Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

A survey on data leakage prevention systems

Sultan Alneyadi*, Elankayer Sithirasenan, Vallipuram Muthukkumarasamy

School of Information and Communication Technology, Griffith Sciences, Griffith University, Gold Coast, Queensland 4222, Australia

ARTICLE INFO

Article history:
Received 22 May 2014
Received in revised form
14 November 2015
Accepted 6 January 2016
Available online 19 January 2016

Keywords:
Data leakage detection
Data leakage prevention
Data security
DLPSs
Leaking channels
Content analysis

ABSTRACT

Protection of confidential data from being leaked to the public is a growing concern among organisations and individuals. Traditionally, confidentiality of data has been preserved using security procedures such as information security policies along with conventional security mechanisms such as firewalls, virtual private networks and intrusion detection systems. Unfortunately, these mechanisms lack pro-activeness and dedication towards protecting confidential data, and in most cases, they require predefined rules by which protection actions are taken. This can result in serious consequences, as confidential data can appear in different forms in different leaking channels. Therefore, there has been an urge to mitigate these drawbacks using more efficient mechanisms. Recently, data leakage prevention systems (DLPSs) have been introduced as dedicated mechanisms to detect and prevent the leakage of confidential data in use, in transit and at rest. DLPSs use different techniques to analyse the content and the context of confidential data to detect or prevent the leakage. Although DLPSs are increasingly being designed and developed as standalone products by IT security vendors and researchers, the term still ambiguous. In this study, we have carried out a comprehensive survey on the current DLPS mechanisms. We explicitly define DLPS and categorise active research directions in this field. In addition, we suggest future directions towards developing more consistent DLPSs that can overcome some of the weaknesses of the current ones. This survey is an updated reference on DLPSs, that can benefit both academics and professionals.

1. Introduction

Prevention of data disclosure to unauthorised entities is one of the main goals in information security. It continuously and rapidly drives both academic and industrial sectors to investigate, design and develop different security solutions to mitigate the risk of data leakage. However, preventing data leakage is not always possible because of the need to access, share and use information, which leads to inevitable release of confidential data. This revelation comes in the form of information leak, which might be the result of a deliberate action or a spontaneous mistake. Recent reports indicate growing concerns in government and business sectors as a result of data leakage. According to datalossdb (2015), in year 2014, about 50% of recorded data leakage occurred in the business sector, about 20% occurred in the government sector and about 30% occurred in the health and education sectors. Private users are also affected from data leakage, but it is hard to know the exact amount and severity of private data leakage. Although some reported leaks were not detrimental to organisations, others have caused several million dollars' worth damage. Business credibility is compromised when sensitive data such as future projects, trade secrets and customer profiles are leaked to competitors. Government data leaks may involve sensitive information about political relationships, law enforcement and internal security. A popular incident involving leaked sensitive government information was the release of the United States diplomatic cables by WikiLeaks. The leak consisted of about 250,000 United States diplomatic cables and 400,000 military reports referred to as 'war logs'. This revelation was carried out by an internal entity using an external hard drive and about 100,000 diplomatic cables were labelled *confidential* and 15,000 cables were classified as *secret* (Karhula, 2011). This incident received a high level of attention as the United States faced much criticism from governments and civil rights organisations worldwide. Another famous incident was the release of 77 million account details of Sony PlayStation network subscribers (Arthur and Stuart, 2011). The leak was due to an external intrusion, which forced the PlayStation network services to shut down for more than 24 days. This incident seriously impacted the reputation of Sony, receiving much criticism from users, and eventually led to a public apology from Sony's chief executive officer. One of the biggest recorded data leakage incidents was the release of names, email addresses and personal data of eBay customers (Wakefield, 2014), where around 145 million customers

* Corresponding author. Tel.: +64 24586815.
E-mail addresses: sultan.alneyadi2@griffithuni.edu.au (S. Alneyadi), e.sithirasenan@griffith.edu.au (E. Sithirasenan), v.muthu@griffith.edu.au (V. Muthukkumarasamy).

http://dx.doi.org/10.1016/j.jnca.2016.01.008
1084-8045/© 2016 Elsevier Ltd. All rights reserved.

were affected severely disrupting the business. These kinds of incidents can cause major financial losses and severely damage an organisation's reputation.

Driven by the need to address such serious issues, security experts endeavour to develop various security measures. Systems such as firewalls, intrusion detection systems (IDSs) or intrusion prevention systems (IPSs), and virtual private networks (VPNs) have been introduced over the past three decades. These proven systems can perform satisfactorily if the data to be protected is well defined, structured and constant. However, using these measures to protect evolving (i.e. edited, differently tagged or compressed) confidential data can be naive. For example, a firewall can block access to a confidential data segment using simple centralised rules; however, the same data segment may be accessible through other means such as an email attachment or instant messaging (IM). Thus, conventional security measures (i.e. firewalls, IDSs, VPNs) lack persistency and understanding of data semantics. To overcome this deficiency, a new direction for data protection was considered leading to the introduction of data leakage (loss) prevention systems (DLPSs). DLPSs are especially designed systems that have the ability to identify, monitor and protect confidential data and detect misuse based on predefined rules. The DLP field is considered relatively new compared with conventional security solutions. Moreover, to many academics and security practitioners, the field is indistinguishable because adequate research, surveys or both are lacking at present.

Motivated by the significance of the DLP field of study and the need for better understanding of current and future DLP trends, we present this survey paper. This paper contributes to the DLP field by explaining the DLP paradigm, including data states and deployments. Further, it identifies the challenges facing DLPSs. Moreover, it comprehensively gathers, categorises, discusses and compares the current DLP methods in industry and academia. It also lists and discusses DLP analysis techniques; and presents future DLPS trends.

This paper is structured as follows. Section 2 discusses the DLP paradigm. Section 3 describes the challenges facing DLPSs. Section 4 categorises the current DLP methods and discusses the advantages and disadvantages of each method. Section 5 explains the DLPS analysis techniques. Section 6 suggests future DLPS trends. Section 7 discusses the survey limitations. Section 8 concludes the survey paper.

2. Data leakage prevention

A number of attempts to study and define the area of data leakage prevention have been made in both academia and industry. These attempts discuss DLPSs from differing perspectives because DLPSs are still new and there is no concrete agreement on a common definition yet. Both academics and practitioners are using various names for DLPSs, such as data loss/leak prevention, information loss/leak prevention, extrusion prevention and content monitoring and filtering/protection (Mogull, 2010).

In academia, some researchers have provided a broad idea about the DLP research area. For example, a review paper by Raman et al. (2011) discussed the importance of the DLP research area and suggested that more attention be paid to it. The authors mentioned common DLP approaches and associated problems. In addition, they suggested new directions for future work, and introduced text clustering and social network analysis as future solutions for the problem. A more comprehensive survey on DLP was presented by Shabtai et al. (2012). The authors define a DLPS as ‘a system that is designed to detect and prevent the unauthorised access, use, or transmission of confidential information’ (p. 10). Their survey describes taxonomy of data leakage prevention solutions along with commercial and academic examples. Academic DLP methods are categorised into misuse detection in information retrieval systems/database, email protection, network/web-based protections, encryption and access control, data hidden in files and honeypots/honeytokens (p. 22). Data leakage/misuse scenarios, case studies and future trends are also given in this survey.

Professional and industrial institutes have also put effort into addressing the DLP area, including SANS, Securosis and ISACA. SANS presented a white paper (Kanagasingham, 2008) that provides a brief history about DLP solutions and how they fit within other network security technologies. Mogull (2010) from Securosis presented a white paper on understanding and selecting a DLP solution. The paper discusses the DLP market, in general, and the difference between a DLP feature and a DLP solution. It also considers the confusion surrounding the definition of DLPS and the variation in commercial products among vendors, which has resulted in the same product having many different names. Mogull defines DLPSs as ‘products that, based on central policies, identify, monitor, and protect data at rest, in motion, and in use, through deep content analysis’ (p. 5) and explains the differences between content and context analysis, suggesting that the former is more promising than the latter. Finally, the paper provides a summary of the strengths and weaknesses of the current content analysis approaches, such as rule-based or regular expressions, fingerprinting, exact file matching, partial document matching and statistical analysis.

ISACA’s (2010) white paper discusses DLPSs from a management point of view. It suggests that implementing a DLPS must be thoroughly planned and studied in terms of the need, the size and the aim of the organisation. The paper explains that unplanned implementation can defeat the purpose of using a DLPS in the first place. For example, if an organisation is using a DLPS to avoid business loss, business can be disrupted by wrong implementation of a DLPS. Wrong implementation includes hindering workflow by extensive traffic inspection and weak integration with other security mechanisms. The paper also discusses the many challenges that must be addressed before using a DLPS to ensure an organisation is ready to use it. The specific challenges vary among organisations, depending on the nature of the business and the volume of transactions.

2.1. Data leakage prevention systems

Data leakage (or data loss) is a term used in the information security field to describe unwanted disclosures of information. This problem is mitigated by using different DLP methods and techniques, including both administrative and technical approaches. In this paper, we define DLPSs as designated analytical systems used to protect data from unauthorised disclosure at all states using remedial actions triggered by a set of rules. This definition contains three main attributes that distinguish DLPSs from conventional security measures. First, DLPSs have the ability to analyse the content of confidential data and the surrounding context. Second, DLPSs can be deployed to provide protection of confidential data in different states, that is, in transit, in use and at rest. The third attribute is the ability to protect data through various remedial actions, such as notifying, auditing, blocking, encrypting and quarantining. The protection normally starts with the ability to detect potential leaks through heuristics, rules, patterns and fingerprints. The prevention then happens accordingly.

DLPSs differ from conventional security controls such as firewalls, VPNs and IDSs in terms of dedication and proactivity. Conventional security controls have less dedication towards the actual content of the data. They might block users' access to data for the sake of sensitive data protection in the case of firewalls, or simply encrypt all the traffic, as in the case of VPNs, which might include both sensitive and non-sensitive data. Moreover, most security systems lack proactivity because they normally work under pre-defined rules. This can be a major drawback when working in a rapidly changing environment. However, some security measures, such as anomaly-based IDSs, can be proactively triggered when certain criteria are met. These systems mainly focus on the metadata (context), such as size, timing, source and destination, rather than the sensitivity of the content. Likewise, context-based DLPSs focus on the context surrounding confidential data to detect any potential leaks. Content-based analysis DLPSs are more common than and preferable to those that are context based, since it is more logical to focus on the protection of the data itself than on the surrounding context (Mogull, 2010).

A typical content-based DLPS works by monitoring sensitive data in its repository or on the go, mainly by using regular expressions, data fingerprinting and statistical analysis. Regular expressions are normally used under a certain rule such as detecting social security numbers and credit card numbers. The problem with DLPSs using regular expressions analysis is that they offer limited data protection and have high false positive rates (Mogull, 2010). For example, it is an easy task to detect and prevent the leakage of a 'project name' through emails, by using a rule that prevents emails containing that specific name from being sent out. However, it is difficult to prevent the leakage of the project's vast details. In addition, if the rule is continually active, a regular email can be blocked if the same project name is used in another context. DLPSs using data fingerprints have better coverage for sensitive data because they have the ability to detect and prevent the leakage of a whole document or parts of a document. However, traditional fingerprinting can lose track when the sensitive data is altered or modified. This happens because traditional hashes that are used to generate data fingerprints, such as MD5 and SHA1 (Shapira et al., 2013), are susceptible to change. A tiny change in the data can result in a totally different fingerprint every time it is hashed. This can lead to data bypassing the DLPS, and thus data can be leaked. This problem can be partially solved by using multiple data hashing, whereby the original data is divided into smaller parts, that is, paragraphs and sentences, and each part is hashed separately (Kantor et al., 2009). This can ensure that parts of the original data fingerprints are retrievable. Nevertheless, these smaller fingerprints are also susceptible to change and the tiniest change can make the method ineffective. More advanced approaches try to overcome this problem by using similarity digests (Roussev, 2010), implementing Rabin fingerprinting (Shu and Yao, 2013) and using piecewise hashing (Kornblum, 2006). However, these solutions have limited capabilities and can be affected by various text obfuscations.

