change tracking function in Microsoft Word to embed data imperceptibly by a document degeneration technique.

In this study, we propose a new covert communication method which applies Shamir's (k, n)-threshold secret sharing scheme (Shamir, 1979) with $n = k + 1$ to a given secret item to yield $k + 1$ shares, and the generated $k + 1$ shares are embedded into the number items in a spreadsheet as if they are part of the spreadsheet content. The purpose of transforming the secret data into secret shares by the $(k, k + 1)$-threshold secret sharing scheme is *not* to enforce robustness, but to yield a blind self-authentication capability for the embedded secret. Conventionally, the concept of (k, n)-threshold secret sharing is applied to provide destruction-tolerant capabilities. That is, any k shares collected from n ones may be processed to reveal the shared secret even though up to $(n - k)$ shares are destroyed. But in the proposed method, the scheme of $(k, k + 1)$-threshold secret sharing is developed for the first time

☆ This work is supported financially by the National Science Council, Taiwan, ROC under Project No. 99-2631-H-009-001.

* Corresponding author at: Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan.
Tel.: +886 3 5728368; fax: +886 3 5734935.
E-mail addresses: paradiserlee@gmail.com (C.-W. Lee), whtsai@cis.nctu.edu.tw (W.-H. Tsai).
[1] Tel.: +886 3 5728368; fax: +886 3 5734935.

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.08.048

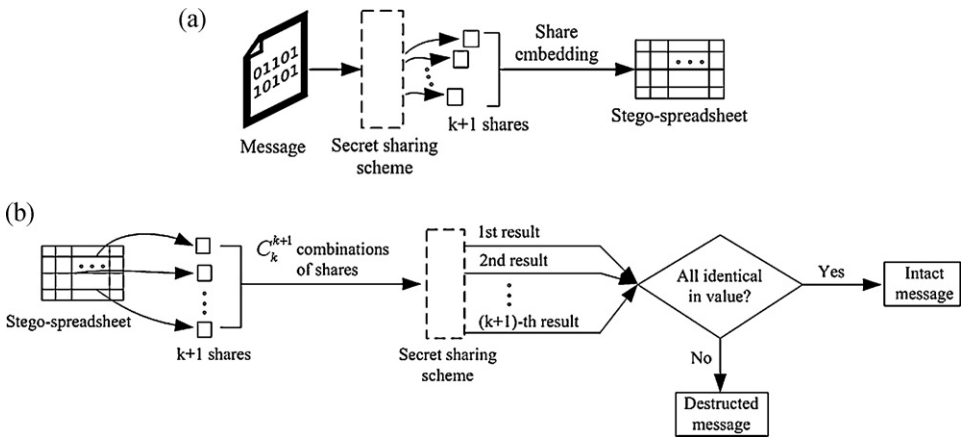


Fig. 1. Illustration of proposed covert communication method via spreadsheets by secret sharing. (a) Generation of a stego-spreadsheet. (b) Self-authentication of the extracted message.

to provide instead a self-authentication capability by checking the *value-consistency* of $k+1$ results coming from all $k+1$ combinations to determine whether the extracted secret is intact or not. That is, only when the results computed from any k shares collected from $k+1$ shares are *all identical* in value can the extracted secret be decided to be intact. Fig. 1 illustrates these core ideas of the proposed method.

Moreover, to conceal the presence of hidden data, secret shares are spread throughout the cover spreadsheet in a *sparsely* fashion. And a spreadsheet containing numeral items with a *high scatter level* is more suitable to be used as a cover spreadsheet for better concealment. Merits of the proposed method include the following. (1) A receiver can confirm the correctness of the extracted secret message. (2) Compared with some methods using hash codes or parity bits as redundant data to ensure the authenticity of retrieved data, only a minor redundancy, i.e. the $(k+1)$-th share, is needed in the proposed method. (3) By adaptively choosing involved parameters, i.e. the value of p, used in the polynomial of Shamir's method for the selected spreadsheet, the numerical items' values generated by the method will fall into a reasonable range of values, arousing little suspicion during covert communication. (4) Using spreadsheets as cover media, the proposed method is free from *unintentional* destruction of hidden data like data compression during the secret transmission or data keeping process, in contrast with cover media like images or videos which are often compressed *ignorantly* in such a process. Two examples of such documents, Microsoft Excel and Google Docs, are shown in Fig. 2.

The remainder of this paper is organized as follows. In Section 2, the Shamir method on which the proposed method is based is reviewed first. In Section 3, the details of the proposed method, including secret message embedding, secret message extraction, and self-authentication of the extracted message, are described. In Section 4, discussions on related issues about the proposed method are given. Experimental results are presented in Section 5, followed by conclusions in Section 6.

2. Review of Shamir's method for secret sharing

In the (k, n)-threshold secret sharing scheme proposed by Shamir (1979) with $k \le n$, a secret d in the form of an integer is transformed into shares which then are distributed to n participants to keep; and as long as up to k of the n shares can be collected, the original secret can be recovered. The detail of the scheme may be described as two algorithms in the following.

Algorithm 1. (k, n)-threshold secret sharing.

Input: a secret d in the form of an integer, the number n of participants, and a threshold k not larger than n.

Output: n shares in the form of integers for n participants to keep.

Steps.

1. Choose randomly a prime number p which is larger than the secret d.
2. Select $k-1$ integer values $c_1, c_2, \ldots, c_{k-1}$ within the range of 0 through $p-1$.
3. Select n distinct real values for the variables $x_1, x_2, \ldots, x_n$.
4. Use the following $(k-1)$-degree polynomial to compute n function values $F(x_i)$, called *partial shares*:

$$F(x_i) = (d + c_1 x_i + c_2 x_i^2 + \cdots + c_{k-1} x_i^{k-1})_{\bmod p}, \quad (1)$$

for $i = 1, 2, \ldots, n$.

5. Deliver the 2-tuple $(x_i, F(x_i))$ as a *share* to the ith participant, where $i = 1, 2, \ldots, n$.

Since there are k coefficients, including d and c_1 through c_{k-1}, in (1) above, it is necessary to collect at least k shares from the n participants to form k equations of the form of (1) to solve these k coefficients in order to recover the secret d. This explains the term, *threshold*, for k and the name, (k, n)-*threshold*, for the Shamir method. Below is a description of the equation-solving process for secret recovery.

Algorithm 2. Secret recovery.

Input: k shares collected from the n participants where k is the threshold mentioned in Algorithm 1; and the prime number p which was chosen in Step 1 of Algorithm 1.

Output: the secret d hidden in the shares and the coefficients c_i used in the equations described by (1) in Algorithm 1, where $i = 1, 2, \ldots, k-1$.

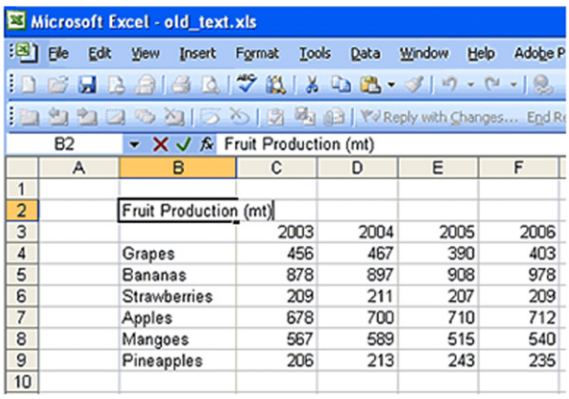
Steps.

