ISSN:0975-9646

Preventing Data Leakage in Distributive Strategies by Steganography Technique

Narendra Babu.Pamula, M.Siva Naga Prasad, K.Deepthi

Computer Science And Enginnering Department,
V.KR.,V.N.B & A.G.K College Of Enginnering
Gudivada,Krishna(d.t),Andhra Pradesh,India-521301

Abstract – Today network has grown abundantly due to rapid growth in advancement in technology we are enjoying the services what it is provided. But at the same time we are losing secrecy of data. Sometimes sensitive data must be handed over to supposedly trust third parties. With the extensive application of database systems, the owners of the databases have urgent requirements to protect their copyright of databases. Some of the data is leaked and found in an unauthorized place the distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Although watermarking techniques have been successfully utilized for copyright protection of multimedia data, yet the research of database water-marking technique is still facing a lot of challenges due to the differences between the relational database and multi- media data. In previous paper focus on a model for calculating "guilt" probabilities in cases of data Leakage & also present algorithms for distributing objects to agents, The option of adding "fake" objects to the distributed set in a way that improves our chances of identifying a leaker. In this paper we focus on identifying by using Steganography techniques. By using Steganography techniques, observers cannot tell the difference between encoded images and their originals .we place Steganography constraints and in order to view the data by agents must has to satisfy the Steganography constraints. Distributor handles the Steganography constraints in between both user &agents.

Key Words: Steganography, Allocation strategies, data distributor, leakage, guilt agents.

INTRODUCTION

Steganography is the hiding of a message within another media. Nowadays; more and more data are sold and transmitted on the intemet. Databases are being used widely in many important fields, such as, banking and so on. With the fast growth of database business on the net, the data may be unsafe after passing through the unsecure network. The data purchasers may hesitate to buy the data service for the following suspicion. First, the data receiver may suspect that the data are tampered with by unauthorized person. Second, they may suspect the data received are not produced and provided by the authorized suppliers. Third, the suppliers and purchasers actually with different interest should have different roles of rights in the database management or using. So how to protect and verify the data becomes very important here. The purpose of this paper is to show how Steganography can be effectively used to transport sensitive data over the Internet in a secure fashion. A prime candidate for Steganography is the use of an image to conceal a hidden ciphered message. Steganography improves encryption and security by creating a medium in which sensitive data can be passed through prying eyes via a file, with- out alerting anyone that the transmitted file actually contains a message. Steganography should be implemented more in practice today as cyber-terrorism and data-theft becomes an everyday occurrence. Steganography is a relatively old technology, but is still very young in regards to modern usage. With the high security concerns of corporations and individual users alike, the secure transportation of data needs to be a viable option. There are numerous techniques and methods for Steganography, which have been highly researched and used in practice. This paper shows how well Steganography performs in real-world applications in regards to the transportation of sensitive data. In this paper uses encoded images at the user profile end and for agents distributor varies different encoded images. So it is easy to identify guilty agents by taking the history of accessing data randomly.

Background

The concept of hiding information in other content has existed for centuries; the formal study of information hiding is called Steganography. Steganography is the practical science of hiding information inside other media with the intention of giving the impression that no hidden data is present. Steganography is not a new technology, having been practiced for thousands of year dating back to early mapmakers. Cryptography has the goal of preventing the viewing of sensitive data by obfuscating the message so only the sender and recipient can view it. Steganography is intended to take cryptography to the next level by attempting to prevent the impression of the existence of any sensitive data. Steganography main goal is to avoid detection; to deny the existence of sensitive data inside the cover file. In the use of Steganography, a cover file and hidden file are used. It is assumed that any eavesdroppers will have no access to the original cover file in question. Steganography techniques try to change the original cover file as little as possible in terms of quality and file size, in order to create the strongest security environment possible. "In steganographic applications there are two levels of security. The first is not allowing an observer to detect the presence of a secret message. The other is not allowing the attacker to read the original plain message after detecting the presence of secret information." Common media for Steganography includes images (e.g.: JPEG, GIF, BITMAP, PNG), audio files (e.g.: MP3 and WAV), and even executable binaries (e.g.: EXE executables). Just about any file type that has slack or white space in it can be used for Steganography – however images are usually the typical medium used for Steganography purposes. In existing system Leakage detection is handled by watermarking e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks were initially used in images, video and audio data whose digital representation includes considerable redundancy. Watermarking aims to identify a data owner and, hence, is subject to attacks where a pirate claims ownership of the data or weakens a merchant's claims.