Although not widely used in DLPSs, statistical analysis can work in a fuzzy environment where the sensitive data is not well structured and the data semantics are distributed over a large corpus. The main advantage of such a technique is the ability to identify sensitive documents even after extreme modification. In particular, DLPSs with statistical analysis capabilities can use machine learning algorithms or Bayesian probability to identify altered documents. They can also use text-clustering techniques to construct scattered traces of sensitive data. Fig. 1 illustrates a simplified deployment of a DLPS, where the DLPS attributes are highlighted in the three main boxes.

2.2. Data states

Data states, as shown in Fig. 2, include data 'in transit', 'in use' and 'at rest'. Data in transit is the data being transmitted from one node to another. This type of data travels internally between nodes within the same network or externally between nodes that belong to different networks. Data in use is the data that is accessible to the user in the forms of documents, emails and applications. This type of data appears in plain text, so it can be easily interpreted and processed. Data at rest is the type of data that is stored in repositories. It consists of application databases, backup files and file systems. It is normally protected by strong access controls, including physical and logical mechanisms.

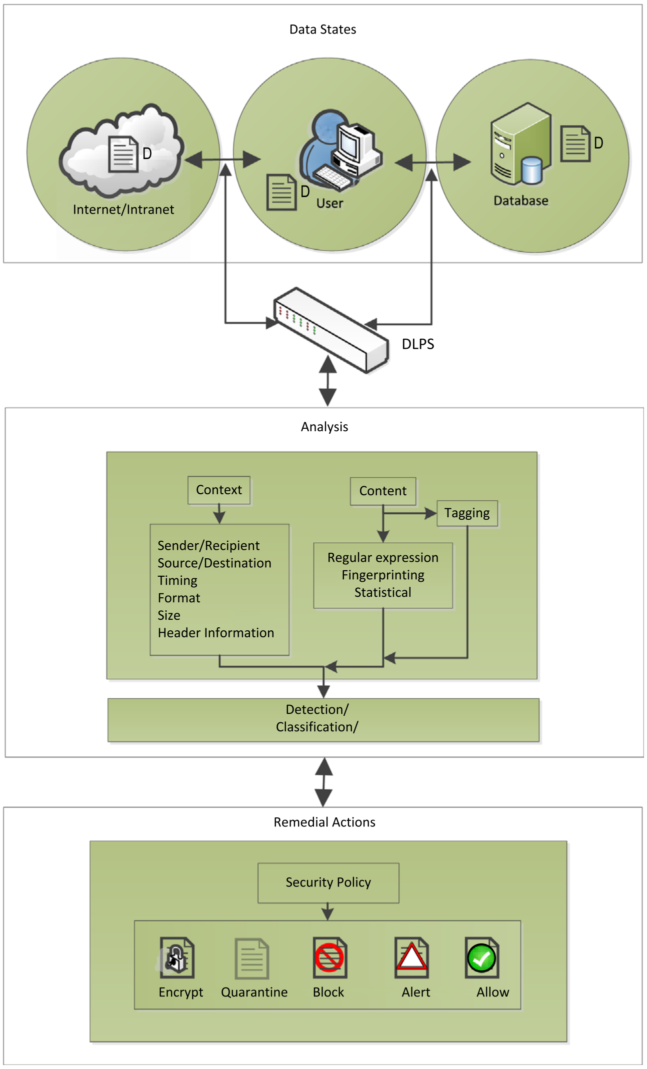

Fig. 1. Typical data leakage prevention system deployment within a network.

2.3. Deployment

Depending on the targeted data for protection, DLPS deployment can take many forms. For example, protecting "in use" data requires built-in software that acts like a DLP agent on endpoints. This agent is responsible for disabling or enabling access to applications that deal with confidential data. It is also responsible for blocking confidential data transfer through portable media, that is, CDs, USB drives and memory cards. Furthermore, it restricts copying, pasting and editing of confidential data as well as restricts making hard copies through printers. Last but not least, it audits all activities related to confidential data access. (Hackl and Hauer, 2009; McAfee, 2015).

For "in transit" data, DLP appliances are normally used. They come with special processing capabilities to handle large amounts of data. This type of DLPS is responsible for inspecting all outbound traffic for confidential data. It also acts as a proxy server when accessing some applications with confidential data. Moreover, it proactively reports

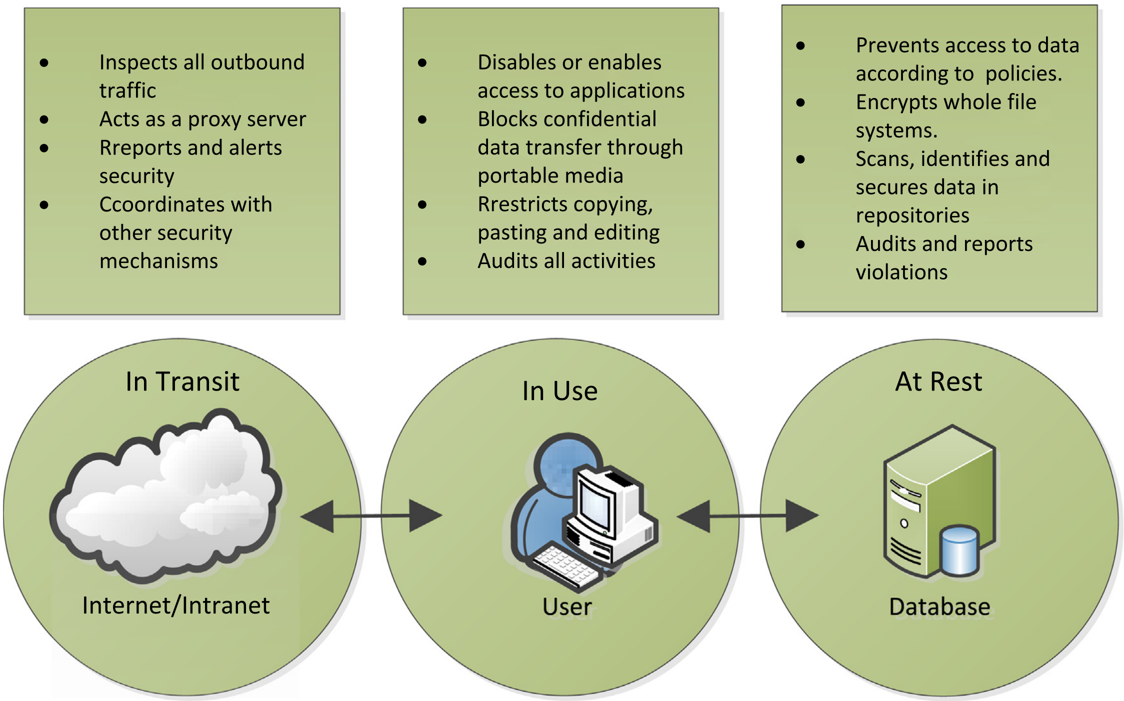
Inspects all outbound traffic
Acts as a proxy server
Rreports and alerts security
Ccoordinates with other security mechanisms
Disables or enables access to applications
Blocks confidential data transfer through portable media
Rrestricts copying, pasting and editing
Audits all activities
Prevents access to data according to policies.
Encrypts whole file systems.
Scans, identifies and secures data in repositories
Audits and reports violations
In Transit
Internet/Intranet
In Use
User
At Rest
Database

Fig. 2. Different data states.

and alerts security administrators and users about potential data leaks. Finally, it coordinates with other security mechanisms such as Secure Sockets Layer (SSL) proxies and network firewalls. (Mogull, 2010).

DLPSs that deal with data "at rest" are normally focused on protecting known data. The protection comes in the forms of preventing access to data based on predefined security policies. Also, this type of DLPS helps in protecting data at rest by encrypting entire file systems. Furthermore, it scans, identifies and secures confidential data in data repositories while auditing and reporting security policy violations. (Websence, 2015a and 2015b).

3. Challenges in data leakage prevention systems

Like any other security mechanism, DLPSs face many challenges when protecting confidential data. From the review conducted on both the academic and the industrial DLP solutions, seven main challenges were identified. To implement a successful DLP solution, these challenges must be considered and addressed adequately. The following subsections discuss the security challenges facing DLPSs.

3.1. Leaking channels

In order to access and share data between entities, intermediate channels must be available. In normal cases, these channels are legitimately used for data exchange. However, the same channels can be associated with confidential data leakage. Moreover, if there is a need for sharing data, some or all of these channels must be kept open. Fig. 3 shows common leak channels; some of the channels are easy to manage, whereas the others require a substantial effort to be fully secured.

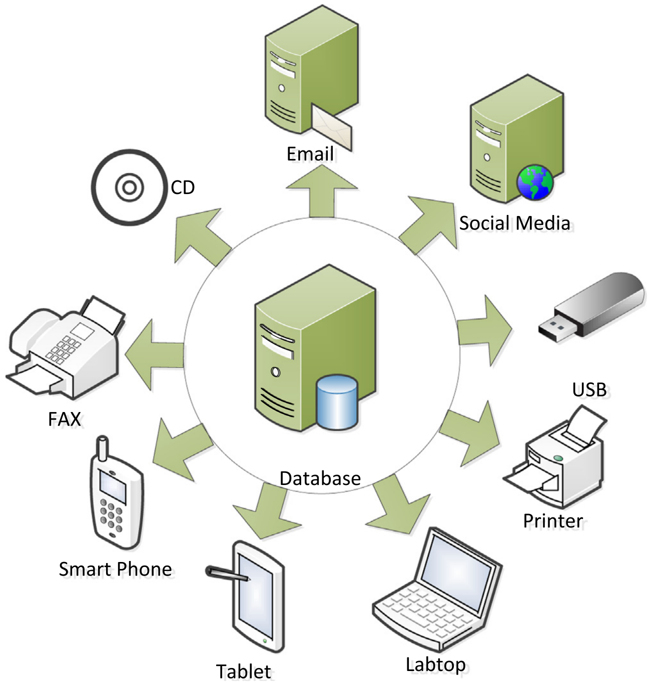

Fig. 3. Examples of data leaking channels (Olzak, 2010).

For data "in use" and "at rest", confidential data can be leaked through channels such as USB ports, CD drives, web services and printed documents. Data leaks through USB ports and CD drives can be mitigated through host DLPSs, but this is not enough since other leak channels, such as emails and IM are always available (Fabian, 2007). In addition, logical access rights can be imposed on users when accessing confidential data, but the same data might be physically accessible in forms of printable documents. Further, leaking channels associated with data "in transit", such as web services and file sharing, might be extremely challenging, since these channels may be business prerequisites. To ensure maximum security for data passing through these channels, extensive traffic filtering must be carried out. Nevertheless, workflow and production can then be hindered. DLPSs should deliver the desirable security for data "in transit" without affecting the interconnectivity between different domains.

In today's information greedy world, technological advances are inevitable. This will always entail an increase in the capacity of existing channels or even in the development of new communicating methods, which can in turn introduce new challenges in the form of new leaking channels.

3.2. The human factor

It is always difficult to predict human behaviour because it is influenced by many psychological and social factors. Many human actions are affected by subjectivity in making decisions, such as defining the secrecy level of data, assigning access rights to specific users and calibrating a detection threshold of a DLPS. In addition, conformity with the organisation's security policy is not always guaranteed, even with strict regulations and guidelines. These factors, along with many others, such as the influence of relationships, can present a major challenge to DLPSs. Most human interactions with information happen when the data is "in use"—typically by a user with an endpoint terminal; therefore, most data leaks happen in this state. Further, many DLPs tend to apply constraints on the user's ability to leak data such as disabling USB ports, CD drives and removable media. However, users can circumvent these techniques, even with a limited IT background. For example, one of the most popular methods for bypassing these constraints is to share access privileges among users, either intentionally or through social engineering (Orgill et al., 2004). Moreover, users can physically copy or take screenshots of sensitive documents or even use a mobile phone camera to take pictures of classified documents. As long as the human factor is present, there will always be challenges for DLPSs.