1. Use the k shares $(x_1, F(x_1)), (x_2, F(x_2)), \ldots, (x_k, F(x_k))$ to set up the following equations:

$$F(x_j) = (d + c_1 x_j + c_2 x_j^2 + \cdots + c_{k-1} x_j^{k-1})_{\bmod p}, \quad (2)$$

where $j = 1, 2, \ldots, k$.

(a)



	A	B	C	D	E	F
1						
2		Fruit Production (mt)				
3			2003	2004	2005	2006
4		Grapes	456	467	390	403
5		Bananas	878	897	908	978
6		Strawberries	209	211	207	209
7		Apples	678	700	710	712
8		Mangoes	567	589	515	540
9		Pineapples	206	213	243	235
10						

(b)

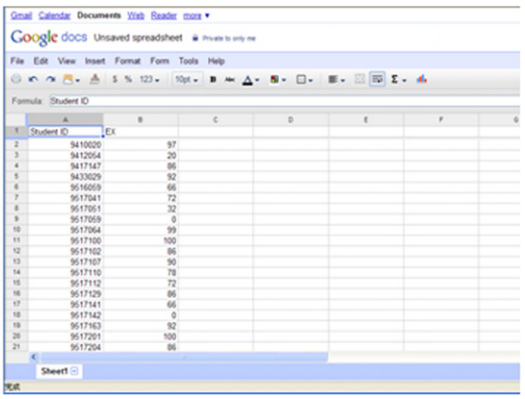


Fig. 2. Examples of spreadsheets. (a) Microsoft Excel. (b) Google Docs.

2. Solve the k equations above by Lagrange's interpolation to obtain the desired secret value d (Lin and Tsai, 2004) as follows:

$$d = (-1)^{k-1}\left[F(x_1)\frac{x_2x_3\ldots x_k}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + F(x_2)\frac{x_1x_2\ldots x_k}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} + \cdots + F(x_k)\frac{x_1x_2\ldots x_{k-1}}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})}\right]_{\bmod p}.$$

3. Compute the values c_1 through c_{k-1} by expanding the following equality and comparing the result with (2) in Step 1 while regarding the variable x in the equality below to be x_j in (2):

$$F(x) = \left[F(x_1)\frac{(x-x_2)(x-x_3)\cdots(x-x_k)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + F(x_2)\frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} + \cdots + F(x_k)\frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})}\right]_{\bmod p}.$$

Step 3 in the above algorithm is included for the purpose of computing the values of the parameters c_i in the proposed method. In other applications, if only the secret value d need be recovered, this step may be eliminated.

3. Proposed covert communication method using spreadsheets

3.1. Generation of a stego-spreadsheet

In the proposed method, an appropriate cover spreadsheet S which contains numeric data for disguising generated secret shares is prepared first. Next, a secret message M to be hidden is divided into several segments, and taken as input to Shamir's (k, n)-threshold secret sharing scheme (Shamir, 1979) with carefully chosen parameters to generate secret shares. Then, numeric items in S which are selected by a secret key are replaced with the shares to generate a stego-spreadsheet S'. In this process, the parameters involved in Eq. (1) of Algorithm 1 are adjusted to satisfy the characteristics of the input secret message and the prepared cover spreadsheet. These parameters include: (a) the number m of bits in each message segment, which is also taken to be the identical numbers of bits in all of the coefficients d, c_1 through c_{k-1}; (b) the number k of message segments processed by the Shamir scheme each time, which is also the minimum number k of secret shares needed to be collected to recover the secret; (c) the total number n of generated shares, which is set to be $k+1$ specifically; (d) and the prime number p, which is the smallest integer larger than all the values of the coefficients d, c_1 through c_{k-1}, and the variables x_1 through x_n used in Eq. (1) (Shamir, 1979).

A detailed algorithm describing the process is presented in the following.

Algorithm 3. Generation of a stego-spreadsheet.

Input: a binary secret message M divided into m-bit segments, a spreadsheet S, a secret key K, and three pre-selected integers k, n ($=k+1$), and m.

Output: a stego-spreadsheet S'.

Steps.

Stage 1 – share generation.

Step 1. Choose the smallest prime number p which is larger than $2^m - 1$.

Step 2. Take sequentially k unprocessed m-bit segments from M to form a group G, called *segment group*, and perform the following steps to transform the segment group into partial shares.

2.1 Transform the k m-bit message segments in G into integers and take the results to be d, c_1, c_2, …, c_{k-1}, respectively.

2.2 Take x_1 through x_n to be the integers 1 through n, respectively, where $n = k+1$.

2.3 Use the following $(k-1)$-degree polynomial to compute n partial shares $F(x_i)$:

$$F(x_i) = (d + c_1x_i + c_2x_i^2 + \cdots + c_{k-1}x_i^{k-1})_{\bmod p}, \quad (3)$$

where $i = 1, 2, \ldots, n$.

2.4 Save all $F(x_i)$ in order into a *partial-share set* F_{ps}.

Step 3. If the message segments in M are not exhausted, then go to Step 2 to process another segment group; otherwise, continue.

Stage 2 – partial share embedding.

Step 4. Take an unprocessed partial share $F(x_i)$ from F_{ps}, and perform the following steps.

4.1 Use the secret key K to randomly select a numeric item I in S.
4.2 Replace I with $F(x_i)$.

Step 5. If there exist unprocessed partial shares in F_{ps}, go to Step 4; otherwise, take the final S as the output S'.

3.2. Algorithm for data extraction and authentication

The proposed blind self-authentication capability for verifying a recovered secret message is fulfilled by the $(k, k+1)$-threshold secret sharing scheme. In the past, the concept of (k, n)-threshold secret sharing is often applied to develop methods for secret image sharing (Lin and Tsai, 2004; Thien and Lin, 2002; Chen and Lin, 2010) or image repairing (Lee and Tsai, 2010a,b) with destruction-tolerant capabilities – any k shares collected from the n ones may be processed to reveal the shared secret even though up to $(n-k)$ shares are destroyed. But in the proposed method here, the scheme of $(k, k+1)$-threshold secret sharing is developed to provide a self-authentication capability for verifying the correctness of a recovered segment group in the secret message – *any* k shares collected from the $k+1$ ones should, after the secret recovery process of Algorithm 2 is conducted, reveal the same secret value in normal cases, meaning that no damage ever occurs to the $k+1$ shares; otherwise, it can be decided that some shares must have been destroyed. By making use of this characteristic, blind self-authentication of each segment group in the recovered secret message is carried out, and verification of the integrity and fidelity of the secret message thus achieved. A detailed algorithm of secret message recovery and self-authentication is described in the following.

Algorithm 4. Secret data recovery and self-authentication.

