Оглавление

Заголовок оглавления оформляется стилем «Заголовок оглавления». Само оглавление добавляется при помощи меню Ссылки - Оглавление – Настраиваемое оглавление - OK. Обновлять содержимое оглавления можно клавишей F9.

[1 Заголовок 1 2](#_Toc112337255)

[1.1 Заголовок 2 2](#_Toc112337256)

[1.1.1 Заголовок 3 2](#_Toc112337257)

[1.2 Начало работы 2](#_Toc112337258)

[1.3 Общие замечания 2](#_Toc112337259)

[1.4 Оформление списков 3](#_Toc112337260)

[1.5 Оформление рисунков 4](#_Toc112337261)

[1.6 Оформление таблиц 4](#_Toc112337262)

[1.7 Оформление формул 5](#_Toc112337263)

[1.8 Ещё о ссылках и метках 5](#_Toc112337264)

[1.9 Оформление листинга 6](#_Toc112337265)

[1.10 Оформление библиографических ссылок 6](#_Toc112337266)

[Заключение 7](#_Toc112337267)

# Заголовок 1

## Заголовок 2

### Заголовок 3

#### Заголовок 4

Основным стилем большей части текста является стиль «Обычный».

## Начало работы

При создании нового отчёта настоятельно рекомендуется сделать копию настоящего документа и начать работать в нём, не удаляя настоящий текст (можно отделить эту часть от актуального текста пустой страницей). Это позволит сохранить в документе примеры использования всех стилей во всех необходимых на практике ситуациях. Таким образом, можно всегда посмотреть на пример оформления и скопировать этот пример в редактируемую часть текста.

## Общие замечания

Корректная работа с использованием данного шаблона должна обеспечиваться в Microsoft Word версии 2013 и новее. Не гарантируется корректность работы в LibreOffice, Google Docs, OpenOffice и пр. хотя она не исключена. Имейте в виду, что даже Microsoft Word, несмотря на свою богатую историю, достаточно глючная программа, которая может рушить форматирование при открытии файла на разных компьютерах и в разных версиях Word. Не рассчитывайте, что ваш документ не поплывёт при попеременном редактировании на разных устройствах в разных текстовых процессорах.

В данном шаблоне всё завязано на использование ограниченного числа стилей форматирования. Крайне нежелательно добавлять без надобности отдельные куски текста, отформатированные по-своему. При необходимости можно создать дополнительный стиль, если что-то здесь упущено. Старайтесь блюсти «стилевую чистоту» вашего документа.

Следует отметить, что в отчётах по курсовым проектам, практикам и в прочих учебных документах необходимо вставлять несколько вспомогательных страниц в начале, заполненных малосодержательной информацией, имеющей весьма пёстрое оформление (титульный лист, реферат, задание). Если пытаться объединять их с основным содержимым, это неминуемо спровоцирует обилие внестилевого форматирования текста и/или расширение числа стилей, которые в основном тексте не нужны. Поэтому можно поступить следующим образом: отделить начальные вспомогательные страницы от основного содержимого и поместить их в двух разных документах. При необходимости их можно объединить на уровне PDF-документа.

## Оформление списков

В шаблоне заведены стили для маркированного и нумерованного списков, а также многоуровневого. Автоматические списки – достаточно удобный инструмент (особенно нумерованный список), позволяющий избежать ручного редактирования. Однако с ним тоже изредка бывают проблемы при открытии на разных устройствах. Поэтому для списков, состоящих из 2-5 элементов, которые вы не планируете редактировать в дальнейшем, бывает целесообразным применять ручное оформление (с использованием стиля «Обычный»). При этом для маркированного списка начинаем каждую строку с «- » (после дефиса необходим неразрывный пробел), для нумерованного списка – с «#) » (где # – номер). Ниже приведены специальные стили, заведённые для списков, которые следует применять, если работа происходит на одной-двух проверенных машинах.

Маркированный список (стиль «Список маркированный»):

* item item item item item item item item item item item item item item item item item item item item item item;
* item;
* item.

Нумерованный список (стиль «Список нумерованный»):

1. item item item item item item item item item item item item item item item item item item item item item item;
2. item;
3. item.

Многоуровневый список в стандарте университета оформляется по-дурацки (см. ниже). Если есть возможность, лучше не использовать его.