3.3. Access rights

It is important for DLPSs to be able to distinguish between different users based on their privileges and permissions. Without a proper definition of access rights, DLPSs cannot decide whether or not the data is being accessed by a legitimate user. Therefore, access control systems play an important role in preventing data leakage. Some DLPSs use existing access lists provided by systems such as Active Directory to gain knowledge of the domain's access control structure (Mogull, 2010). However, obsolete access rights can negatively affect DLPSs. For example, downgraded or dismissed employees can maliciously access data using their old privileges if their access rights are not revoked. Consequently, DLPSs will not be able to detect any violation by that specific user. Moreover, a leak can also be caused by legitimate users, either accidently or intentionally. An efficient DLPS should have the ability to maintain the access rights while protecting data from accidental leaks and otherwise.

3.4. Encryption and steganography

For network-based DLPSs, encryption is considered a major challenge. Network-based DLPSs attempt to identify copies of confidential data using various analysis techniques, and then compare them to the original data. However, use of strong encryption algorithms makes it very difficult or nearly impossible to analyse the data content. For example, a secret document can bypass the detection mechanism if a user encrypts the document and then sends it through an email attachment. In this case, a DLP detection mechanism cannot view the encrypted document as sensitive; hence, the document can be leaked. In addition, many applications work transparently and provide encryption services to users, such as SSL proxies and VPNs (Reed et al., 1998). In this case, the intercepted traffic is anonymous and DLPSs are ineffective unless a proper integration with these services is used. The use of steganography can create another challenge similar to that caused by cryptography. Stenographic tools use common techniques such as least significant bit to hide data within other media such as digital pictures and audio files (Davidson and Paul, 2004). This technique can be used to maliciously leak confidential data since it is highly likely to bypass traffic inspection mechanisms. Therefore, encryption and steganography are considered the ultimate challenge for current DLPSs. Further, sometimes a document needs to be compressed, translated or reformatted to a suitable format. This also can be a challenge for DLPSs that are limited in the format, types and languages they are capable of analysing.

3.5. Data Modification

Some DLPSs use data patterns and signatures to compare between inspected traffic and original confidential data, in order to detect data leakage. The detection takes place when these patterns and signatures are matched or when a high degree of similarity is noticed. However, confidential data is not always sent as is. In fact, confidential data can be exposed to many types of modifications. For example, users can perceptively edit confidential documents by adding, subtracting or substituting lines or even paragraphs before sending it. In addition, the semantics of a document can be rewritten in the form of abstracts or lengthy elaborations. These variations can change the identity of the original document; therefore, data patterns and signatures become ineffective.

Essentially, some DLPSs use data hashing to check outgoing traffic. Hash values—including traditional MD5 and SHA1—of intercepted traffic and existing confidential data are compared for similarity. If the two values are matching then a potential leak is detected. The problem with hashing is that any modification to the original document can lead to a totally different hash value, resulting in a disclosure. A better approach is to divide the confidential document into smaller parts, and then calculate the hash value of each part (Shu and Yao, 2012). In this case, parts of the original document can be detected even after modification. More advanced hashing such as similarity digests, Rabin fingerprinting and using piecewise hashing are designed especially to detect evolved data. However, these techniques are not effective if the data is extensively modified.

3.6. Scalability and integration

Like many other network security mechanisms, DLPSs too can be affected by the amount of data being processed. Whether the DLPS is host based, network based or storage based, DLPSs should be able to process data without delaying the workflow. Factors such as the inspected data size, the computational capabilities and the analysis technique used should be fully investigated to run a scalable DLP system.

DLPSs tend to have a poor association within a network setup. This is because some of their mandatory features already exist in other solutions such as firewalls, IDSs and proxy servers. If a DLPS is to be integrated within a network, a careful scrutiny should be carried out for best overall performance. This should be done to avoid inconsistencies with other security mechanisms and to ease the DLPS's task. For example, if a DLPS is deployed as a proxy gateway, which queues up traffic for inspection, it is recommended to integrate existing proxy services such as Hypertext Transfer Protocol (HTTP) and IM instead (Mogull, 2010). Having two similar services running at the same time can delay or disrupt the workflow. This should also be done between firewalls and DLPSs that have filtering capabilities. To advance the traffic inspection process, services such as SSL proxies and VPNs should be fully integrated with DLPSs. This ensures that DLPSs can analyse traffic in plain text.

3.7. Data classification

DLPSs depend considerably on appropriate data classification. If the data is not classified into different levels, DLPSs will not be able to distinguish between confidential and normal traffic. In military and government sector applications, the use of data classification is common. Military classifications use terms such as ‘restricted’, ‘confidential’, ‘secret’ and ‘top secret’ (Landwehr et al., 1984), which make identifying confidential data easier. This can make DLPSs more oriented towards protecting a specific type of data. Another challenge regarding data classification is assigning responsibility for secrecy levels. As a good practice, the data owners should be responsible for determining how sensitive the data is and whether or not it should be protected (Chen and Liu, 2005). Unfortunately, many data owners ignore this practice and leave data classification tasks for less informed people, such as IT department employees. This can create many uncertainties and cause deterioration to DLP tasks. Hence, without proper data classification, confidential data can easily be revealed even with the presence of DLPSs.

4. Current methods

Since DLPSs are relatively new to both academic and commercial sectors, there is still no agreement on a proper categorisation. As mentioned in Section 2, both sectors have dealt with DLPSs independently. Therefore, each sector is discussed separately in this section. Although data leakage can be prevented through management procedures and security awareness, all of the current methods mentioned in this section are technical solutions. An academic DLPS is normally presented as a full study of a particular concept. Therefore, based on the method used in academic research, we structurally categorised academic DLPs. On the other hand, due to the unavailability of some of the features of industrial DLPs, industrial DLPs are offered based on their special features and common features with other industrial DLPs, as it was reported in the industrial solutions summarised in Table 1. Therefore, we categorised industrial DLPs based on their common and special features.

4.1. Industrial solutions

Data leakage detection and prevention solutions are offered by a wide range of vendors. There are dozens of DLP solutions available as features that can be added to existing security schemes or as standalone systems. Major security vendors are continuously developing new DLPSs with special capabilities to mitigate new data leakage threats. McAfee, Symantec, Trend Micro and Websense are just a few of the long list of vendors that are providing DLP solutions (JafSec, 2015). Most of these solutions are designed to detect and prevent data leakage in different states using content and context analysis. The only clear differences between them are the analysis techniques used, the remedial action taken and the special features offered. Hence, categorising these solutions into groups may not be adequate. To highlight some of the top commercial DLPSs’ features, we selected the top six nominated DLPSs of 2015 according to *SC Magazine* (2015). These DLPSs are listed and compared in Table 1.

As shown in Table 1, all the DLPSs are combined detective and preventive solutions except for the Varonis IDU Classification Framework (detective) and AirWatch (preventive). Moreover, most of the DLPSs were capable of performing content and context analysis. Five of the listed DLPSs were capable of performing at least two of the content analysis techniques: RE (regular expressions), FP (data fingerprinting) and SA (statistical analysis). Depending on the options offered by the vendors, remedial action such as alert, block, encrypt, audit and quarantine can be taken. Triton and McAfee Data Loss Prevention are the only DLPSs that offer all the mentioned remedial actions. Further, the listed DLPSs are all software solutions except for Fidelis XPS and McAfee Data

Table 1
List of Nominated Best Data Leakage Prevention Solutions in 2015.

		Triton (Websense)	Fidelis XPS (General Dynamics Fidelis Cybersecurity Solutions)	McAfee Data Loss Prevention (McAfee)	Check Point DLP (Check Point Software Technologies)	Varonis IDU Classification Framework (Varonis Systems)	AirWatch (VMware)
Type	Detective	✓	✓	✓	✓	✓	
	Preventive	✓	✓	✓	✓		✓
Deployment	In use	✓		✓		✓	✓
	In transit	✓	✓	✓	✓	✓	✓
	At rest	✓		✓		✓	✓
Analysis Type	Content	✓	✓	✓	✓	✓	
	Context	✓	✓	✓	✓	✓	✓
Content Analysis Technique	RE	✓	?	✓	✓	✓	
	FP	✓	?	✓	✓	✓	?
	SA		✓	✓			
Remedial Action	Alert	✓	✓	✓	✓	✓	
	Block	✓	✓	✓	✓		✓
	Encrypt	✓		✓			✓
	Audit	✓	✓	✓	✓	✓	✓
	Quarantine	✓	✓	✓	✓		
Available in:	Software	✓			✓	✓	✓
	Appliance		✓	✓			
Special Features		Detecting data within images and encrypted files+ (DLP drip)	Real time deep session inspection	Forensics analysis prior to the creation of rules	SSL inspection capabilities	Advanced contextual analysis	Very flexible deployment on mobile devices

Note: that all this information was obtained from the vendors' data sheets. Some of the features were not available for product privacy reasons. Therefore, in some cases we used ‘?’ to denote uncertainty. The listed content analysis techniques are RE (regular expressions), FP (data fingerprinting) and SA (statistical analysis).

Loss Prevention since they are available as standalone appliances. As mentioned earlier, one of the distinctive points among these DLPSs is the special features offered. Triton, for example, has the ability to detect sensitive data within images and encrypted files. It also has the ability to detect small data leaks over long periods using a feature called drip DLP (Websence, 2015a and 2015b). Fidelis XPS offers a built-in real time session inspection with low system impact (GeneralDynamics, 2015). McAfee Data Loss Prevention provides the ability to run a forensics analysis prior to the creation of DLP rules (McAfee, 2015). This allows the security administrator to identify existing but not detected data leaks. The remaining listed DLPSs offer special features such as SSL inspection (CheckPoint, 2015), advanced contextual analysis (Varonis, 2015) and flexible deployment for mobile devices (AirWatch, 2015). One of the shared features among all the listed DLPSs is the possibility for these DLPSs to be integrated within existing security systems.

4.2. Academic research

Although the term DLPS is not widely used in academic research, many proposed methods for data leakage prevention can be found in the literature. Whether the proposed methods are used for detection or prevention, an academic DLPS is normally presented as a full study of a particular concept. Academic DLPSs can be categorised according to the application or the method. Categorising DLPSs based on their applications may result in excluding some existing and emerging ones. As novel applications of DLP are emerging, categorizing them based on applications would result in too many categories. On the other hand, categorizing DLPSs based on their methods allow us to include various DLPSs used in different applications within fewer categories. After an extensive literature survey, in Fig. 4 we have categorised DLPSs based on the technique used.

These methods were gathered from a number of academic studies, and listed into seven categories to maintain simplicity. Each method is discussed and the strengths and weaknesses are mentioned under Section 4.3 for evaluation purposes. Note that the category ‘Quantifying and Limiting’ is considered as both preventive and detective method.

4.2.1. Policy and access rights

Preventing data leaks through strict security policies and access rights is widely implemented by many organisations, even before DLPSs appeared as an independent technology. In the literature, there are some DLP management approaches using security policy, data classification and access rights (Sailer et al., 2004). Some host-based DLPSs work by disabling the usage of USB thumb drives and CDs. These DLP systems work according to a security policy, such as preventing a certain department or group of users from using removable media on personal computers (Halpert, 2004). DLPSs that are working on data at rest follow an existing security policy based on who is allowed to access which data object. In both of these examples, access is normally granted to users with credentials that meet the organisation’s policy. Policy and access right DLPSs must have predefined user privileges and data secrecy levels to work properly. These types of DLPDs normally import the organisation’s access control structures from access directories such as Microsoft Active Directory. As mentioned in Section 4, improper data classification or unmaintained access rights may affect DLP performance dramatically.