Input: a stego-spreadsheet S'; the prime number p, the three integers k, n ($=k+1$), and m, and the secret key K used in Algorithm 3.

Output: a secret message M hidden in S' presumably, and a report about the authenticity of the segments within M.

Steps.

Stage 1 – message segment computation.

Step 1. Use the secret key K to select randomly numeric items in S'; take out their values which presumably are the partial shares $F(x_i)$ embedded by Algorithm 3; and put the items sequentially into a set F_{ps} as a partial-share set.

Step 2. Take out in order n partial shares from F_{ps}, set their corresponding x values as 1 through n, respectively, and perform the following steps to recover a binary segment M_i of the secret message M, if possible.

2.1 For every k partial shares $F_1, F_2, \ldots, F_k$ in the n ones and their corresponding x values $x_1, x_2, \ldots, x_k$, perform the following steps.

2.1.1 Use the k shares $(x_1, F_1), (x_2, F_2), \ldots, (x_k, F_k)$ to set up the following equations:

$$F_j = F(x_j) = (d + c_1 x_j + c_2 x_j^2 + \cdots + c_{k-1} x_j^{k-1})_{\bmod p}, \quad (4)$$

where $j = 1, 2, \ldots, k$.

2.1.2 Compute the values d and c_1 through c_{k-1} by expanding the following equality and comparing the result with (4) in Step 2.1.1 above while regarding the variable x in the equality below to be x_j in (4):

$$F(x) = \left[F(x_1)\frac{(x-x_2)(x-x_3)\cdots(x-x_k)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + F(x_2)\frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} + \cdots + F(x_k)\frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})} \right]_{\bmod p}.$$

2.1.3 Put the computed values of d and c_1 through c_{k-1} as a set into a buffer B.

(There will be $n = k+1$ sets of values of d and c_1 through c_{k1} at the end of Step 2.)

Stage 2 – self-authentication of the computed message segment.

Step 3. Take out the n sets of the coefficient values of d and c_1 through c_{k-1} in B and perform the following operations.

3.1 Transform the coefficients d and c_1 through c_{k-1} into k binary segments, and concatenate them as a message segment M_i.

3.2 If all the n sets of the coefficient values are identical to one another, then mark M_i as *authentic* and append it to the end of the desired secret message M; else, mark M_i as *having been damaged* and continue.

Step 4. If all shares embedded in S' are processed, then take the final M as the output; otherwise, go to Step 2.

4. Discussions on related issues about proposed method

4.1. Statistical undetectability

A statistical anomaly caused by information embedding is a reliable clue to detect the presence of the steganographic content (Provos and Honeyman, 2003). For the purpose of resisting such statistical analysis, two strategies are used in the proposed method. One is to spread secret shares throughout the cover spreadsheet in a *sparsely* and randomly distributed fashion so that less affection is incurred to the statistical properties of the cover spreadsheet after information embedding. This way of achieving undetectability for a hidden message used in the proposed method follows the concept of the *frequency-hopping spread spectrum* technique (Pickholtz et al., 1982) in which radio signals are transmitted by many frequency channels selected according to a pseudorandom sequence known to the sender and the receiver. The other strategy is to choose *comparatively insignificant parts* of numeric data in the spreadsheet for embedding secret shares in order to keep a low level of embedding strength for maintaining the statistical properties in a stego-spreadsheet. For example, we may choose the decimal fractions of the numbers in a cover spreadsheet and replace their values with those of the secret shares, resulting in insignificant alterations to the statistical property in the stego-spreadsheet.

4.2. Active security consideration

The proposed method not only can passively prevent the stego-spreadsheet from detection but also can actively ensure the fidelity and integrity of the transmitted secret. In the active attack model mentioned in Liu and Tsai (2007), if an adversary subtly made modifications to passing-by stego-spreadsheets for the purpose of misleading a receiver, the blind self-authentication capability provided by the proposed method can be used to check the authenticity of the retrieved secret message. When the authenticity check fails, it reveals that the communication between the two sides has been threatened and appropriate measures should be adopted.

4.3. Embedding capacity analysis

The value k mentioned in Step 2 of Algorithm 3 determines the number of message segments, or equivalently, the total number of

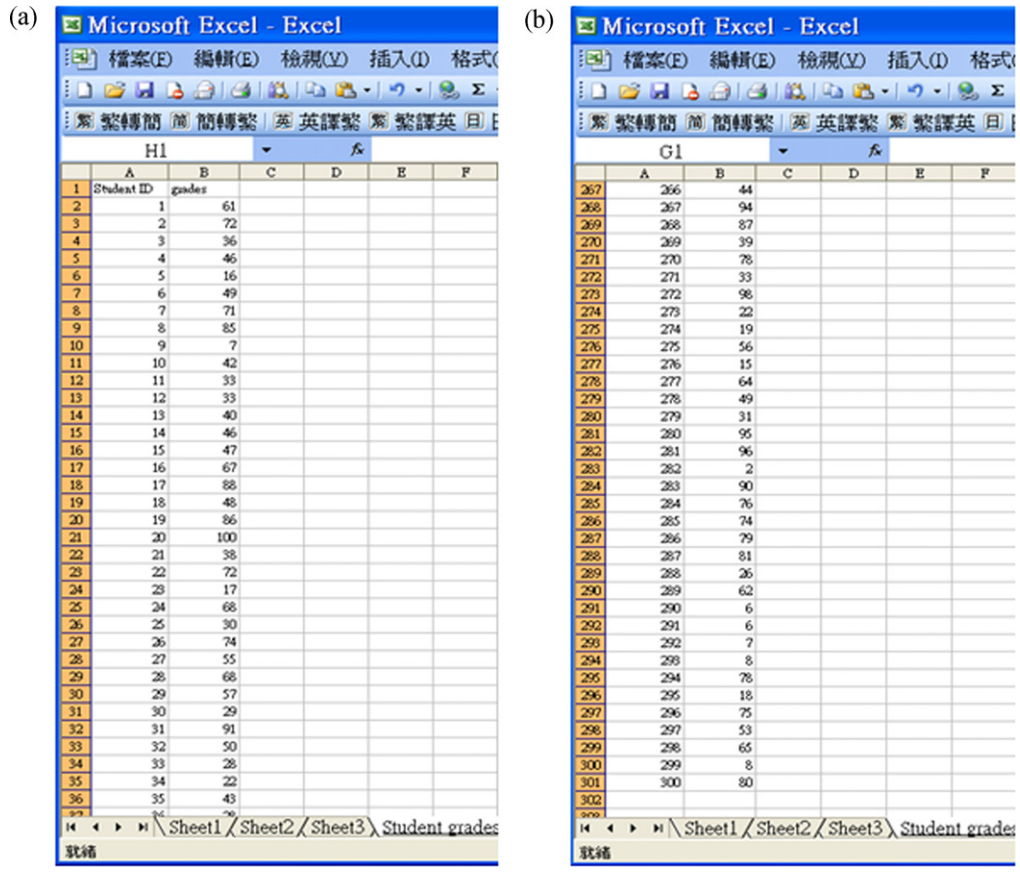


Fig. 3. A cover spreadsheet with 300 numeric items of students' test scores. (a) List of the first 36 items in the spreadsheet. (b) List of the last 34 items in the spreadsheet.

bits, in each segment group processed by the algorithm. It can be figured out that under the condition of using the same number of numeric items in a spreadsheet for data embedding, a larger k implies a larger embedding capacity but a coarser integrity check in the later process of self-authentication, while a smaller k means the reverse. There exists a tradeoff here.