PROPOSED SYSTEM

The distributor's data allocation to agents has one Steganography Constraint and one objective. The distributor's Steganography constraint is to satisfy agents' requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to control an agent to not to make leaks any portion of his data. The Steganography constraint is not decoded because in one research a few stenographic images are decoded out of thousands so it is considered for sensitive data. The distributor may not deny serving an agent request and may not provide agents with different perturbed versions of the same objects. The detection objective is ideal and tractable. The main objective to maximize the chances of detecting a guilty agent that leaks all his data objects. In this paper we developed a model for assessing the "guilt" of agents is developed. The option of adding "Steganography" object to the distributed set is considered. If it turns out an agent was given one or more Steganography objects that were leaked, then the distributor can be more confident that agent was guilty. Sometimes data is leaked and found in unauthorized places.

SYSTEM ANALYSIS AND IMPLEMENTATION

Steganography can be implemented in various ways. "There are many different kinds of steganography, but all are based on finding unused space on paper, in sound, or in files in which to hide a message." Many algorithms have been developed to provide robust and secure steganography – each of which uses different embedding techniques. A common technique is Least-Significant- Bit (LSB) embedding. This technique hides data bits in the last two significant bits of an image pixel. For example, a 500x500 pixel image has 250,000 pixels. If a small 249 character null- terminated ASCII message is embedded, we would need approximate 1992 bits (249 * 8 bits per character) for storage of the string. By breaking up the bit pattern for each character into pairs, we would need 4 pixels per character to store the message by storing 2 bits per pixel. This means only 1000 pixels ((249+1)*4) would need changing in their LSB (from out of a total 250,000). The "+1" in the above calculation comes from the null character used to terminate the string. This has very little effect on the over- all image. Figure 1 shows the steganography process of the cover image being passed into the embedding function with the message to encode – resulting in a steganographic image containing the hidden message. A key is often used to protect the hidden message. This key is usually a password used by the decoding software to unlock the hidden message. Most steganography tools offer encryption of the hidden message before the embedding function is executed, so this key is also used to encrypt and decrypt the message before and after the embedding process.

More extensive and robust techniques are available for review, but they are beyond the scope of our intent in this paper. Many tools are available for the steganography of various media, including binary executables and MP3 audio files. In this paper we use the tools S-Tools and JPHIDE. Each of these tools is further discussed in the Methodology section of this paper presented to test subjects in pairs (one encoded file and its original). The subjects were then asked to analyse them and attempt to pick out the steganographic media from each pair. This study shows that, even when the original non-encoded file is available for comparison, steganographic files are not discernable by standard observation. This paper also shows how modern steganography detection techniques perform when put to the test.

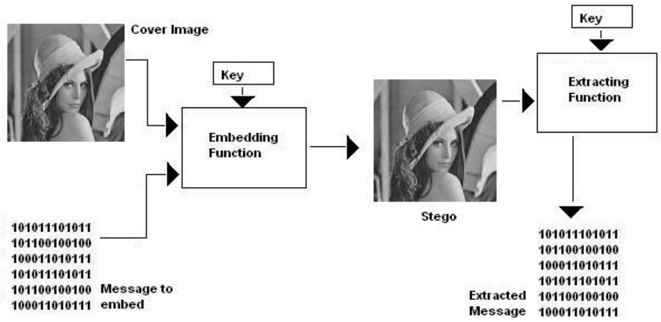
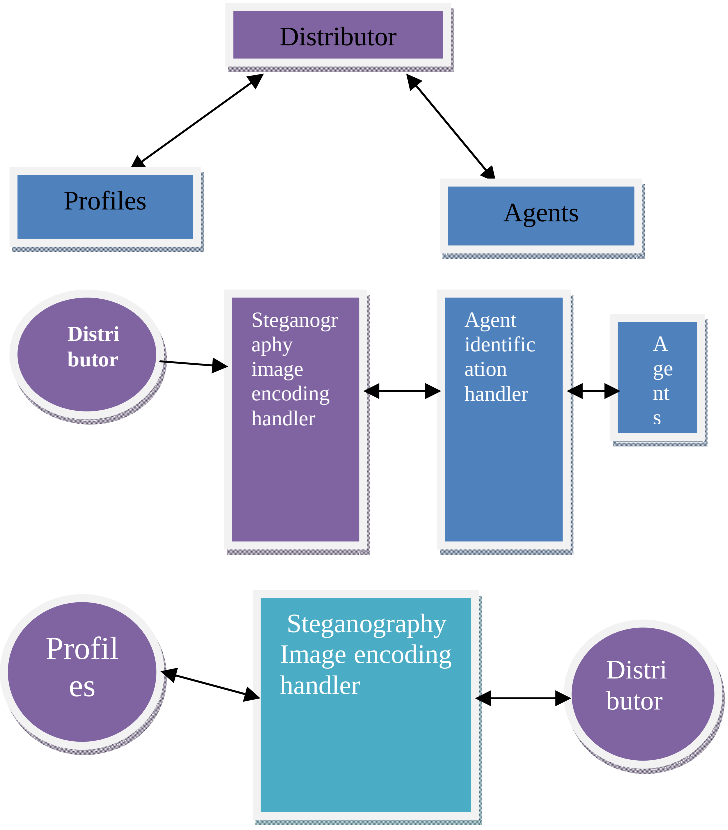