Further, three main access control policies are found in the literature: discretionary, mandatory and role based (Samarati and de Vimercati, 2001). These access control policies have some limitations in terms of their flexibility and vulnerability, which may have a direct effect on the DLPS. In conclusion, a DLPS based on security policy and access rights is the simplest way to prevent data leakage because it is mature enough and follows well-established foundations.

4.2.2. Virtualisation and isolation

The virtualisation and isolation DLP method uses the advantages of virtualisation to protect sensitive data. The method is based on creating virtual environments when accessing sensitive data, where the user activities are isolated and only trusted processes are allowed. Griffin et al. (2005) introduced a security framework by creating trusted virtual domains that are linked through secure bridges. They envisioned an environment where computing services can be dependably offloaded into trusted execution environments that can maintain security requirements. Another DLP idea was introduced by Burdonov et al. (2009). Their idea was based on using two different virtual machines. One has

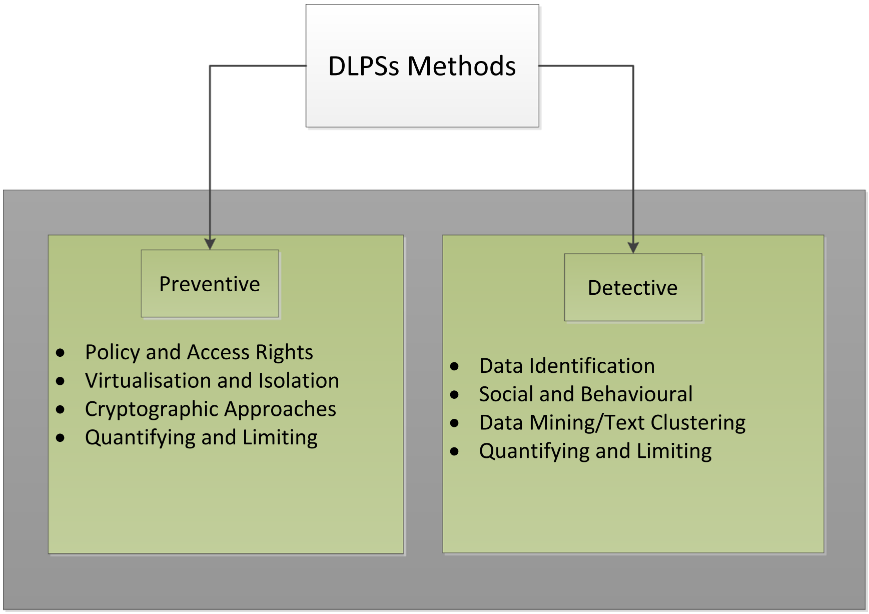

Fig. 4. Data leakage prevention categorisation by method.

unlimited access to the internet and the outer environment—public—and the other is used to process sensitive data—private. A hypervisor separates the two virtual machines, preventing any real interaction between them. Only a trusted application within the private virtual machine is allowed to access the internet using the public virtual machine.

Moreover, Wu et al. (2011) introduced a method to isolate users virtually when they access sensitive data. The authors presented a theoretical active DLP model that combines secured storage with virtual isolation technologies. They suggested that the most severe threats come from inside, that is, from users with privileges; therefore, the main idea of the model is to create a secure data container (SDC) for every user when dealing with sensitive data. The SDC is created dynamically with a corresponding active defence module (ADM) in a data storage layer. The ADM completes the SDC by conducting an information flow analysis for every user. For example, if the user requests access to sensitive data, the ADM will first authenticate the process. If the authentication is passed then an isolated environment will be created with a credible channel between the SDC and the data storage. The process will then be migrated to the corresponding SDC. This is called 'read isolation'. Similar procedures are taken with 'write isolation' and 'communicate isolation' to ensure that all processes are available but use secure channels.

4.2.3. *Cryptographic Approaches*

Cryptography is normally used to protect data from unauthorised disclosure. Cryptographic tools and algorithms have reached a level of maturity where it is extremely difficult or nearly impossible to decrypt data without the right decryption key. The aim of using cryptography is to make it difficult for adversaries to read and understand the secret message (plaintext). However, this cannot prevent them from obtaining the encrypted data (ciphertext). For example, encrypted emails, VPNs and HTTPS secure web are methods used to protect data from being read. This is achieved even when the data is travelling in untrusted environments and where it is vulnerable towards capturing by others. Therefore, encryption might ensure the secrecy of the plaintext but not the ciphertext. This might lead to various types of attacks such as ciphertext, known-plaintext and chosen-plaintext attacks (Schneier and Sutherland, 1995). Although data leakage prevention as a general term means protection of data—which makes all encryption mechanisms eligible to be called DLPSs—encryption methods that deal with data in transit do not lie within this definition. This is because they are involved in releasing data fingerprints, that is, ciphertext. However, some approaches use cryptography to prevent data leakage in data in use and at rest states, and these systems are protecting data within the organisation's confines.

A common practice that uses cryptography to protect data at rest is desk encryption or encrypted file systems (Wright et al., 2003). These systems protect data from adversaries with physical access to computers and storage. They are typically used when a user tries to access an encrypted folder; the user will be asked to supply a key—normally a password—to decrypt the file, and access will be denied if the key is not provided. Another idea was presented by Blanke (2011), based on using a temporary cryptographic key (ephemeral) per user login. This key is used to store sensitive data directly to a remote storage device instead of performing two write operations (first in the local machine storage and second in the remote storage). However, this method requires an existing secure file system, which might not be available in some organisations.

4.2.4. *Quantifying and limiting*

Many activities such as surfing the web, running a process or accessing a specific file can release sensitive data. Even if the released data is not considered sensitive by the data owner, it can help an adversary to gain information about the sensitive data. To overcome such problems, security administrators try to mimic an attacker action using quantifying methods to construct sensitive data and then block the unnecessary leaks. This concept is used in some DLP approaches. For example, Clark et al. (2002) presented a quantifying approach that uses basic information theory to analyse the quantity of sensitive data. This sensitive data may be released by programs written in a very simple imperative language. Moreover, the approach tries to check the absolute leakage rate by bits, using two quantitative models of information leakage.

Borders and Prakash (2009) discussed an approach to measure and constrain the maximum volume of sensitive data in web traffic, instead of trying to detect the presence of sensitive data. The approach uses a measurement algorithm for HTTP, the main protocol for web browsing. These methods can solve problems related to data leakage such as salami attack or covert channel, shown by Aldini and Di Pierro (2008). However, they come with some limitations. For example, they are unable to filter parts of uniform resource locators that contain random numbers to prevent cashing. They can also have problems when quantifying released data that is not written in a simple imperative language.

Quantifying and limiting methods are closely related to *secure publishing*, which is an area that focuses on providing methods and tools for publishing information while preserving privacy (Fung et al., 2010). There have been some attempts to prevent data leakage through document publishing. For example, Yang and Li (2004) proposed a method for secure Extensible Markup Language (XML) publishing in the presence of data inference. They studied the data inference using common knowledge when carelessly publishing a partial document and proposed an algorithm that finds partial documents without causing data leaks, while allowing the maximum amount of publishing.

4.2.5. *Social and behaviour analysis*

Social network analysis focuses on mapping and measuring interactions and relationships between people, groups and organisations, by representing the interactions in terms of nodes and links (Raman et al., 2011). Social interactions include emails, IMs and social networks. By drawing links between nodes and analysing attributes such as the nature, the frequency and the size of transactions, it is possible to visualise a big map of communications between entities. Although human behaviour is unpredictable, it is not always random; therefore, it is useful to keep a history of human reactions for behaviour analysis. A DLPS that uses social and behaviour analysis normally checks the data flow between users and detects any irregularity, and then raises an alarm so a security administrator can react accordingly.

Zilberman et al. (2011) introduced a new approach to prevent data leak through emails based on analysing emails exchanged between members of an organisation by identifying common topics. A relationship line using the term frequency–inverse document frequency (TF–IDF) term weighting function is drawn between members with common topics. A classification model consisting of two phases (training and classification) is then developed. This classification model uses cosine similarity of existing history between users as the detection baseline. If the exchanged topic has little similarity with the existing history, there could be a leak. This concept has limitations, such as false positive leaks because of a short history between a sender and a recipient.

Boehmer (2010) presented a theoretical model for analysing human behaviour. The theory uses case-based reasoning in combination with directed acyclic graphs and a Hamming similarity function. The main idea discussed in this paper is to create a compliance profile for every case or violation and then compare new cases with existing cases profiles. By using Hamming distance with a specific threshold, the model actively reacts to a new violation using existing case profiles. Such a method can predict future human behaviour; however, it needs existing or synthetic compliance profiles for the comparison process. This might introduce some difficulties when identifying detection thresholds.

One approach used in information security to monitor malicious activities is honeypot. A honeypot is a virtual environment created to trick adversaries into falling into a trap. By letting outsiders and malicious insiders access fake data repositories, the system administrator can study their behaviour and prevent data leakage (Mokube and Adams, 2007). Spitzner (2003) introduced novel honeypot applications to detect and identify insider threats. By combining the capabilities of honeytokens and honeynets, the paper proposed a honeypot that can produce early identification and confirmation of an insider threat.

4.2.6. Data identification

Most DLPSs detect sensitive data within legitimate traffic by using deep packet inspection. This method is also used in many applications, such as antiviruses and spam filtering. The method requires previous knowledge of the targeted content, including data fingerprints, regular expressions and exact or partial data match. Data fingerprints are normally created by hashing confidential data using hash functions, whereas regular expression is created from character sequences that form detection patterns. Exact and partial data matching uses various similarity functions to match inspected traffic with existing confidential data. An example of such DLPSs was presented by Shu and Yao (2012), who introduced a network-based data leak detection technique based on message digests or shingles fingerprints, to detect inadvertent data leaks in network traffic. The detection technique uses special digests instead of handling all the sensitive data, which minimises the exposure of sensitive data.

Kantor et al. (2009) presented a patent for document-to-template matching to detect data leak. This theoretical method includes steps to transform a document into a stream of characters, and then split the stream into serialised data lines (TLines). A hash value is then calculated for each line and mapped to a template with a specific score. Finally, a similarity match is measured when a threshold of TLines is found in a document.

4.2.7. Data mining and text clustering

The data mining field has many capabilities for performing sophisticated tasks such as anomaly detection, clustering and classification by extracting data patterns from large datasets. Data mining is strongly related to machine learning, which has a set of algorithms capable of dealing with large data to recognise complex patterns and make an intelligent decision (Witten and Frank, 2005). These relatively new yet powerful techniques are used in some DLPSs such as that presented by Lindell and Pinkas (2000). In their paper, a data-mining algorithm was introduced to preserve privacy between two entities' databases. The algorithm was designed to share confidential data on the union of the entities' databases without releasing unnecessary information. The algorithm was based on decision tree learning for efficient computation. In addition, Marecki et al. (2010) discussed an information flow between the sender and the receivers. The partially observed Markov decision processes (POMDP) method is used over a fixed period, called the decision epoch, and in every epoch, a sender can share only one information object (packet) with a recipient. If a leak is detected, the sender receives a penalty and the recipient receives a positive reward. In this approach, watermarking is used as a monitoring technique to detect leaked data. Although this machine learning method was able to identify data leaks, it was limited to only five recipients. Using POMDP requires a huge amount of calculations; therefore, this method suffers from scalability limitations.

Text clustering and the closely related field 'information retrieval' are also used in DLPSs. Although these methods are well established and utilised in the natural language processing field, they have limited use in DLPSs. The unstructured nature of these methods facilitates the use of artificial intelligence and statistical analysis for detection proposes. For example, Gomez-Hidalgo et al. (2010) used a named entity recognition (NER) approach to identify and extract words from texts. A prototype system was introduced to prevent data leakage into social networks such as Facebook, Twitter, emails and interactive webpages. The NER approach mentioned in this paper uses FreeLing software, which is a language analysis tool. This method achieved a high level of efficiency of more than 90% in classifying both English and Spanish languages. However, there was a limitation in this method when extracting words from documents, as NER could be affected by spelling mistakes and connected words.