Specifically, for instance, assume that 10 numeric items in a cover spreadsheet are to be replaced with secret shares, and a (k, n)-threshold secret sharing scheme with $k = 9$, $n = k + 1 = 10$ is adopted. In this case, the 9 coefficients $d, c_1, c_2, \ldots$, and c_8, with each being an m-bit segment of the secret message, form the coefficients of the 8-degree polynomial described by (3), and so provide $9 \times m = 9m$ bits as the embedding capacity by generating 10 secret shares and embedding them into the cover spreadsheet. As a comparison, under the same condition but with $(k, n) = (k, k+1) = (4, 5)$, a 3-degree polynomial including four m-bit coefficients is formed, providing a data embedding capacity of $4 \times m = 4m$ bits after 5 partial shares are generated and embedded. Therefore, if 10 number items of a cover spreadsheet is provided as well, then the 10 items can be used to embed 2 sets of 5 secret shares generated from 2 distinct segment groups in the secret message, yielding a total of $2 \times 4m = 8m$ bits as the data embedding capacity. As can be observed from the two cases, the former case provides a larger embedding capacity of $9m$ secret message bits yet with a segment group of $9m$ bits as the unit for later self-authentication. Contrastively, the latter case provides a smaller embedding capacity of $8m$ secret message bits but a finer authentication unit of $4m$-bit segment group in the secret message.

From the above discussions, a general conclusion about the data embedding capacity of the proposed method is made as follows: if I denotes the total number of numeric items in a cover spreadsheet available for embedding secret shares, then the embedding capacity C of the proposed method based on a (k, n)-threshold secret sharing scheme with $n = k + 1$ is:

$$C = \left\lfloor \frac{I}{n} \right\rfloor \times m \times k \tag{5}$$

where $\lfloor I/n \rfloor$ denotes the number of segment groups in the secret message M and m is the number of bits in a segment of M.

5. Experimental results

5.1. Experimental results using spreadsheets recording students' scores

A result of the experiments we conducted using the proposed method was based on the use of a cover spreadsheet recording 300 students' scores saved as an Excel file as shown in Fig. 3. Note that this is just an example; the type of cover spreadsheet and the content of it need not be restricted to be so.

The values of the involved parameters p, m and k in Eq. (3) of the Shamir method were set to be 101, 6, and 7, respectively. The value of the prime number p was taken to be 101 because it is the smallest integer larger than the full marks of 100 of the students' test scores. The value of $m = 6$ means that the length of each segment of the input secret message M was taken to be 6 bits, which satisfies the requirement of $2^m - 1 = 63 < p$ mentioned in Step 1 of Algorithm 3. And each message segment in M was transformed into an integer for use as one of the coefficients $d, c_1, c_2, \ldots, c_{k-1}$ in Eq. (3). As for $k = 7$, it means that the value n is $n = k + 1 = 8$ in the applied (k, n)-threshold secret sharing scheme, and that every 7 message segments in M are used as the coefficients $d, c_1, c_2, \ldots, c_6$ of the polynomial in Eq. (3). Then, a total of 8 (=7 + 1) secret shares were

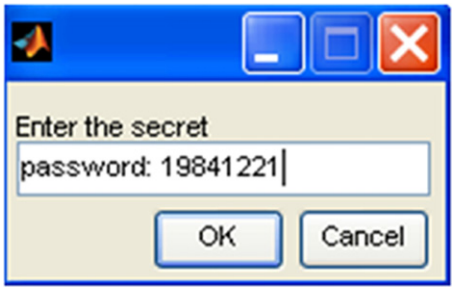


Fig. 4. A dialog for entering input secret message.

generated by Algorithm 3, yielding a self-authentication capacity of checking every 7 message segments in M.

Furthermore, as shown in Fig. 4, the input secret message M was taken to be the note: "password: 19841221". In this case, the 18 characters of the message were transformed into a binary string with $18 \times 7 = 126$ bits (7 bits per ASCII-coded character). The 126 bits then were divided into 3 segment groups with each group composed of 7 segments and each segment consisting of $m = 6$ bits. The three segment groups correspond to the following three message sections:

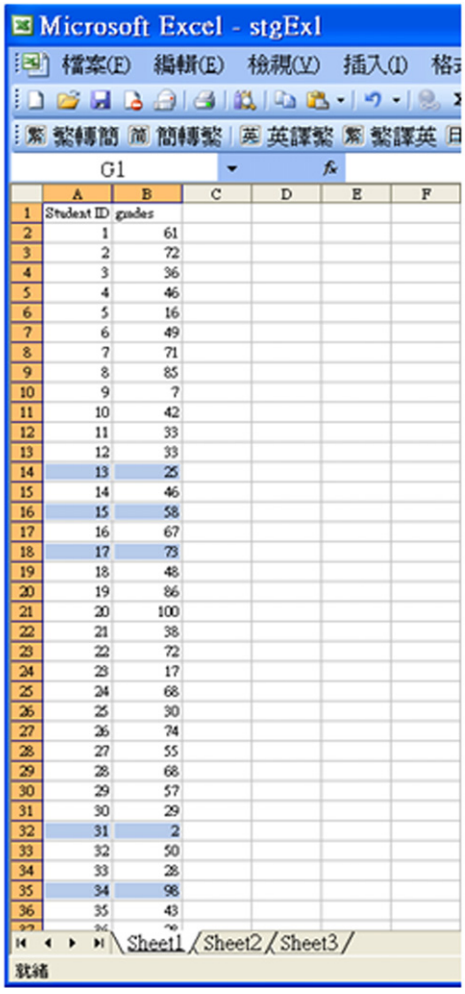
Group 1 : "Passwo"; Group 2 : "rd : 19"; Group 3 : "841221."

Totally, the 3 segment groups generated $3 \times 8 = 24$ secret shares which at last, by the use of a secret key, were randomly embedded into the cover spreadsheet to yield a stego-spreadsheet. We list the first 36 items in the stego-spreadsheet in Fig. 5(a), where items having been replaced with the secret shares are marked in blue. A list of the first 36 items in the cover spreadsheet is given in Fig. 5(b) for comparison.

If the stego-spreadsheet is intentionally modified illegally, Algorithm 4 will detect such tampering by the self-authentication operation (see Step 3). Besides, if some embedded secret shares survive the modification, Algorithm 4 can reconstruct the partially correct secret message from them by the recovery steps (Steps 2–4). Some experimental results of these functions are described now. Fig. 6 shows a modified stego-spreadsheet where items 16 through 26 were altered by replacing them with other numbers. Within the 11 modified items, items 15 and 17 include two embedded secret shares. The secret message extracted from such a modified spreadsheet using Algorithm 4 is shown in Fig. 7. As can be seen, segment groups 2 and 3 of the secret message were reconstructed correctly, while segment group 1 is authenticated to have been modified and marked by the algorithm with asterisk symbols "*."