Figure1: stegnograpghy image encoding technique


Distributor
Profiles
Agents
Distributor
Steganography image encoding handler
Agent identification handler
Agents
Profiles
Steganography Image encoding handler
Distributor

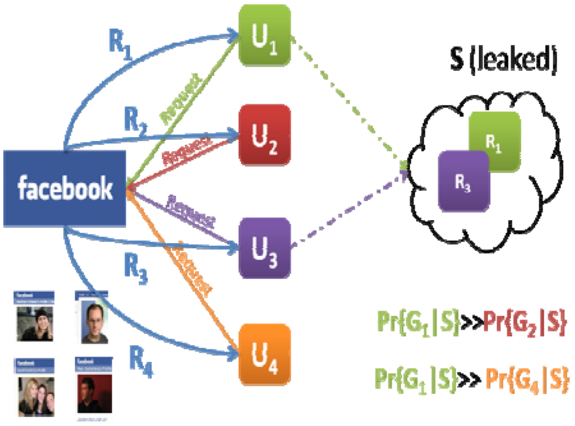
Figures illustrates how Data served to agents in secure fashion

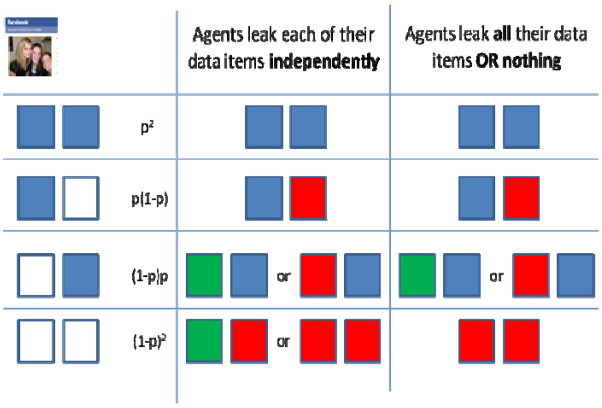
DATA LEAKAGE DETECTION

Entity: Distributor (social networking site)
Dataset: T (set of all profile)
Agents: Apps U1….Un, Dataset R1,…..,Rn (Ri: set of profiles who have added the application Ui)
Entity Leaker, Dataset: S (set of profiles leaked)

DISTRIBUTOR'S OBJECTIVE

- To achieve his objective the distributor has to distribute sets R_i, …, R_n that minimize
- $\sum_i \frac{1}{|R_i|} \sum_{j \neq i} |R_i \cap R_j|, \quad i, j = 1, \ldots, n$
- Intuition: Minimized data sharing among agents makes leaked data reveal the guilty agents





		Agents leak each of their data items **independently**	Agents leak **all** their data items **OR nothing**
🟦 🟦	p^2	🟦 🟦	🟦 🟦
🟦 ⬜	$p(1-p)$	🟦 🟥	🟦 🟥
⬜ 🟦	$(1-p)p$	🟩 🟦 or 🟥 🟦	🟩 🟦 or 🟥 🟦
⬜ ⬜	$(1-p)^2$	🟩 🟥 or 🟥 🟥	🟥 🟥

Analysis of detection of guilty agents in previous scenario

IMPLEMENTATION

The distributor maintains the entire database˙ The distributor registers the details of all agents**.** All Entities must select "New" when they enter and register for first time. The new register will enter details. The distributor validates the request and if he finds the agent is guilty, he adds fake objects. Choose "Add/Update" to make changes to an existing registration entered using the new Registration process or to add "tagging" information. Use "View" to query the contents of the registration database or download the database to local computer. It will be asked to Steganography login a logon username and password to validate login process. And it verifies the username and password with database. Once verified, it allows continuing the requesting process. The objects are serialized to prevent the data leakage. Only the valid user can unserialize the objects. In the implementation of the system we maintain Database maintenance of the following Agent maintenance.

a)
Registration
b) History
c) Detection of guilty
Agent d) Data Allocation
e) Addition fake object

DATABASE MAINTENANCE

Here the agent registration details are maintained and each agent has supplied an Steganography login and the sensitive data which are provided to agents are specified but without authentication they cannot view profiles. The designing of the whole database is done.

REGISTRATION

Here details of agents are registered and it collects the information about them like what are the sensitive data they want. At the same time ask to Steganography login.