4.3. Strengths and weaknesses of current dlp methods

All DLP methods have advantages and disadvantages. From the comprehensive review of the current DLP methods, we summarise strengths and weaknesses in Table 2.

4.4. Summary of most relevant academic methods

Table 3 shows the most relevant studies in relation to academic methods. These studies are considered the state-of-the-art in data leakage detection and prevention. All identified DLPSs are categorised according to the method used, analysis type and the deployment technique. In addition, the contributions and the limitations are also listed.

5. Data leakage prevention systems analysis techniques

Whether DLPSs are used for detection or prevention, usually there is an analysis phase involved in these tasks. Two main analysis techniques are used in DLPSs: context analysis and content analysis. This section explains the differences between the two techniques and discusses some examples. The importance of content analysis compared to context analysis is exemplified. The significance of content statistical analysis as one of the state-of-the-art method in data leakage detection is also apparent.

A third technique called content tagging is used in some DLPSs. This technique is used to tag the file containing confidential data. Even with the most excessive alteration of content, such as changing format, compressing and encrypting, the same file tag can remain intact. Suen et al. (2013) introduced a technique called S2Loggeer to track files while travelling in the cloud. S2Loggeer is able to detect malicious actions, data leakages and data policy violation. Although this technique might seems robust, it can be bypassed if the same confidential data appears in a different unrecognised tag. Hence, content tagging can preserve the identity of the file but not the contained confidential data.

5.1. Context analysis

Context analysis uses metadata associated with actual confidential data. To keep track of confidential data, DLPSs perform contextual analysis of the transaction rather than of the actual data. For example, if a user is sending data to another entity, contextual attributes such as source, destination, timing, size and format will be studied. These attributes can be used to form process or transaction patterns, and based on predefined policies,

Table 2
Strengths and weaknesses of current DLP methods.

Method	Strengths	Weaknesses
Policy and Access Rights	• suitable for any organisation if access rights and data classification are properly established • easy to manage • suitable for data in use and at rest • strong prevention mechanism	• affected by improper data classification • affected by the access control policy in use • not a detective method, hence if a leak is happening the method is ineffective
Virtualisation and isolation	• requires small hardware implementation • dynamic as it does not need regular administrative interference • accessing sensitive data can use existing data classification	• not mature enough • produces considerable amount of overheads • not a detection method
Cryptographic Approaches	• strong cryptography can produce maximum security • cryptographic methods are wieldy to use and have many options	• cryptography can secure sensitive data but may not deny its existence • does not detect data leak • confidential data can be accessed by weak credentials
Quantifying and limiting	• goes beyond studying sensitive data, and focus of the leaking channels • useful against specific types of attacks such as salami attacks • effective for all data states	• does not ensure total blockage of the leaking channel • limited to specific situations or scenarios • can disrupt workflow
Social and behaviour analysis	• proactive data leakage prevention by detecting malicious relations • suitable for all data states	• produces high level of false positive • requires regular administrative interference • requires huge amount of profiling and indexing
Data identification	• very strong in detecting unmodified data • very low false positive level for analysis using fingerprints • some robust hashing can detect modified data	• extremely modified data cannot be detected • lacking semantic understanding
Data Mining and Text Clustering	• can predict future data leaks • powerful in detecting unstructured data • less dependent on administrative help • flexible and adaptable	• requires a great deal of processing • requires learning phase, which means many false positives

outliers can be identified. Various papers (Blanke, 2011; Boehmer, 2010; Burdonov et al., 2009; Clark et al., 2002; Griffin et al., 2005; Wu et al., 2011; Zilberman et al., 2011) have discussed DLP methods using contextual analysis. DLPSs using this type of analysis share many features with anomaly-based IDSs, such as outlier detection. For example, Sithirasenan and Muthukkumarasamy (2008, 2011) introduced an anomaly detection technique in wireless networks using group outlier scores. This method is used to detect rare individual or group-associated events in a wireless environment. Tests show effectiveness of this method to detect various potential attacks such as replay attacks, malicious wireless associations and session hijacks.

Likewise, some DLPSs use context analysis combined with some content features. For example, Carvalho and Cohen (2007) presented a method for preventing information leaks in emails. The authors presented the first attempt to prevent information leaks through emails, in which the goal was to predict unintended message recipients. The problem was addressed as an outlier detection task, where the unintended email addresses were considered outliers. By using combined textual features and social network analysis, the introduced model was able to predict leak recipients in almost 82% of the test emails. The textual analysis calculates cosine similarity scores from new emails intended to be shared and existing ones. Email recipients can then be ranked according to the similarity scores. Recipients with low scores can be considered outliers. In addition, the social network analysis included score features such as normalised sent frequency and normalised received frequency. Although the paper showed encouraging results in detecting outliers, the created scenarios criteria were totally subjective. In addition, since the experiments used simulated email leaks, no real sensitive emails were tested. Moreover, it was noted that the threshold proposed in this study can create false outliers. For example, new emails with no existing interactions will have low scores, which means there is a potential information leak.

In addition, Zilberman et al. (2011) introduced an approach for analysing group communication to prevent data leakage via emails. The approach is based on analysing emails exchanged between members of the organisation and identifying common topics. A relationship line is drawn between members with common topics. The approach consists of two phases (training and classification) to develop a classification model. This classification model is used to set a detection baseline. False positive results were found because of short history between the sender and the recipient. The dataset used in this approach was limited to the Enron dataset, which has problems such as repetition and unknown users.

5.2. Content analysis

Since the main purpose of using DLPSs is the protection of confidential data, it is more important to focus on the content itself than on the context. This is what DLPSs with content analysis capabilities are trying to achieve. Content analysis in DLPSs is done through three main techniques: data fingerprinting (including exact or partial match), regular expression (including dictionary-based match) and statistical analysis.

5.2.1. Data fingerprinting

Data fingerprinting is the most common technique used to detect data leakage. In many DLPSs, a whole file can be hashed using conventional hash functions such as MD5 and SHA1. Such DLPSs can have 100% detection accuracy if the file is not altered by any means. Since confidential documents are subject to change, DLPSs with conventional hashing can be ineffective because of hash value susceptibility to change. There have been many attempts to generate robust fingerprinting techniques that can represent sensitive data even after modification. Kantor et al. (2009) attempted to reduce the effect of document modification by hashing smaller parts within a document. Although this might help in the case of deleting or adding a few sentences, small but scattered changes can make this method ineffective. The state of the art fingerprinting methods are based on more randomised hashing developed by Rabin in 1981 (Broder, 1993). For example, Shu and Yao (2013) used a fuzzy fingerprints algorithm to detect inadvertent data leaks in network traffic. The aim of this approach

Table 3
List of the most relevant work in the literature.

	Paper and category	Method	Analysis	Suitable for:	Contribution	Limitations
1	Wuchner and Pretschner, 2012 (*Policy and Access Rights*)	Detective/ Prevention	Context	In use	UC4Win, a data loss prevention solution for Microsoft Windows operating systems	Requires predefined policy. Cannot identify sensitive data
2	Squicciarini et al., 2010 (*Policy and Access Rights*)	Preventive	Context	In use	Introduces a three layers data protection framework	Requires a pre-defined classification for data. Mis-classified sensitive data can be leaked
3	Agarwal and Tarbotton, 2009 (*Policy and Access Rights/Virtualization and Isolation*)	Preventive	Context	In use/ In transit	Controls data transmission between virtual machines using wrapped applications and a policy module	No experimental results or verification are given
4	Griffin et al., 2005 (*Virtualization and Isolation*)	Preventive	Context	In use	Proposes Virtual Trusted Domains VTD to offload processes to secure environments	Imposes challenge to computational capabilities
5	Burdonov et al., 2009 (*Virtualization and Isolation*)	Preventive	Context	In use	Uses public and private virtual machines to separate secure environment and the internet	Can introduce significant system overheads
6	Wu et al, 2011 (Virtualization and Isolation/ *Cryptographic Approaches*)	Preventive	Context	In use/At rest	Introduces a combination of encrypted storage and virtual environment to prevent data leakage	Suitable for data at rest only. Cannot prevent data leakage caused by privileged used
7	Blanke, 2011 (*Cryptographic Approaches*)	Preventive	Context	At rest	Uses ephemeral encryption to protect data whenever accessed by a user	Requires a pre-implementation of an encrypted file system
8	Clark, 2002 (*Quantifying and Limiting*)	Detective	Content	In transit	Proposes two quantitative models to quantify data leaks	Unable to achieve high detection results with unbounded iteration
9	Yoshihama et al., 2010 (*Quantifying and Limiting*)	Detective	Content/ Context	In transit	Uses an application-level proxy to detect potential data leakage risks	Cannot detect data leaked through covert channels
10	Borders and Prakash 2009 (*Quantifying and Limiting*)	Detective	Content	In transit	Constrains the maximum volume of sensitive data in web traffic	Unable to filter parts of URLs that contains random numbers to prevent cashing
11	Suen et al., 2013 (*Quantifying and limiting/ Social and Behaviour Analysis*)	Detective	Context	In transit	Uses S2Loggeer to track files while travelling in the cloud	Based on content tagging. Cannot track sensitive content
12	Boehmer, 2010 (*Social and Behaviour Analysis*)	Detective/ Preventive	Context	In use	Uses case-based reasoning (CBR) in combination with directed acyclic graph (DAG) and Hamming similarity function	Needs existing or synthetic compliance profiles for comparison process
13	Shapira et al., 2013 (*Data Identification*)	Detective	Content	In use	Robust fingerprinting to overcome shortcoming in ordinary hashing	Requires extensive data indexing for both sensitive and normal data
14	Kantor et al. 2009 (*Data Identification*)	Detective	Content	In use/ In transit	Represents documents as serialized data lines (TLines) and hash values map	Not suitable for large scale data leakage. Documents template creation is time consuming
15	Kale and Kulkarni 2012 (*Data Identification*)	Detective	Context	In use/ In transit	Uses Watermarking to detect the leaker of sensitive information.	Dose not provide complete security since it is only used for tracking leakers
16	Shu and Yao, 2013 (*Data Identification*)	Detective	Content	In transit	Uses message shingles/fuzzy fingerprints to detect inadvertent data leak in network traffic	Modified data can cause false negatives because the shingles fingerprints are different from the original ones
17	Hart et al., 2011 (*Data Mining and Text Clustering*)	Detective	Content	In use	Uses SVM Machine learning to classify documents to private and public.	Inflexible, Limited to two categories
18	Lindell and Pinkas, 2000 (*Data Mining and Text Clustering*)	Preventive	Content	In use/At rest	Sharing confidential data on the union of the entities databases, without releasing unnecessary information	Theoretically proven but lacking practical experiments
19	Marecki et al., 2010 (*Data Mining and Text Clustering*)	Detective	Context	In transit	Uses Partially Observed Markov Decision Processes) over decision epochs	POMDP requires huge amount of calculations
20	Gomez-Hidalgo et al., 2010 (*Data Mining and Text Clustering*)	Detective/ Preventive	Content	In transit	NER (named entity recognition) approach is used to identify and extract words from texts	Named entity recognition could be affected by spelling mistakes and connected words
21	Sokolova et al., 2009 (*Data Mining and Text Clustering*)	Detective	Content	In transit	Uses support vector machine to classify enterprise documents as sensitive non-sensitive	Not fixable because it classifies data to public or private only
22	Parekh et al., 2006 Data (*Mining and Text Clustering*)	Detective	Content	In transit	A new approach to enable the sharing information of suspicious payloads	Polymorphic/obfuscated worms and mimicry attacks may create a big challenge
23	Carvalho et al., 2009 (*Data Mining and Text Clustering*)	Detective	Content/ Context	In transit	Presents an extension –Cut Once- to “Mozilla Thunderbird”	Introduces high level of false positives, since it requires existing messages in the sent folder
24	Zilberman et al., 2010 (*Data Mining and Text Clustering/Social and Behaviour Analysis*)	Detective/ Preventive	Content/ Context	In transit	Uses TF-IDF and cosine similarity to compute existing links between users	High false positive rates because of short history between senders and recipients
25	Carvalho and Cohen 2007 (*Data Mining and Text Clustering/Social and Behaviour Analysis*)	Detective	Content/ Context	In transit	Predicts unintended message recipients	Can create false outliers because of limited interaction history

is to quantify and restrict confidential data exposure when dealing with a cloud computing provider. Methods based on Rabin's randomised fingerprinting show some advantages over conventional fingerprinting. However, they suffer from some limitations such as coverage and inescapable false positive rates since Bloom filters are used (Broder and Mitzenmacher, 2004).