In this case, the strategy of yielding a low embedding rate mentioned previously is used to achieve the goal of creating undetectability of the stego-spreadsheet. In order to ensure that this strategy works, the two-sample Kolmogorov–Smirnov test (KS test), which is a non-parametric statistical test and is useful to check whether two data samples come from the same probability

(a)



(b)

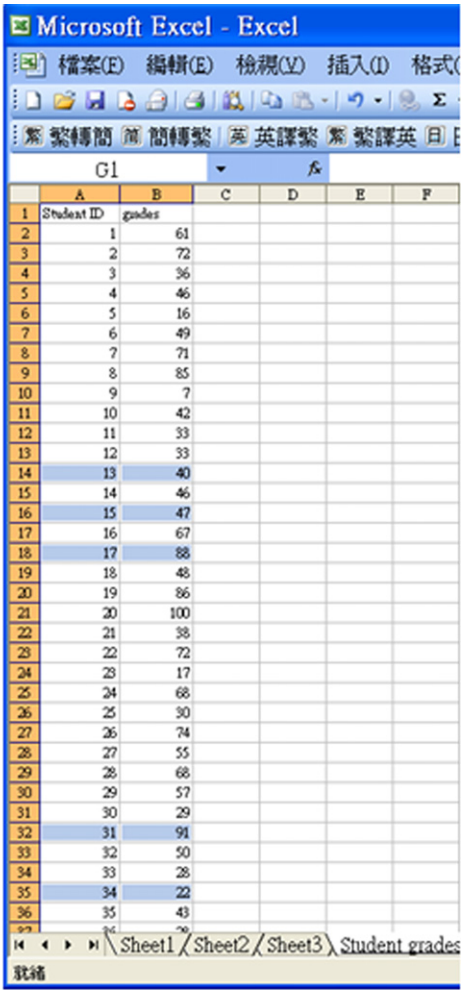


Fig. 5. Comparison of a cover spreadsheet and the stego-spreadsheet generated from it. (a) The stego-spreadsheet. (b) The cover spreadsheet.

Table 1
Experimental results of using strategy 1 with a cover spreadsheet with high scatter level of numeric data.

Scores1 (300 numeric items with variance 917.76 and size 25k)	# of replaced numeric items I	Resulting hypothesis (5%)	p value	Capacity = $\lfloor I/n \rfloor \times m \times k$ (bits)	Embedding bit rate per numeric item	Embedding bit rate
Embedding rate 5%	16	0 (cannot reject)	1	$2 \times 6 \times 7 = 84$	0.28 b	1/298
Embedding rate limit 50.67%	152	1 (reject)	0.0309	$19 \times 6 \times 7 = 798$	2.66 b	1/31

distribution, is used to quantitatively compare the probability distribution of numeric data in a stego-spreadsheet with that in a cover spreadsheet. The null hypothesis is that two data samples come from the same underlying distribution at the 5% significance level, and the alternative hypothesis is that they are from different distributions. The result of applying the test to the contents of the cover spreadsheet and the stego-spreadsheet shown in Fig. 5 is shown in Table 1 given below, in which the resulting hypothesis 0 means that the test cannot reject the null hypothesis, that is, a third party cannot think that the probability distribution of the stego-spreadsheet is different from that of the cover spreadsheet. The limit of the embedding rate at which the two-sample KS-test will reject the null hypothesis, according to our experiments, is 50.67% in this case. This means that the embedding rate should be smaller than 50.67% in order to keep the undetectability property of the stego-spreadsheet when a steganalyst has the information of the probability distribution related to the stego-spreadsheet.

How to choose an embedding rate which is secure against such a statistical test depends on the *scatter level* of the chosen numeric data of the cover spreadsheet. Here, the scatter level is computed as the variance of numeric data values. In terms of this parameter, three spreadsheets Scores1, Scores2, and Scores3 with the scatter level from high to low were tested further in our experiments using the same setting of parameters. Scores1 is just the one used in the first experiment mentioned above and the corresponding statistics is shown in Table 1. The results of using Scores2 and Scores3 are shown in Tables 2 and 3, respectively. From Table 2, the limit of the embedding rate using Scores2 is seen to be 26% which is lower than that using Scores1. As for Scores3, the corresponding limit of the embedding rate is down to be 6.04% as seen in Table 3. These experimental statistics indicate that the numeric data of a cover spreadsheet with a higher scatter level can yield a higher embedding rate without causing statistical anomalies. This fact can also be seen from the message embedding bit rate *per numeric item*, also shown in the tables. Specifically, the upper bound of the embedding bit rate per numeric item in Scores1 is 2.66 b, which is higher than those in Scores2 (1.36 b) and Scores3 (0.32 b).

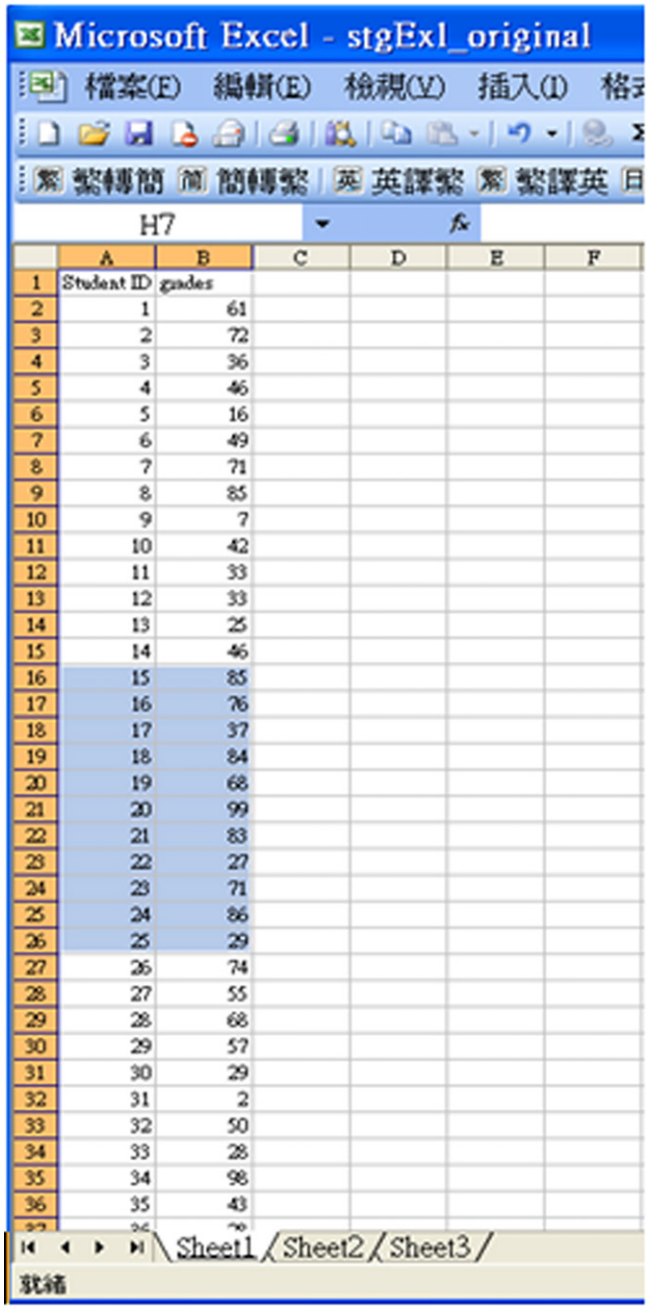


Fig. 6. An altered spreadsheet with fake items 16–26.