1. буквенный 1;
2. буквенный 2:
   1. циферный 1;
   2. циферный 2;
3. буквенный 3.

## Оформление рисунков

Для рисунков проще всего скопировать рисунок, приведённый ниже, вместе с подписью и вставить его себе. Далее желательно сразу сменить номер. Для этого нужно выделить строку подписи и нажать F9. Ссылка на рисунок 1 оформляется при помощи меню Вставка – Перекрёстная ссылка – Рисунок. Там выбираете рисунок, на который ссылаетесь, и выбираете формат «Постоянная часть и номер». Чтобы исчезло автоматически добавленное вместе со ссылкой слово «Рисунок», кликните правой кнопкой мыши по номеру рисунка в ссылке и выберите меню «Коды/значения полей». Будет показан код ссылки. Отредактируйте этот код таким образом, чтобы он заканчивался на “ \h \# 0 ” (скопируйте текст в кавычках вместе с пробелами и замените конец имеющегося кода – он по умолчанию оканчивается на “ \h ”). Обратите внимание – рисунок и подпись к нему оформляются разными стилями.

4.emf(стиль «Рисунок»)

Рисунок 1 – Его подпись (стиль «Рисунок - подпись»)

## Оформление таблиц

Для вставки таблиц копируйте строку заголовка таблицы ниже и вставляйте себе. Далее желательно сразу сменить номер. Для этого нужно выделить строку и нажать F9. Ссылка на таблицу 1 оформляется при помощи меню Вставка – Перекрёстная ссылка – Таблица. Там выбираете таблицу, на которую ссылаетесь, и выбираете формат «Постоянная часть и номер».

Таблица 1 – Координаты центра эталонных участков (стиль «Таблица – заголовок»)

|  |  |  |
| --- | --- | --- |
| Эталонный участок | Географическая широта | Географическая долгота |
| № 1 | 53º 04′ 09,51″ с.ш. | 50º 41′ 38,12″ в.д. |
| (стиль «Таблица – текст внутри») |  |  |

После таблицы вставляйте одну пустую строку, как показано выше.

## Оформление формул

1. Формула без номера (стиль «Формула»):

где ниже приводится какое-то пояснение (оно оформляется стилем «После формулы»). Обратите внимание, что запятая в формуле выше является частью формулы.

1. Формула с пояснением на несколько строк:

где  – неизвестная переменная (стиль «После формулы»);

– известная переменная (стиль «После где»).

1. Формула с номером. Для создания такой формулы копируйте и вставляйте строку с формулой ниже целиком. Далее выделите вставленную строку и нажмите F9 для смены номера. Ссылка на формулу (1) оформляется при помощи меню Вставка – Перекрёстная ссылка – Формула. Там выбираете формулу, на которую ссылаетесь, и выбираете формат «Название целиком». Для формул настоятельно рекомендуется использовать встроенный редактор формул Microsoft Word версии 2007 и выше.

|  |  |
| --- | --- |
|  | (1) |

## Ещё о ссылках и метках

Инструментарий меток и перекрёстных ссылок гарантирует корректность номеров ссылок и возрастающий порядок номеров при перемещении частей документа в его пределах, а также при слиянии нескольких документов. В настоящем шаблоне метками оформлены формулы, таблицы и рисунки. Помимо этого, также можно использовать и автоматические ссылки на разделы и подразделы. Например, описание оформления формул представлено в подразделе 1.7 (ссылка добавлена при помощи меню Вставка – Перекрёстная ссылка – Заголовок). Обратите внимание, что бюрократы заголовки первого уровня называют разделами, второго уровня – подразделами, третьего уровня – пунктами, четвёртого – подпунктами. Однако лучше этими канцеляризмами не увлекаться: по возможности, желательно ограничиться терминами «раздел» и «подраздел».