Another example of advanced fingerprinting was presented by Shapira et al. (2013). They proposed an extended fingerprinting approach. They presented modified full data fingerprinting to overcome the shortcomings in fingerprints produced by ordinary data hashing. Ordinary fingerprints are vulnerable and can be bypassed even with minor modification to the original data; therefore, *k*-skip-*n*-grams are used to produce modified fingerprints. The *k*-skip-*n*-grams introduce a robust method to identify the original data even after data modification (addition, subtraction, word synonyms). Both confidential and non-confidential documents are processed to produce fingerprints in this method, in which non-confidential documents produce non-confidential *k*-skip-*n*-grams. The non-confidential *k*-skip-*n*-grams help to eliminate unnecessary n-grams in the confidential documents. The proposed method outperformed ordinary full fingerprinting in almost all the experiment scenarios. However, it requires intensive indexing for all confidential and non-confidential documents. This requires extra storage and processing capabilities, which can be a major drawback of this method.

5.2.2. *Regular expression*

Regular expression, introduced by Kleene (1951), is another popular method used in DLPSs. It consists of sets of terms or characters that are used to form detection patterns. These patterns are used to compare and match sets of data strings mathematically. Regular expression is typically used in search engines and text processing to validate, extract and replace data. For example, through a set of patterns, similar words such as *centre* and *center* or *colour* and *colour* can be identified. Patterns normally include two types of characters, normal characters (literal meaning) and metacharacters (special meaning). Metacharacters are a set of symbols such as (. | * $? +) that are used along with normal characters to form a detection pattern. In information security, regular expression is used mostly in data inspection for malicious codes or confidential data. Becchi and Crowley (2007) used regular expression along with a compression algorithm to accelerate the detection of critical patterns in packet payload. This method was designed for intrusion detection and showed considerable benefits over related work in terms of speed and simplicity. Yu et al. (2006) proposed a regular expression rewrite technique to reduce the memory use for packet inspection. The proposed technique was designed as a deterministic finite automaton formulation of regular expression. Although the method showed notable improvement in inspection speed, new attacks may not be detected without rewriting them. Typically, regular expressions are used in DLPSs for exact and partial detection of social security numbers, credit card numbers, and confidential corporate and personal records (Mogull, 2010.). Moreover, special dictionaries can be used to identify specific data such as medical terms and geographical information (Sokolova et al., 2009). These dictionary-based techniques can help by accelerating and improving the detection significantly.

5.2.3. *Statistical analysis*

Statistical content analysis can facilitate powerful tools such as machine learning classification and information retrieval term weighting. Moreover, by using different techniques and classification algorithms, statistical analysis can deal with nebulous types of data rather than a limited scope. It mainly depends on analysing the frequency of terms and n-grams within a document. A term simply means a word, while an n-gram can be a word or pieces of a word such as unigram (one character), bigram (two characters) and trigram (three characters). N-gram analysis and term weighting analysis are the main statistical analysis techniques.

5.2.3.1. N-gram analysis. N-gram analysis is a widely used method in many applications, especially in linguistics. It is one of the simplest ways to analyse data based on the frequency of data of interest. In the context of data leakage detection and prevention, n-gram statistical analysis was used by Hart et al. (2011), who introduced a method based on machine learning to classify enterprise documents as sensitive or not. This approach used support vector machine algorithms to classify three types of data: enterprise private, enterprise public and non-enterprise. The data was represented by the most frequent binary weighted unigrams found across all corpora. The method was able to identify 97% of data leaks with a false negative rate of 3.0%. Unfortunately, this method classifies data as public or private only, ignoring more flexible classification levels such as top secret, secret and confidential. In reality, this method can obstruct the work process, making the enforcement of security policies a difficult task.

In addition, Sokolova et al. (2009) presented a system to detect sensitive data in heterogeneous texts. The system works by analysing peer-to-peer file exchanges within a networked environment. The authors indicated that the system uses an untraditional way to detect sensitive information, by processing data of unknown content, context and structure. That is because the proposed system can process unstructured data. Traditionally, most systems used in personal health information (PHI) leak prevention use specific tools to detect 'personally identifiable' data. This identifiable data may include names, ages and phone numbers without any focus on PHI. Moreover, traditional PHI systems work in closed hospital environments of medical records, where the processed information is guaranteed to have PHI. The system proposed in this paper is based on the separation of possible and impossible containers of PHI. Eliminating impossible containers and focusing on the possible ones is the key factor of the proposed system. The process of eliminating impossible containers of PHI consists of using string matching and character n-gram modelling methods. This was conducted to check and eliminate published titles, non-text and non-English files. This method was limited to identifying PHI files only and might introduce some shortcomings if applied to regular files.

Parekh et al. (2006) presented a new approach to enable the sharing of information of suspicious payloads. Two anomaly detectors (PAYL and Anagram) were used to support generalised payload correlation and signature generation between sites and domains. The approach uses three types of alert correlation (raw packet, frequency-based and n-gram alert correlations) to classify payloads. Polymorphic/obfuscated worms and mimicry attacks may create a major challenge for this approach. In addition, the Anagram anomaly detector uses Bloom filters bit representation, which suffers from false positives.

5.2.3.2. Term weighting. Term weighting is a statistical method that indicates the importance of a word within a document. It is normally used in text classification using vector space models, where documents are treated as vectors. This method was introduced by Salton et al. (1975) in a novel approach based on space density computations. It was shown that separation between document spaces resulted in better retrieval. Therefore, a clustered document space was considered best, where related documents were grouped into classes. These classes are formed around cluster centroids, which are also formed around a main centroid.

The term weighting technique is mostly based on the classical TF–IDF. This is the best-known term weighting function that and it shows efficiency and accuracy when used for classification purposes. The TF–IDF function is illustrated below:

$$\text{Term frequency }(tf)\text{:}W_{t,d} = \begin{cases} 1+\log_{10}tf_{t,d}, \text{if } tf_{t,d} > 0 \\ 0, \text{otherwise} \end{cases} \quad (1)$$

where *tf* simply means the frequency of a specific term within one document. The use of logarithmic representation is important to avoid dealing with huge figures, and (1) is added to avoid an undefined result, in case a term is not present.

$$\text{Inverse document frequency }(idf)\text{: } idf_t = \log_{10}\left(\frac{N}{df_t}\right) \quad (2)$$

This is an inverse measure of the informativeness of *t*, where *N* is the total number of documents in a corpus, and *dft* means the number of documents within a corpus that contain a specific term. For example, if we have a corpus of 1000 documents, and if the term'the' appears in all the documents, $\log_{10}(1000/1000)$ will result in (0). This means that the term ‘the’ has no real value. In contrast, if a term is rarely sighted in the corpus then the *idf* is bigger. The total weight for a term is given by the product value of *tf* and *idf* (Manning et al., 2008).

This term weighting approach has been limitedly used in data leakage prevention. For example, Carvalho et al. (2009) presented an extension—Cut Once—to the publicly available email client Mozilla Thunderbird. This extension was built with capabilities to recommend trustworthy recipients and predict potential leaks through wrongly addressed emails. It works by treating each email address available in the user address book as a TF–IDF centroid vector. This vector is calculated using term frequencies from messages available in the sent folder. When a new message is composed, a TF–IDF vector is calculated and compared with the already calculated TF–IDF centroid vectors for each email address. The Cut Once extension ranks email addresses intended to receive the new message according to the calculated TF–IDF score. Email addresses—that is, contacts—with high scores indicate existing exchanged messages with similar topics. Lower scores indicate wrong recipients or an unrecognised new topic. The experiment was limited to 26 real users of Mozilla Thunderbird. A total of 2315 emails were exchanged between the subjects over a course of four weeks. Unfortunately, the detection of the email leaks was totally subjective, since it was up to the users to select appropriate recipients. Only five email leaks were reported by users. Additionally, this approach may introduce a high level of false positives, since it requires existing messages in the sent folder.

6. Future trends in data leakage prevention systems

Ongoing developments for DLPSs involve two main areas: the analysis techniques and the applications. It is noticed that attention is directed towards developing content analysis techniques that can preserve data semantics, and applications developers are focusing on mitigating insider threats and data leakage through mobile devices.

6.1. Analysis techniques

Many DLPSs vendors and researchers have focused on developing analysis techniques to produce robust DLPSs. Varonis (2015), for example, claimed that context analysis is the future for efficient DLPs. This is because content analysis techniques are sophisticated and cumbersome. Therefore, having a robust contextual analysis with an auditing system on data usage, permissions and ownership can provide the protection needed for data. However, content analysis techniques are being developed to overcome various shortcomings. As mentioned previously, current data fingerprinting techniques are susceptible to change. Therefore, advanced types of hashing such as similarity digests and locality-sensitive hashing (LSH) can enhance the semantic content analysis. Unlike conventional hashing, where comparison means yes or no answers, similarity digests provide more fixable measures. A degree of similarity is approximated when using similarity digests (typically 0% to 100%). Roussev (2010) presented data fingerprinting based on similarity digests. The method generates statistically improbable fingerprints to overcome uneven coverage and high false positives found in randomised fingerprinting such as Rabin’s fingerprint. This method proved its robustness in correctly classifying small segments of data from six common file types, including doc, pdf and html. Although the method introduced better coverage for feature selection, it was not clear whether it could identify whole altered documents and to what degree. Along the same lines, a comprehensive comparison of similarity hashes was given by Roussev (2011), who evaluated the capabilities of fuzzy hashing (ssdeep)—pioneered by Rabin 1981—and similarity digest (sdhash). The study showed that in all test cases, similarity digest outperformed fuzzy hashing in terms of accuracy and scalability.

LSH is another form of advanced fingerprinting. This method was developed by Indyk and Motwani (1998) and Gionis et al. (1999). The method is based on having multiple hashing for neighbouring elements to increase the chance of value collisions. Higher collision probability for close-by elements can help in determining neighbouring points, by performing the same hash procedure for queries. Unlike conventional hashing, where collision is not desired, LSH collided values mean a higher degree of similarity. This method is widely used in many similarity-measuring applications such as data clustering, data mining and digital forensics (Cohen et al., 2001; Haveliwala et al., 2000; Ke et al., 2004). In addition, many forms of enhanced LSH are used for specific cases. Datar et al. (2004) introduced an LSH algorithm to utilise Euclidean space to overcome the drawback of Hamming space being limitedly used. This method was used in a fast colour-based image similarity search. Another form of LSH, called TLSH, was introduced by Oliver et al. (2013). TLSH is the TrendMicro LSH, and it uses a sliding window of five bytes. At a certain window place, the five bytes form a trigram that is finally mapped into a counting bucket using Pearson’s (1990) hash. To improve the accuracy, this method uses quartiles to track counting bucket heights.

Statistical content analysis is also expected to leverage the capabilities of future DLPSs. Advanced types of n-gram and term weighting techniques have shown encouraging results in classifying data semantics. For example, n-gram statistical analysis is used in intrinsic detection such as the new plagiarism detection method presented by Stamatatos (2009), which uses n-gram profiles to detect style variation within a document. It is an intrinsic plagiarism detection method, in which there is no need for a reference corpus to detect plagiarised passages. In other fields, n-gram statistical analysis has been used to classify data of special interest. For example, Reddy and Pujari (2006) presented a new approach that uses relevant n-grams to detect computer viruses. Their technique, called class-wise document frequency, uses relevant n-grams to classify executable files. The classes were V (viruses) and B (benign). The classification theory used in this paper is the Dempster–Shafer theory of evidence, which combines support vector machines and decision trees. In addition, Shabtai et al. (2012) used OpCode n-gram patterns to detect unknown malicious code. The idea is based on extracting n-gram patterns from files after disassembly. The disassembly process is different from extracting normal n-gram patterns from files, as it translates machine code to more human readable language (*assembly language*). The result is a sequence of OpCode n-grams, and depending on the n-gram size required, multiple OpCode patterns may be created. The evaluation results showed a high level of accuracy of more than 96% with a true positive rate above 0.95 and false positive rate around 0.1.