5.2. Experimental results using a spreadsheet of a financial statement

Another experimental result using the Microsoft Excel file of a financial statement of a company as the cover spreadsheet is shown in Figs. 8–11. Fig. 8 shows the cover spreadsheet with 32 candidate numeric items for data embedding. In this case, the strategy of choosing insignificant parts of numeric data in the cover spreadsheet for embedding secret shares is used to keep a low level of embedding strength for consideration of the undetectability of the generated stego-spreadsheet. Fig. 9 shows the input secret message which was transformed into 32 shares by Algorithm 3. Correspondingly, the decimal fractions of all of the 32 numeric items in the cover spreadsheet of Fig. 8 were used to embed the shares. Each share was transformed into two digits and embedded to the right of the decimal point of a numeric

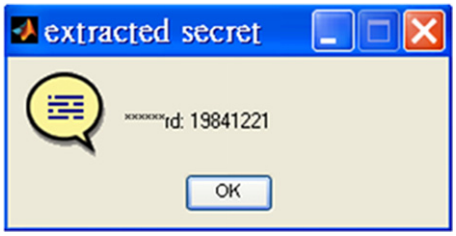


Fig. 7. An extracted secret message with a message segment retrieved from tampered items in the stego-spreadsheet marked by symbols "*".

Table 2
Experimental results of using strategy 1 with a cover spreadsheet with medium scatter level of numeric data.

Scores2 (1296 numeric items with variance 465.62 and size 105k)	# of replaced numeric items	Resulting hypothesis (5%)	p value	Capacity = $\lfloor l/n \rfloor \times m \times k$ (bits)	Embedding bit rate per numeric item	Embedding bit rate
Embedding rate 5%	64	0 (cannot reject)	0.9999	$8 \times 6 \times 7 = 336$	0.26 b	1/313
Embedding rate limit 26%	336	1 (reject)	0.049	$42 \times 6 \times 7 = 1764$	1.36 b	1/60

Table 3
Experimental results of using strategy 1 with a cover spreadsheet with low scatter level of numeric data.

Scores 3 (2250 numeric items with variance 283.11 and size 31k)	# of replaced numeric items	Resulting hypothesis (5%)	p value	Capacity = $\lfloor l/n \rfloor \times m \times k$ (bits)	Embedding bit rate per numeric item	Embedding bit rate
Embedding rate 5%	112	0 (cannot reject)	0.3557	$14 \times 6 \times 7 = 588$	0.26 b	1/53
Embedding rate limit 6.04%	136	1 (reject)	0.0383	$17 \times 6 \times 7 = 714$	0.32 b	1/43

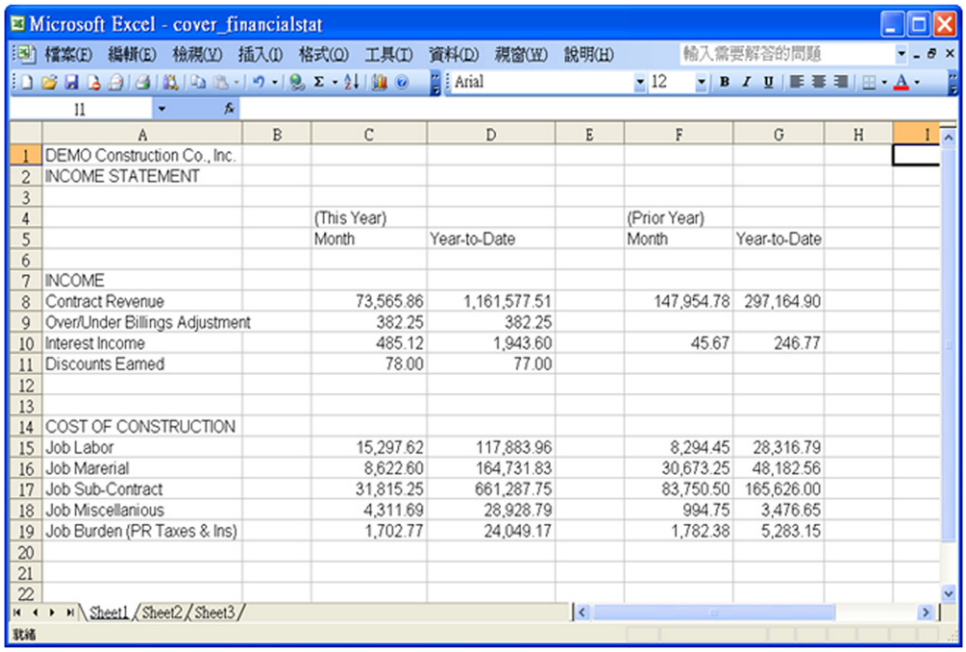


Fig. 8. A cover spreadsheet of financial statement with 32 numeric items.

Table 4
Experimental results of using strategy 2 for a cover spreadsheet of a financial statement.

Financial statement (32 numeric items and size 15k)	# of replaced numeric items	Result of hypothesis (5%)	p value	Capacity = $\lfloor l/n \rfloor \times m \times k$ (bits)	Embedding bit rate per numeric item	Embedding bit rate
Embedding rate 100%	32	0 (cannot reject)	1	$4 \times 6 \times 7 = 168$	5.25 b	1/89

item. The resulting stego-spreadsheet is shown in Fig. 10 which looks like a common spreadsheet. As done in the previous experiment, the two-sample Kolmogorov–Smirnov test was used, and the result is shown in Table 4 which supports the use of the strategy,

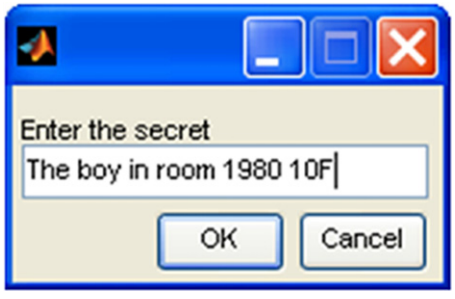


Fig. 9. A dialog with the input secret message.

accomplishing the goal of yielding statistical undetectability in the stego-spreadsheet.

Fig. 11 shows the stego-spreadsheet with 3 numeric items (highlighted) being modified. The secret message extracted from the modified stego-spreadsheet is shown in Fig. 12(b) in which the destructed part of the secret message is marked by asterisk symbols. As a comparison, the secret message extracted from the intact stego-spreadsheet shown in Fig. 10 is shown in Fig. 12(a).