HISTORY

Here the agent history is maintained like what all the details are given by distributor previously. It maintains entire details of the agent. To detect the guilty agents it checks the history and detects those agents who have fake details from the third party.

DETECTION RESULTS

The detection results provided an interesting picture, which also supports our idea that bigger is better in regards to cover file size. The results of our steganography detection are shown in Table 2.Our steganography detection tool, *StegDetect*, managed to detect some steganographic images, but not many. We embedded various messages ranging from 220 characters to 6KB in the 34KB JPEG image. *StegDetect* was unable to detect any messages in the altered files with its default settings. When *StegDetect's* sensitivity was increased (a feature that allows *StegDetect* to be "more accurate") nearly all steganographic images were detected minus the first 220-character file. This file could not be detected regardless of the sensitivity settings in *StegDetect*. We then tested *StegDetect* on various steganographic images built from the 174KB JPEG cover file. *StegDetect* enjoyed much less success on this test run – only one of the six-steganographic images generated was detected. Message sizes used in this run ranged from a 220-character text message to a 16KB image. The file detected was the steganographic image with a very large hidden message the largest *JPHIDE* could hide within the cover file (the 16KB image). Regardless of sensitivity settings, *StegDetect* could not find the other steganographic images with hidden messages of varying sizes (including a file that was 11KB in size). This leads us to believe that *StegDetect* works somewhat reliably on smaller images, but not on larger images – especially those with small hidden messages. To bypass *StegDetect*, only a small hidden message in a large cover file is needed. *StegDetect* only supports JPEG images and four steganography encoding pro- grams (all of which only use JPEG cover files). Given that *StegDetect* was the only testable software we could find, we conclude that steganography detection is very primitive in software form – and that it can be bypassed by simply using GIFS, BITMAPS, WAVS, and other media for steganography cover files.

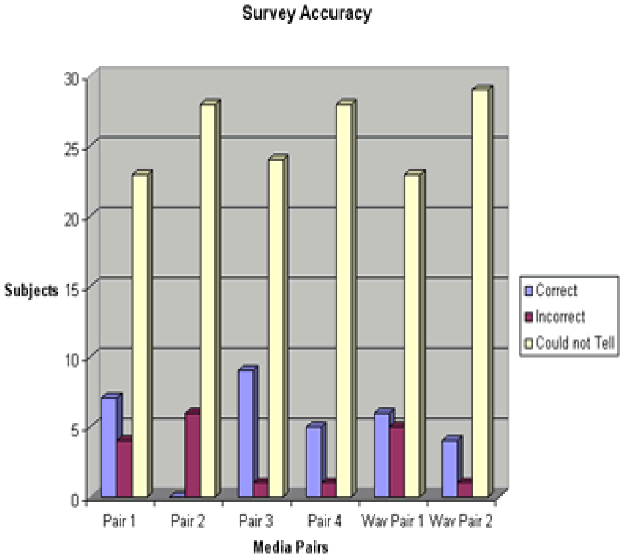


Fig: Graph version of survey results

CONCLUSION

The data steganography login distribution strategies improve the distributor’s chances of preventing a leaking. It has been shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data that agents must receive.

REFERENCES

[1] Panagiotis Papadimitriou, Hector Garcia- Molina (2010)‘Data Leakage Detection’, IEEE Transactions on knowledge and data engineering, Vol.22, No.3.

[2] P. Papadimitriou and H. Garcia-Molina, “Data Leakage Detection,” technical report, Stanford Univ., 2008.

[3] R. Agrawal and J. Kiernan (2002) ‘Watermarking relational databases’, In VLDB: Proceedings of the 28th international conference on Very Large Data Bases, pp.155– 166.

[4] P. Bonatti, S. D. C. di Vimercati, and P.Samarathi (2002)‘an Algebra for Compose Access Control Policies’- ACM Trans. Inf.Syst. Secure., Vol.5, No.1, pp. 1–35

[5] Y. Cui and J. Widom (2001) ‘Lineage Tracing For General Data Warehouse Transformations’ In the VLDB Journal ,pp. 471– 480.

[6] R. Agrawal and J. Kiernan, “Watermarking RelationalDatabases,” Proc. 28th Int’l Conf. Very Large Data Bases (VLDB ’02), VLDB Endowment, pp. 155-166,2002.

[7] Y. RichardWang, Stuart E. Madnick. A polygen model For heterogeneous database systems: the source tagging Perspective// Proceedings of the 16th International Conference on Very Large Data Bases, Brisbane, Queensland, Australia, February 5-9, 1990, 16:519-538.

[8] D. P. Lanter. Design of a lineage-based meta-data base For GIS. Cartography and Geographic Information Systems, 1991, 18:255-261.