## Оформление листинга

Листинг программы чётко не определён стандартом. Но можно его оформить следующим образом (стиль «Листинг»):

while b do Q

## Оформление библиографических ссылок

В стандарте университета присутствует неоднозначность касательно оформления библиографического списка: в тексте упоминается ГОСТ 7.0.5-2008, а в примере оформления список оформлен по ГОСТ 7.1-2003. Важно отметить следующие факты:

* оба стандарта визуально достаточно страшные, к тому же они не очень удобны для автоматического формирования ссылок средствами библиографических менеджеров;
* для обоих стандартов можно найти в интернете стили Zotero (если постараться). Для ГОСТ 2008 этот стиль более корректно формирует список литературы, для ГОСТ 2003 есть ошибки, которые нужно немного подчищать вручную;
* на практике в научном сообществе, да и в реальности университете в основном пользуются ГОСТ 2003;
* очень многие, в том числе и среди преподавателей, в этих ГОСТах не разбираются, не отличают их. Это может облегчить задачу студента.

Не стоит всецело полагаться на библиографические менеджеры при написании крупной работы. Могут возникнуть проблемы на разных компьютерах из-за некорректной синхронизации или иной версии программы. Из-за этого есть риск потерять ссылки. Как этого избежать? Использовать один компьютер; делать резервное копирование, использовать систему контроля версий или работать в облачном хранилище с поддержкой истории версий; дробить крупный документ на несколько. Кроме того, можно использовать гибридный вариант работы с Zotero: хранить коллекцию статей в нём, но в тексте использовать для ссылок текстовые метки, например [Автор, Год]. После написания документа автоматически экспортировать коллекцию Zotero в список использованных источников и вручную поменять метки. Минусы этого подхода очевидны; главный плюс – надёжность. Но для студенческих работ, когда количество публикаций в списке – 10-20, а количество ссылок на них – 20-30, вероятность проблем с Zotero невелика.

Примеры ссылок, оформленных по ГОСТ 2003 (они также отформатированы с использованием специального стиля «Список источников»):

1. Kandi, H. Exploring the learning capabilities of convolutional neural networks for robust image watermarking [Text] / H. Kandi, D. Mishra, S.R.S. Gorthi // Computers & Security. – 2017. – Vol. 65. – P. 247–268.
2. Zhu, J. HiDDeN: Hiding Data With Deep Networks [Text] / J. Zhu, R. Kaplan, J. Johnson, L. Fei-Fei // Lecture Notes in Computer Science. – 2018. – Vol. 11219. – P. 682–697.

Заключение

Реферат, введение, заключение, обозначения, сокращения, список использованных источников, приложения, оформляются стилем «Заголовок 1 без номера».

# Proposed method

We have proposed a Sudoku and MD5 hash algorithm based self-embedding fragile watermarking method to recover tampered images. Sudoku is a logic puzzle, usually played on a 9× 9 grid, based on a game called Latin Square [4]. The aim of the puzzle is to fill the 9 × 9 grid logically with 3 × 3 sub-blocks, columns and rows using the numbers 1 to 9 and the previously given numbers. In this proposed method, block-based image recovery data is embedded in four different regions of the image using the Sudoku algorithm. Then, the authentication data generated using the MD5 hash algorithm is embedded in the LSBs of the image blocks. The proposed algorithm comprises four main stages; sequentially, recovery bits embedding stage which is based on Sudoku; authentication bits embedding stage; tamper detection stage and recovery stage.

## Recovery bits embedding stage

This section comprises the generation of recover bits and embedding them into an image. Figure 1shows the block diagram of how the recovery bits are generated and how they are embedded into the image. This stage starts with setting the first and second LSBs of pixel values of the original gray-level image, I o, to zero. First LSBs are used for embedding authentication bits and second LSBs are used for storing recovery bits. The , I z,image is divided into twenty-five nonoverlapping blocks according to 5×5 Sudoku shown in the Fig. 1. Due to the random and single solution structure of Sudoku, resistance against tamper attacks is provided. Then, the image which is divided by Sudoku is grouped according to the Sudoku numbering, as formed 1, 2, 3, 4, 5. Thus, different regions of the image are used for embedding the recovery bits. Then, the sub-group which is grouped according to the Sudoku map is selected and the blocks in this sub-group are divided into non-overlapping 5 × 5 sub-blocks. For each sub-block, the average value of the sub-block is calculated to generate the recovery data, and this value is converted to a binary value to generate the recovery bits. A sample of generating the recovery bits for a sub-block is shown in Fig. 2. 24-bit quad recovery information for each sub-block is constructed by concatenating the first 6-bits of recovery bits of the other four sub-blocks as shown in Fig. 3, The generated 24-bit recovery bits are embedded into the second LSBs of the other sub-blocks in the group, excluding itself. The procedure of recovery bits embedding stage is represented as follows:

Set the first and second LSBs of original image, , pixel values to zero.