5.3. Comparison with existing methods

For the purpose of presenting the contributions made in this study, a comparison of the capabilities of the proposed method with those of some existing covert communication methods is given in Table 5.

Most existing information hiding methods for covert communication (Bender et al., 1996; Wu and Tsai, 2003; Yang et al.,

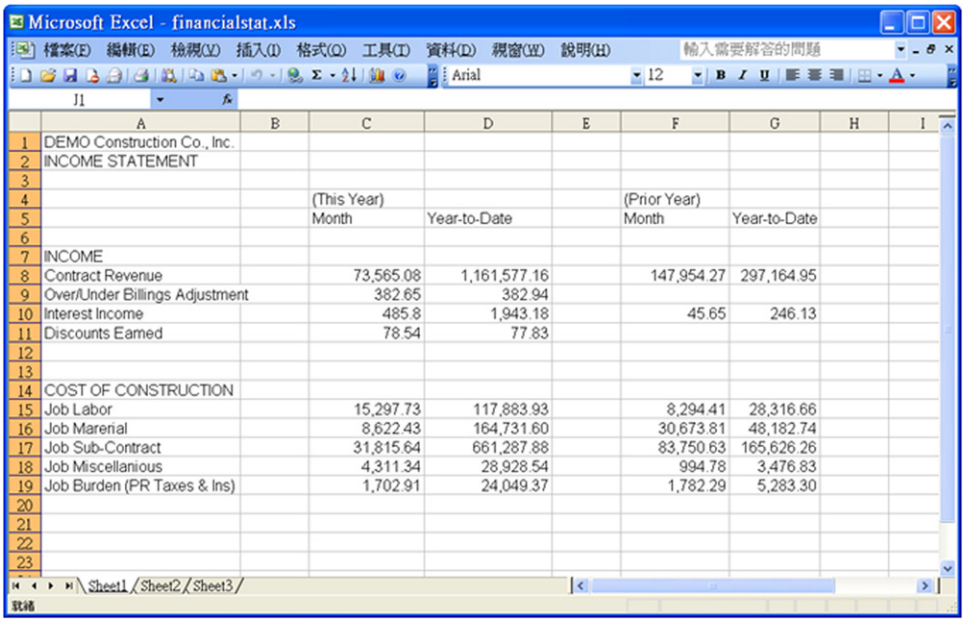


Fig. 10. A stego-spreadsheet in which the decimal fractions of the numeric items have been modified by embedded shares.

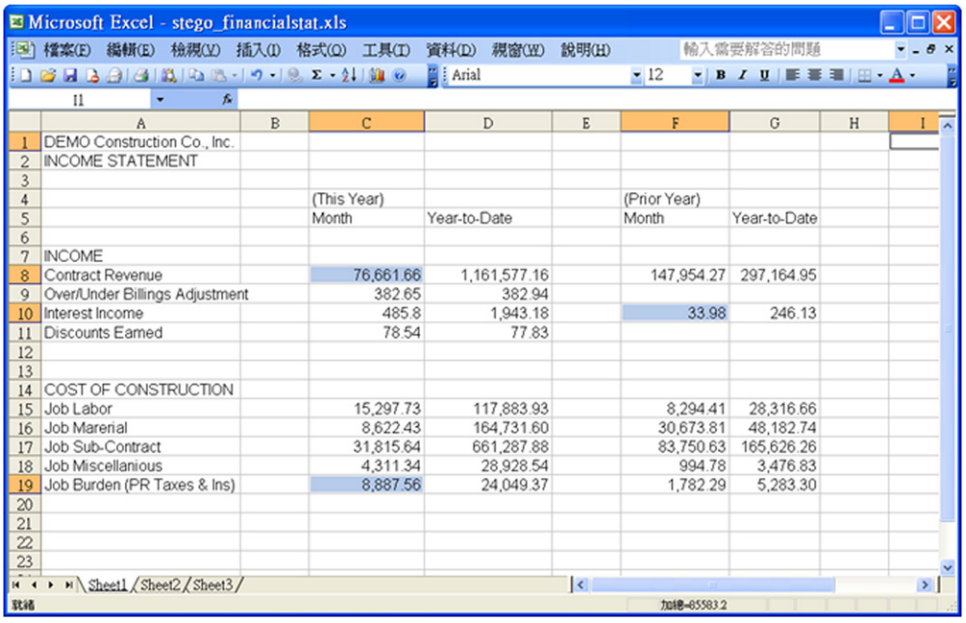


Fig. 11. A stego-spreadsheet with 3 numeric items (highlighted) being modified.

2008; Fridrich and Du, 2000; Lee and Tsai, 2010a,b; Zhong et al., 2007; Liu and Tsai, 2007) were developed based on the premise that an adversary always works in the passive mode. However, in practical covert communication, an active attack is defined as the action of an adversary who seeks to destroy the stego-content or to actively introduce subtle modifications to passing-by stego-objects between the two parties. Such an active attack may possibly cause a receiver to extract an incorrect secret message with no awareness.

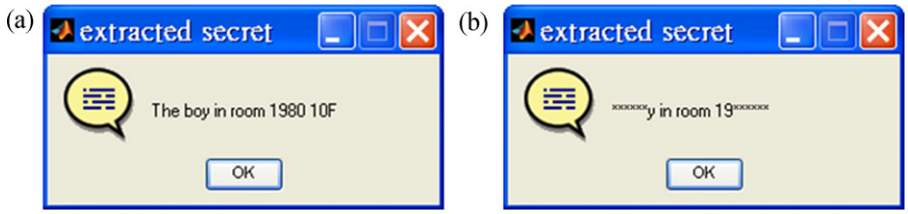


Fig. 12. The recovered secret message. (a) Message extracted from the intact stego-spreadsheet shown in Fig. 10. (b) Message extracted from the modified stego-spreadsheet shown in Fig. 11.

Table 5
Comparison of existing steganographic methods and proposed method.

	Manipulation of data embedding	Against active attack	Modification localization capability	Free from need of auxiliary information for message extraction	Keeping the size of a cover file after transformed into stego-version
Bender et al. (1996), Wu and Tsai (2003), Yang et al. (2008)	LSB-based (image)	No	No	Yes	Yes
Fridrich and Du (2000)	Parities of palette colors (image)	No	No	Yes	Yes
Lee and Tsai (2010a,b)	Certain ASCII codes(PDF)	No	No	Yes	No
Zhong et al. (2007)	Character space varying (PDF)	No	No	Yes	Yes
Liu and Tsai (2007)	Change tracking technique (MS word document)	No	No	No	No
Proposed method	Partial replacement of numeric items (spreadsheet)	Yes	Yes	Yes	Yes

Contrastive with the existing methods, the proposed method is the only one which has the self-authentication capability against active attacks and simultaneously takes the passive steganalytic attack into the consideration. Furthermore, the destructed part of a secret message can be localized precisely by the proposed method, that is, the proposed method has the capability of modification localization which is useful for verifying the integrity of the secret message in the proposed method.