New term weighting techniques have also been developed to define accurate data semantics. For example, Nunes et al. (2011) presented four term weighting functions to estimate the current (most accurate) term score. The idea is based on checking the revision history for terms in documents. In theory, a document may be edited or revised many times during its existence. Therefore, terms might have different frequencies in a course of time or might be eliminated. This method can help in identifying accurate term scores after a series of document modifications, but it is limited to sporadic changes. A combination of TF–IDF term weight and n-gram statistics was presented by Moskovitch et al. (2009) as a new methodology for the representation of malicious and benign. In their paper, the terms are represented by different size n-grams extracted from malicious and benign files. A comprehensive study on feature selection (ranking) was performed using document frequency, gain ratio and Fisher score. This method was able to achieve a high level of accuracy of 95% when the malicious file percentage was less than 33%, which simulates a real life scenario. Further, a semantic hashing approach was presented by Salakhutdinov and Hinton (2009) to achieve higher precision and recall than TF–IDF or LSH. The method is based on using binary codes as memory addresses to find semantically similar documents. Such a method can leverage content analysis DLPSs' capabilities significantly.

6.2. Applications

The second main development area for DLPSs is the applications. This is mainly happening in commercial DLPSs, as vendors are trying to satisfy consumer requirements. One of the main concerns attracting DLPS developers' attention is internal misuse. Encouraged by incidents in which insiders are the cause of the data leak—for example, WikiLeaks (Karhula, 2011)—new DLPSs are focusing on internal misuse detection. Internal misuse happens when authorised users use queries to retrieve small pieces of data. These small pieces of data can be accumulated to construct meaningful restricted information. Users can also use their access rights to access data maliciously. Methods such as that used by Yaseen and Panda (2009) are aimed at studying knowledge acquisition by insiders and the types of dependencies between data objects. Yaseen and Panda (2009) proposed a dependency graph called the Neural Dependency and Inference Graph. It can measure the amount of information that can be acquired about some data items using their dependency relationships. In addition, Stolfo, Salem and Keromytis (2012) presented a data access monitoring approach based on profiling a user's behaviour when accessing data in the cloud. Patterns are then checked to determine if and when a malicious insider illegitimately accesses data. When unauthorised access is detected, the malicious insider is flooded with bogus information to dilute the real sensitive data.

Another area where DLPS applications are being developed is smart phone security. More and more capabilities are put into smart phones, which enables them to handle a large amount of data. This can jeopardise the security of personal and corporate sensitive data, since smart phones are susceptible to theft and loss. AirWatch (2015) by VMware is an example of the new DLP solutions made for mobile devices. AirWatch mobile content management is designed to protect data accessed by smart phones and tablets anytime, anywhere. This is done by creating secure containers enabling users to access, store and update data securely from mobile devices. TRITON AP-Mobile by Websense (2015a, 2015b) uses a special agent and DLP policies in mobile devices to restrict unauthorised data sharing. This includes preventing data from being leaked through images and encrypted data.

7. Limitations

In this paper we considered a wide range of aspects in the data leakage prevention field. However, there are some aspects that are not fully reflected here. This is due to the fact that this paper is structured on categorising DLPSs based on the technique rather than the application. For instance, there is a huge body of literature on secure publishing and privacy preserving. An example was given under 'Quantifying and Limiting', however; privacy-preserving data publishing uses many other methods such as differential privacy, *k*-anonymity and *i*-diversity (Fung et al., 2010). Similarly, cryptographic methods are widely used in securing and auditing data access (Wang et al., 2010) and not limited to what is mentioned under 'Cryptographic Approaches'. Such methods are used in DLP applications in specific domains such as in healthcare and cloud computing (Narayan et al., 2010; Li et al., 2013; Cong et al., 2010). Covering all possible DLP techniques and applications in such domains and possible sub-domains is challenging.

8. Conclusion and future work

Data leakage is an ongoing problem in the field of information security. Both academics and practitioners are continuously working towards developing data leakage prevention and detection methods to mitigate this problem. DLPSs are increasingly recognised as preferred solutions for identifying, monitoring and protecting confidential data. In this paper, we conducted a survey on both commercial and academic DLPSs. The aim of the survey was to draw attention towards this inadequately researched area and to provide academics and practitioners with a comprehensive reference to DLPSs. We have explained the DLP paradigm and identified the current challenges. Further, we have discussed a number of commercial DLPSs and highlighted their capabilities. In addition, we have listed all relevant academic research studies and suitably categorised the DLPS methods. All of the methods were systematically analysed and the contributions and limitations were acknowledged.

From industry references and the literature, it is evident that data leakage detection and prevention methods are inadequately studied. Moreover, most of the current methods suffer from serious limitations, especially when the confidential data is evolving. This is because they mainly depend on inflexible techniques. Even with some robust fuzzy fingerprinting and statistical analysis, the confidential data semantics can be leaked using various obfuscations. Therefore, a potential research question in this area is "how to detect semantically the content of confidential data in order to prevent data leakage". An effective future DLPS should have the ability to classify confidential data semantically even if it is evolving. Although some researchers insist on relying on contextual analysis, it is hard to protect the semantics without knowing the content. Moreover, current DLPSs maintain copies or references of confidential data. This allows successful identification of leaks when they are occurring. Unfortunately, this is not sufficient, since confidential data can be created without going through classification procedures. Therefore, DLPSs should have the ability to heuristically detect such data without the need for managing exact copies of existing and new data. In addition, detection techniques require extensive analysis that includes deep content inspection and comprehensive indexing. This can impose a serious challenge to DLPSs by exhausting their computational and storage capabilities. Hence, an efficient DLPS should be able to perform the detection tasks with minimal computational and storage requirements.

Privileged malicious insiders are considered the most detrimental threat to confidential data. Not only do they know what data should be targeted and where it is kept, but they also know its value. Such a challenge can be addressed by combining content and context analysis capabilities along with procedural security. Hence, DLPSs targeting privileged malicious insiders should be able to integrate technical capabilities with information security audit and assurance. This can result in a DLP framework that allows confidential data protection beyond normal data states, that is, 'under authority'.

References

AirWatch. AirWatch Mobile Content Management. Retrieved from ⟨http://www.air-watch.com⟩; 2015.

Agarwal S, & Tarbotton LCL. System and method for preventing data loss using virtual machine wrapped applications: USA Patent No. US20110113467 A1;2009.

Aldini A, Di Pierro A. Estimating the maximum information leakage. Int J Inf Secur 2008;7(3):219–42. http://dx.doi.org/10.1007/s10207-007-0050-.

Arthur C & Stuart K. PlayStation Network users fear identity theft after major data leak. Retrieved from ⟨http://www.theguardian.com/technology/2011/apr/27/playstation-users-identity-theft-data-leak⟩; 2011.

Becchi M & Crowley P. An improved algorithm to accelerate regular expression evaluation. In Proceedings of the 3rd ACM/IEEE symposium on architecture for networking and communications systems. Orlando, Florida, USA; 2007. p. 145–54.

Blanke WJ. Data loss prevention using an ephemeral key. Paper presented at the 2011 international conference on high performance computing and simulation (HPCS); 2011, 4–8 July 2011. p. 412–18.

Boehmer W. Analyzing human behavior using case-based reasoning with the help of forensic questions. Paper presented at the 2010 24th IEEE international conference on advanced information networking and applications (AINA); 2010, 20–23 April 2010. p. 1189–94.

Borders K, & Prakash A. Quantifying information leaks in outbound web traffic. Paper presented at the 2009 30th IEEE symposium on security and privacy. Berkeley, CA; 2009, 17–20 May 2009. p. 129–40.

Broder A, Mitzenmacher M. Network applications of bloom filters: a survey. Internet Math. 2004;1(4):485–509. http://dx.doi.org/10.1080/15427951.2004.10129096.

Broder AZ. Some applications of Rabin's fingerprinting method. In Sequences II. Springer; 1993. p. 143–52.

Burdonov I, Kosachev A., & Iakovenko P. Virtualization-based separation of privilege: working with sensitive data in untrusted environment. Paper presented at the Proceedings of the 1st EuroSys workshop on virtualization technology for dependable systems. Nuremberg, Germany; 2009. p. 1–6.

Carvalho VR, & Cohen WW. Preventing Information Leaks in Email. Paper presented at the SDM; 2007. p. 68–77.

Carvalho VR, Balasubramanyan R, & Cohen WW. Information leaks and suggestions: A case study using mozilla thunderbird. Paper presented at the CEAS 2009-sixth conference on email and anti-spam; 2009. Mountain View, California USA.

CheckPoint. Check Point Software Technologies Ltd. Retrieved from ⟨http://www.checkpoint.com/products/dlp-software-blade/⟩; 2015.

Chen K & Liu L. Privacy preserving data classification with rotation perturbation. In Fifth IEEE international conference on data mining; 2005. p. 4. 10.1109/ICDM.2005.121.

Clark D, Hunt S, Malacaria P. Quantitative analysis of the leakage of confidential data. Electron Notes Theor Comput Sci 2002;59(3):238–51. http://dx.doi.org/10.1016/S1571-0661(04)00290-7.

Cohen E, Datar M, Fujiwara S, Gionis A, Indyk P, Motwani R, Yang C. Finding interesting associations without support pruning. IEEE Trans Knowl Data Eng 2001;13(1):64–78. http://dx.doi.org/10.1109/69.908981.

Cong W, Qian W, Kui R, & Wenjing L. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. Paper presented at the 2010 Proceedings IEEE INFOCOM; 2010, 14–19 March 2010.

datalossdb. Data loss ltatistics. Retrieved from ⟨http://datalossdb.org/⟩; 2015.

Datar M, Immorlica N, Indyk P & Mirrokni, VS. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the twentieth annual symposium on computational geometry. Brooklyn, New York, USA; 2004. p. 253–62.

Davidson I, & Paul G. Locating secret messages in images. Paper presented at the Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. Seattle, WA, USA; 2004. p. 545–50.

Fabian M. Endpoint security: managing USB-based removable devices with the advent of portable applications. Paper presented at the Proceedings of the 4th annual conference on Information security curriculum development. Kennesaw, Georgia; 2007.

Fung BCM, Wang K, Chen R, Yu PS. Privacy-preserving data publishing: a survey of recent developments. ACM Comput Surv 2010;42(4):1–53. http://dx.doi.org/10.1145/1749603.1749605.

GeneralDynamics. Fidelis XPS. Retrieved from ⟨http://www.fidelissecurity.com/network-security-appliance/Fidelis-XPS⟩; 2015.

Gionis A, Indyk P & Motwani R. Similarity search in high dimensions via hashing. In Very Large Data Bases; 1999. p. 518–29.

Gomez-Hidalgo JM, Martin-Abreu JM, Nieves J, Santos I, Brezo F & Bringas PG. Data leak prevention through named entity recognition. In Proceedings of the 1st international workshop on privacy aspects of social web and cloud computing. Minneapolis, USA. p. 1129–34.

Griffin JL, Jaeger T, Perez R, Sailer R, Van Doorn L, & Cáceres R. Trusted virtual domains: toward secure distributed services. Paper presented at the Proceedings of the 1st IEEE workshop on hot topics in system dependability (HotDep'05); 2005.

Hackl A & Hauer B. State of the art in network-related extrusion prevention systems. In Proceedings, 7th international symposuim on database engineering and applications; 2009. p. 329–35.

Halpert B. Mobile device security. In Proceedings of the 1st annual conference on information security curriculum development. Kennesaw, Georgiapp; 2004. p. 99–101.

Hart M, Manadhata P & Johnson R. Text classification for data loss prevention. In Privacy Enhancing Technologies. Springer; 2011. p. 18–37.

Haveliwala T, Gionis A & Indyk P. Scalable techniques for clustering the web. In third international workshop on the web and databases (WebDB 2000). Dallas, Texas, USA; 2000.

Indyk P & Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the thirtieth annual ACM symposium on theory of computing. Dallas, Texas, USA; 1998. p. 604–13.

ISACA. Data leak prevention. Retrieved from ⟨http://www.isaca.org/Knowledge-Center/Research/Documents/DLP-WP-14Sept2010-Research.pdf?id=f07c59b7-1bec-4381-9a26-16cb109f5606⟩; 2010.

JafSec. DLP solutions vendor list. Retrieved from ⟨http://jafsec.com/DLP/DLP-A-B.html⟩; 2015.

Kanagasingham P. Data loss prevention. Retrieved from ⟨http://www.sans.org/reading_room/whitepapers/dlp/data-loss-prevention_32883⟩; 2008.

Kale SA, Kulkarni S. Data Leakage Detection. Int J Adv Res Comput Commun Eng 2012;1(9):32–5.

Kantor A, Antebi L, Kirsch Y & Bialik U. Methods for document-to-template matching for data-leak prevention. USA Patent No. US20100254615 A1; 2009.

Karhula P. What is the effect of WikiLeaks for freedom of information? FAIFE Spotlight, 19. Retrieved from ⟨http://www.ifla.org/files/assets/faife/publications/spotlights/wikileaks-karhula.pdf⟩; 2011.

Ke Y, Sukthankar R & Huston L. Efficient near-duplicate detection and sub-image retrieval. In ACM Multimedia, 5; 2004. p. 869–76.

Kleene SC. Representation of events in nerve nets and finite automata. DTIC Document. Retrieved from ⟨http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA596138⟩; 1951.

Kornblum J. Identifying almost identical files using context triggered piecewise hashing. Digit Investig 2006;3:91–7.

Landwehr CE, Heitmeyer CL, McLean J. A security model for military message systems. ACM Trans Comput Syst (TOCS) 1984;2:198–222.

Li M, Yu S, Zheng Y, Ren K, Lou W. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans Parallel Distrib Syst 2013;24(1):131–43.

Lindell Y & Pinkas B. Privacy preserving data mining. In Advances in cryptology—CRYPTO; 2000. p. 36–54.

Manning CD, Raghavan P, Schütze H. Introduction to information retrieval, Vol. 1. Cambridge, England: Cambridge University Press; 2008.

Marecki J, Srivatsa M, & Varakantham P. A decision theoretic approach to data leakage prevention. Paper presented at the 2010 IEEE second international conference on social computing (SocialCom); 2010. p. 776–84.

McAfee. McAfee Total Protection for Data Loss Prevention (DLP). Retrieved from ⟨http://www.mcafee.com/au/products/total-protection-for-data-loss-prevention.aspx⟩; 2015.

Mogull R. Understanding and selecting a data loss prevention solution. Retrieved from ⟨https://securosis.com/assets/library/reports/DLP-Whitepaper.pdf⟩; 2010.

Mokube I & Adams M. Honeypots: Concepts, approaches, and challenges. In Proceedings of the 45th annual Southeast regional conference. Winston-Salem, North Carolina, USA; 2007. p. 321–26.

Moskovitch R, Stopel D, Feher C, Nissim N, Japkowicz N, Elovici Y. Unknown malcode detection and the imbalance problem. J Comput Virol 2009;5(4):295–308. http://dx.doi.org/10.1007/s11416-009-0122-8.

Narayan S, Gagné M, & Safavi-Naini, R. Privacy preserving EHR system using attribute-based infrastructure. Paper presented at the Proceedings of the 2010 ACM workshop on cloud computing security workshop; 2010.

Nunes S, Ribeiro C, David G. Term weighting based on document revision history. J Am Soc Inf Sci Technol 2011;62(12):2471–8. http://dx.doi.org/10.1002/asi.21597.

Oliver J, Cheng C & Chen Y. The trend locality sensitive hash: TLSH. Retrieved from ⟨https://github.com/trendmicro/tlsh/blob/master/TLSH_Introduction.pdf⟩; 2013.

Olzak T. Data leakage: Catching water in a sieve [Blog post]. Retrieved from ⟨http://blogs.csoonline.com/1187/DataLeakage⟩; 2010.

Orgill GL, Romney GW, Bailey MG, & Orgill PM. The urgency for effective user privacy-education to counter social engineering attacks on secure computer systems. Paper presented at the Proceedings of the 5th conference on Information technology education. Salt Lake City, UT, USA; 2004. p. 177–81.

Parekh JJ, Wang K & Stolfo SJ. Privacy-preserving payload-based correlation for accurate malicious traffic detection. In: Proceedings of the 2006 SIGCOMM workshop on large-scale attack defense; 2006. p. 99–106.

Pearson PK. Fast hashing of variable-length text strings. Commun ACM 1990;33 (6):677–80. http://dx.doi.org/10.1145/78973.78978.

Rabin MO. Fingerprinting by random polynomials. Cambridge, MA: Center for Research in Computing Technology, Harvard University; 1981.

Raman P, Kayacık HG, & Somayaji A. Understanding data leak prevention. Paper presented at the 6th annual symposium on information assurance (ASIA'11). Albany, New York, USA; 2011. p. 27–31.

Reddy DKS, Pujari AK. N-gram analysis for computer virus detection. J Comput Virol 2006;*2*(3):231–9. http://dx.doi.org/10.1007/s11416-006-0027-8.

Reed MG, Syverson PF, Goldschlag DM. Anonymous connections and onion routing. IEEE J Sel Areas in Commun 1998;16(4):482–94. http://dx.doi.org/10.1109/49.668972.

Roussev V. Data fingerprinting with similarity digests. In: Advances in digital forensics VI. Springer; 2010. p. 207–26.

Roussev V. An evaluation of forensic similarity hashes. Digit Investig 2011;8:S34–41. http://dx.doi.org/10.1016/j.diin.2011.05.005.

Sailer R, Jaeger T, Zhang X, & Doorn Lv. Attestation-based policy enforcement for remote access. Paper presented at the Proceedings of the 11th ACM conference on computer and communications security; 2004. Washington DC, USA.

Salakhutdinov R, Hinton G. Semantic hashing. Int J Approx Reason 2009;50(7):969–78. http://dx.doi.org/10.1016/j.ijar.2008.11.006.

Salton G, Wong A, Yang C-S. A vector space model for automatic indexing. Commun ACM 1975;18(11):613–20. http://dx.doi.org/10.1145/361219.361220.

Samarati P & de Vimercati S. Access control: policies, models, and mechanisms. In: Foundations of security analysis and design*;* 2001. p. 137–96.

Schneier B & Sutherland P. Applied cryptography: protocols, algorithms, and source code in C. Wiley; 1995.

SC Magazine. 2015 SC Awards U.S. finalists: Round Four. Retrieved from ⟨http://www.scmagazine.com/2015-sc-awards-us-finalists-round-four/article/392362/⟩; 2015.

Shabtai A, Elovici Y, Rokach L. A survey of data leakage detection and prevention solutions. Springer; 2012.

Shabtai A, Moskovitch R, Feher C, Dolev S, Elovici Y. Detecting unknown malicious code by applying classification techniques on OpCode patterns. Secur Inf 2012;1:1–22. http://dx.doi.org/10.1186/2190-8532-1-1.

Shapira Y, Shapira B & Shabtai A. Content-based data leakage detection using extended fingerprinting. arXiv preprint arXiv:1302.2028; 2013.

Shu X, & Yao D. Data Leak Detection as a Service. In A Keromytis & R Di Pietro, editors. Security and privacy in communication networks. Springer Berlin Heidelberg, Vol. 106; 2013. pp. 222–40.

Sithirasenan E & Muthukkumarasamy V. Substantiating security threats using group outlier detection techniques. In: IEEE GLOBECOM 2008. IEEE Global telecommunications conference; 2008. p. 1–6.

Sithirasenan E, Muthukkumarasamy V. Substantiating anomalies un wireless networks using group outlier scores. J Softw 2011;6:678–89.

Sokolova M, El Emam, K, Rose S, Chowdhury S, Neri E, Jonker E & Peyton L. Personal health information leak prevention in heterogeneous texts. In: Proceedings of the workshop on adaptation of language resources and technology to new domains; 2009. p. 58–69.

Spitzner L. Honeypots: catching the insider threat. In: Proceedings of the 19th annual computer security applications conference; 2003. p. 170–9.

Squicciarini A, Sundareswaran S, & Lin D. Preventing information leakage from indexing in the cloud. Paper presented at the 2010 IEEE 3rd international conference on cloud computing (CLOUD); 2010, 5–10 July 2010.

Stamatatos E. Intrinsic plagiarism detection using character n-gram profiles. Threshold 2009;2(500):1.

Stolfo SJ, Salem MB & Keromytis AD. Fog computing: Mitigating insider data theft attacks in the cloud. In IEEE symposium on security and privacy workshops (SPW); 2012. p. 125–8.

Suen CH, Ko RK, Tan YS, Jagadpramana, P & Lee BS. S2logger: end-to-end data tracking mechanism for cloud data provenance. In 12th IEEE international conference on trust, security and privacy in computing and communications (TrustCom); 2013. p. 594–602.

Varonis. IDU classification framework. Retrieved from ⟨http://www.varonis.com/products/data-classification-framework.html⟩; 2015.

Wakefield J. eBay faces investigations over massive data breach. Retrieved from ⟨http://www.bbc.com/news/technology-27539799⟩; 2014.

Wang C, Ren K, Lou W, Li J. Toward publicly auditable secure cloud data storage services. Network, IEEE 2010;24(4):19–24.

Websence. TRITON AP-Mobile. Retrieved from ⟨http://www.websense.com⟩; 2015a.

Websense. TRITON AP-WEB. Retrieved from ⟨http://www.websense.com/content/triton-ap-web.aspx⟩; 2015b.

Witten IH, Frank E. Data mining: practical machine learning tools and techniques. Morgan Kaufmann, Elsevier; 2005.

Wright CP, Dave J, & Zadok E. Cryptographic file systems performance: what you don't know can hurt you. Paper presented at the Proceedings of the second IEEE international security in storage workshop, 2003. SISW '03, 31–31 October; 2003.

Wu J, Zhou J, Ma J, Mei S & Ren J. An active data leakage prevention model for insider threat. Paper presented at the 2011 2nd international symposium on intelligence information processing and trusted computing (IPTC); 2011. p. 39–42.

Wuchner T, & Pretschner A. Data loss prevention based on data-driven usage control. Paper presented at the 2012 IEEE 23rd international symposium on software reliability engineering (ISSRE), 2012, 27–30 November; 2012.

Yang X & Li C. Secure XML publishing without information leakage in the presence of data inference. In Proceedings of the thirtieth international conference on very large data bases – volume 30; 2004. p. 96–107.

Yaseen Q & Panda B. Knowledge acquisition and insider threat prediction in relational database systems. International conference on computational science and engineering, In CSE'09; 2009. p. 450–5.

Yoshihama S, Mishina T & Matsumoto T. Web-based data leakage prevention. Paper presented at the IWSEC (Short Papers); 2010.

Yu F, Chen Z, Diao Y, Lakshman T & Katz RH. Fast and memory-efficient regular expression matching for deep packet inspection. In ANCS 2006. ACM/IEEE symposium on architecture for networking and communications systems; 2006. p. 93–102.

Zilberman P, Dolev S, Katz G, Elovici Y, & Shabtai A. Analyzing group communication for preventing data leakage via email. Paper presented at the 2011 IEEE international conference on intelligence and security informatics (ISI), 10–12 July; 2011. p. 37–41.