## Experimental Results and Analysis

Experiments were conducted on a group of gray-level images to verify the effectiveness of the proposed scheme. In the experiment, the sizes of the divided non-overlapping image blocks were 8 × 8, i.e., s = 8. The parameter k for block mapping was set to 2, which means two blocks map into one block and each block can also map into the other two different blocks. Obviously, the larger the parameter value of λ was set, the better performances of tampering detection and content recovery become, because larger embedding capacity can provide more valid reference bits and corresponding detailed information to recover the tampered blocks and more authentication bits can reduce the probabilities of false judgment for the tampered block and the reference bits. But, on the other hand, the larger the value of λ was set, more hiding payload were caused, which led to the degradation of visual quality for the watermarked image. During the watermark embedding procedure of our scheme, we set the parameter λ to 3 for sufficient embedding capacity, which means that three LSBs of each cover block were replaced with watermark bits and the five MSBs were preserved. Assume that the distribution of the three LSBs of cover image is uniform. Thus, the average energy of distortion caused by the watermark embedding for each pixel is:

|  |  |
| --- | --- |
| ,  where a - number of filled blocks;  b - number of empty blocks. | (2) |

The theoretical peak signal-to-noise ratio (PSNR) of the watermarked image can be calculated approximately using Eq. (9), which demonstrates the visual quality of watermarked image after embedding with λ = 3 is also satisfactory: 37.92 dB.

|  |  |
| --- | --- |
|  | (3) |

In order to meet the relationship in Eq. (2), Eq.(3), DCT coefficients of direct-current and lower frequencies representing the principle content in each block were used to produce reference bits, i.e., Q = 15. The assigned bits, i.e., r1, r2, …, r15, for the representation of the 15 DCT coefficients, i.e., c1, c2, …, c15, in each block were: 8, 6, 6, 4, 4, 4, 3, 3, 3, 3, 2, 2, 2, 2, 2, individually. Thus, the length R of the reference bits for each block was 54. The lengths L1 and L2 of the authentication bits for five MSBs and 54-bits reference bits were 40 and 22, respectively. Fig. 3 shows the standard cover image Lena sized 512 × 512 and its watermarked version. The PSNR value of the watermarked image is 37.98 dB. It can be observed from Fig. 3 that the visual distortion due to watermark embedding is imperceptible.



Figure 3 – Lena

For convenience of description, we divide the 512 bits of each 8 × 8 received watermarked block into three sets: 1) Φ1: 320 bits of the five MSBs, 2) Φ2: 40 bits of the authentication bits for the five MSBs, 3) Φ3: 152 bits of the reference bits and corresponding authentication bits for the two mapping blocks. The 192 bits in Φ2 and Φ3 come from the three LSBs of each block. We first conducted the testing of meaningful content tampering for the proposed scheme, in which the bits of Φ1, Φ2, and Φ3 for each modified block may all be changed randomly. Figs. 4-6 show the results of tampering detection and content recovery for our scheme. In Figs. 4-6, subfigures (a) show the original images Lena, Elaine, and Lake, and all of them are sized 512 × 512; subfigures (b) show the tampered, watermarked versions of (a) correspondingly; subfigures (c) are the tampered detection results, in which the black blocks indicate the tampered regions; subfigures (d) show the content recovery results for (b) correspondingly. The tampering percentages of subfigures (b) in Figs. 4-6 are 2.44%, 7.40%, and 9.67%, respectively, and the PSNR values of the recovered images in subfigures (d) of Figs. 4-6 are 37.39 dB, 36.11 dB, and 32.66 dB, respectively. Because the watermark

Table 1 – Results of tampering recovery for Scenario 1

|  |  |  |  |
| --- | --- | --- | --- |
| Number of tampered block pairs | Recovered percentage  Of [23] | Recovered percentage of proposed scheme | PSNR of recovered results by proposed scheme(dB) |
| 50 | 48.00% | 100% | 37.14 |
| 100 | 48.50% | 100% | 36.53 |
| 250 | 46.69% | 100% | 35.08 |
| 500 | 40.90% | 100% | 33.92 |
| 750 | 36.44% | 100% | 33.92 |
| 1000 | 31.80% | 100% | 32.22 |

Table 2 – Results of tampering recovery for Scenario 2

|  |  |  |  |
| --- | --- | --- | --- |
| Number of tampered block pairs | Recovered percentage of [23] | PSNR of the scheme [23] (dB) | PSNR of the proposed scheme (dB) |
| 50 | 48.00% | 37.66 | 37.74 |
| 100 | 48.50% | 37.48 | 37.70 |
| 250 | 46.69% | 36.70 | 37.61 |
| 500 | 40.90% | 36.15 | 37.45 |
| 750 | 36.44% | 35.59 | 37.42 |
| 1000 | 31.80% | 35.49 | 37.20 |

## Basic Principles and Characteristics of Semi-Fragile Watermark

Semi-fragile watermarking algorithms for image authentication focus on the ability of detection, location and recovery from tamper attacks carried out on images. Because the semi-fragile watermark has a certain degree of fragility, the algorithm can realize the image authentication according to whether the watermarking data is tampered with. When the host image suffers attacks, the watermarking information will make a corresponding change as well. However, for normal image-processing operations, the semi-fragile watermark also has a certain degree of robustness, so that the normal image processing operations and malicious tampering operations can be distinguished. The framework of the semi-fragile watermarking authentication system contains the watermark embedding process and the watermark detection process. The watermark embedding process is shown in Figure 1, and the watermark detection process is shown in (the dashed part may not be needed in some cases).

At present, a lot of semi-fragile watermarking algorithms for content authentication have appeared. In general, to achieve good performance, a semi-fragile watermarking algorithm must meet the following characteristics:

* Balance between robustness and fragility

On the one hand, after experiencing malicious tampering, such as collage attack, constant average attack, and statistical analysis attack, a watermark should be able to make an alarm response to show its fragility; on the other hand, after experiencing reasonable image-processing operations, such as JPEG compression, brightness adjustment, affine transform, filtering, and noise addition, a watermark should be able to accept them to show its robustness. Therefore, the watermark embedded into the semi-fragile watermarking algorithms must balance the robustness and the fragility well;

* Invisibility

After embedding watermark into the image, the quality of the image cannot have obvious degeneration, and the embedded watermark needs to be invisible to ensure that the original image is not damaged;

* Blind detection

Sometimes people cannot find the original image in many applications, and even if sometimes the original image is provided by a third party, people still cannot believe it. So this requires that content authentication algorithms can achieve the detection process without original images;

* Localization and recoverability

If the image has been maliciously tampered with, the algorithm should be able to accurately determine and locate the tampered regions, and then recover it using the recovery data;

* Security

Attacks on the security of watermarks are intentional, and the next step is to threaten the robustness of the watermark. Semi-fragile watermarking algorithms should be able to resist malicious tampering, and cannot be easily detected, imitated, tampered with or fabricated by unauthorized parties;

* Collision resistance

Means that for two different messages, it should be difficult to find the same digest;

* Preimage resistance

Given only a hash value (message digest), it should be difficult to find any message hashing (or preimage) that generates that digest;

* Second preimage resistance

Given some message, it should be difficult to find a different message having the same hash.

## Characteristics of fragile watermarking schemes

The characteristics of fragile watermarking schemes are imperceptibility, data payload, computational cost, blind detection, tamper detection, and tampered image recovery. These properties are as follows.

1. Imperceptibility: Imperceptibility implies that embedded watermarks should be invisible. This measures the similarity between the original image and the watermarked image. 19308 Multimedia Tools and Applications (2021) 80:19307–19333;
2. Data payload: Data payload refers to the number of bits that can be embedded in the cover image;
3. Computational cost: It refers to the amount of time taken for watermark generation, embedding, extraction, tamper detection, and recovery;
4. Blind detection: This implies that watermark extraction and tamper detection can be done in the absence of the original cover image;
5. Tamper detection: Fragile watermarking scheme should detect any change in the image content and able to locate the tampered regions precisely;
6. Tampered image recovery: Once the tampered regions are located, a fragile watermarking scheme should be able to recover the tampered regions to the original contents of the image.