Furthermore, auxiliary information for message decoding is required in some methods like (Liu and Tsai, 2007). Extra storage space is thus required to save the information for both parties in the communication, adding a burden to the system in practical use. Contrarily, like the methods of Bender et al. (1996), Wu and Tsai (2003), Yang et al. (2008), Fridrich and Du (2000), Lee and Tsai (2010a,b) and Zhong et al. (2007) the proposed method does not need any auxiliary information. In addition, the methods in Lee and Tsai (2010a,b) and Liu and Tsai (2007) increase the size of the generated stego-file due to the procedure of adding encoding codes or changing tracking records for data embedding. In contrast, the manipulation of substitution/replacement for data embedding used in methods of Bender et al. (1996), Wu and Tsai (2003), Yang et al. (2008) and Fridrich and Du (2000) as well as the proposed method keep the size of a cover file unchanged after it is transformed into a stego-version.

The embedding bit rate of the proposed method is comparatively smaller than that yielded by the methods of Bender et al. (1996), Wu and Tsai (2003) and Yang et al. (2008) using images as cover media. However, it is noted that these methods are vulnerable to the well-known RS steganalysis (Wang and Wang, 2004). This study aims at providing a new way of covert communication, and the issue of improving the embedding capacity deserves further investigation in the future.

On the other hand, for further resisting adversary's attacks on destroying the stego-content, secret shares may be spread over to different sets of spreadsheets to increase the possibility of retaining an enough number of shares for revealing the secret message. At last, it is worth to note that the proposed method can be equally applied to other document formats and not limited to only spreadsheet files, although the numbers appearing in such files maybe looks more natural than those appearing in other files.

6. Conclusions

A new covert communication method with a self-authentication capability via spreadsheets using Shamir's $(k, k+1)$ secret sharing scheme has been proposed in this study. The segment groups of a secret message are transformed into secret shares and then embedded as if they are part of the content in a cover spreadsheet, yielding a camouflage effect and generating a self-authentication capability. Each segment group of the secret message extracted from a stego-spreadsheet can be blindly authenticated by checking the results computed from all the $k+1$ possible combinations of k shares out of $k+1$ ones—if the resulting $k+1$ copies of the recovered secret are all identical to one another, then the stego-spreadsheet is decided to be intact. In case the stego-spreadsheet is authenticated to have been modified, the altered part of the hidden secret message may be identified, and the undamaged part recovered correctly. Experimental results have been shown to prove the feasibility and effectiveness of the proposed method. Derivations of the data embedding capacity and authentication precision have also been conducted, and discussions on the steganalysis issue included. Future studies may be directed to applications of the proposed method to multimedia protection in the field of fragile watermarking.

References

Bender, W., Gruhl, D., Morimoto, N., Lu, A., 1996. Techniques for data hiding. IBM Systems Journal 35 (3–4), 313–336.

Brassil, J.T., Maxemchuk, N.F., 1999. Copyright protection for the electronic distribution of text documents. Proceedings of the IEEE 87 (7), 1181–1196.

Chae, J.J., Manjunath, B.S., 1999. Data hiding in video. In: Proc. of 1999 IEEE International Conference on Image Processing, Kobe, Japan, pp. 243–246.

Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P., 2010. Digital image steganography: survey and analysis of current methods. Signal Processing 90 (3), 727–752.

Chen, L.S.T., Lin, J.C., 2010. Multithreshold progressive image sharing with compact shadows. Journal of Electronic Imaging 19 (1), 013003.

Fridrich, J., Du, R., 2000. Secure steganographic methods for palette images. In: Proc. of 3rd International Workshop Information Hiding, September 1999, Dresden, Germany. Springer-Verlag, Berlin, pp. 61–76 (also in Lecture Notes in Computer Science).

Gopalan, K., et al.,2003. Covert speech communication via cover speech by tone insertion. In: Proc. of the 2003 IEEE Aerospace Conference. Big Sky, MT, USA.

Lee, C.W., Tsai, W.H., 2010a. Authentication of binary document images in PNG format based on a secret sharing technique. In: Proceedings of 2010 International Conference on System Science and Engineering, Taipei, Taiwan, pp. 133–138.

Lee, I.S., Tsai, W.H., 2010b. A new approach to covert communication via PDF Files. Signal Processing 90 (2), 557–565.

Lin, C.C., Tsai, W.H., 2004. Secret image sharing with steganography and authentication. Journal of Systems and Software 73 (3), 405–414.

Liu, T.Y., Tsai, W.H., 2007. A new steganographic method for data hiding in Microsoft Word documents by a change tracking technique. IEEE Transactions on Information Forensics and Security 2 (1), 24–30.

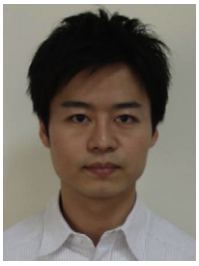
Pickholtz, R.L., Schilling, D.L., Millstein, L.B., 1982. Theory of spread spectrum communications—a tutorial. IEEE Transactions on Communications 30 (5), 855–884.

Provos, N., Honeyman, P., 2003. Hide and seek: an introduction to steganography. IEEE Security and Privacy Magazine 1 (3), 32–44.

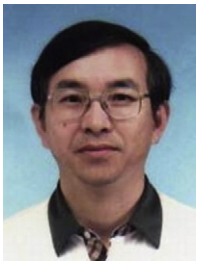
Shamir, A., 1979. How to share a secret. Communication of ACM 22 (11), 612–613.

Thien, C.C., Lin, J.C., 2002. Secret image sharing. Computers and Graphics 26 (1), 765–770.

Wang, H., Wang, S., 2004. Cyber warfare: steganography vs. steganalysis. Communications of ACM 47 (10), 76–82.
Wu, D.C., Tsai, W.H., 2003. A steganographic method for images by pixel-value differencing. Pattern Recognition Letters 24 (9–10), 1613–1626.
Wu, M., Yu, H., Gelman, A., 1999. Multi-level data hiding for digital image and video. In: Proc. of SPIE Photonics East, Boston, MA, USA, pp. 10–21.
Yang, C.H., Weng, C.Y., Wang, S.J., Sun, H.M., 2008. Adaptive data hiding in edge areas of images with spatial LSB domain systems. IEEE Transactions on Information Forensics and Security 3 (3), 488–497.
Zhong, S., Cheng, X., Chen, T., 2007. Data hiding in a kind of PDF texts for secret communication. International Journal of Network Security 4 (1), 17–26.



Che-Wei Lee receives the B.S. degree in civil engineering and the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2002 and 2005, respectively. He is a Ph.D. student in the Department of Computer Science at National Chiao Tung University since 2005. His research interests include information hiding, image processing, and video technologies.



Wen-Hsiang Tsai received the B.S. degree in EE from National Taiwan University, Taiwan, in 1973, the M.S. degree in EE from Brown University, USA in 1977, and the Ph.D. degree in EE from Purdue University, USA in 1979. Since 1979, he has been with National Chiao Tung University (NCTU), Taiwan, where he is now a Chair Professor of Computer Science. His current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications.