Chapter 1

Introduction

In today's fast moving and competitive world, information is considered as an important and vital resource of an organization. *Timely, accurate* and *relevant* information is important for effective decision making to achieve the objectives of an organization. To ensure that, every organization has a set of centres for processing organizational information and such centres are called *offices*. Therefore, an office can be viewed as an information processing centre. An office receives, stores, structures, processes and provides information. Information is thus the basic commodity of an office. Like any other commodity, it is produced, stored and distributed. In a conventional office, information is captured in documents. Thus a document is one of the most essential objects present in an office. Therefore, production, storage and distribution of documents occupy a major portion of office activities.

With the increase of complexity and competition in the world, modern offices have to deal with a huge volume of information. Paper-based offices are failing to serve the basic purpose of ensuring timely, accurate and relevant information. Moreover, paper-based documents consume a large amount of storage space. It also takes longer to retrieve a document and process it.
1.1 Document Production

Document production is a major activity in an office. Development of word processors is an important event in office productivity. A digital document has two aspects to look into: the content and the presentation. The content is captured as normal text encoded with some markup elements for processing and presentation. The markup is an encoding for making a particular interpretation of structure, layout etc. of the body of a text clear and explicit during processing and presentation. There are two types of markup languages: *procedural* and *descriptive*. Procedural markup languages are not human readable and also not application independent. The descriptive markup languages on the other hand are both human and computer readable and application independent. The *Standard Generalized Markup Language (SGML)* is an international standard for the formal definition of device, system, and application independent digital text using descriptive markup. The *Hyper-Text Markup Language (HTML)* is a derivative of SGML. HTML is about *describing* content with annotations for presentation. It lags in its ability to describe the semantics of a document. Recently developed markup language, which is said to define the content of the future Web, is *Extensible Markup Language (XML)*. The tags of XML are user defined and hence flexible. Thus content of a page can be organized more semantically using XML tags. Detailed specifications and standards of these markup languages are available in the web site [www.w3c.org](http://www.w3c.org).

Compound document is another concept increasingly becoming popular. It is a component based software concept. Component software is based on the notion of a component, a reusable object, which can be plugged into other components from other vendors with relatively little effort. When components are used in content-centric documents, the resulting document con-
tains components or parts pertaining to different applications and is termed a compound document. A compound document is a *multi-part document*. A simple example of a compound document is a Microsoft WORD document consisting of a part belonging to a graphic package, another that is a spreadsheet and another containing a paragraph belonging to the word processor. The compound document model with different parts that pertain to various applications has important implications for groupware systems. Perhaps the most important of these is that the individual parts could actually be located on physically distributed sites in a network. Thus the components or parts can inter-operate and be located not only in different geographical site in a network, but also on different operating systems. The two leading proposals for compound document interoperability are *OLE2 (Object Linking and Embedding)* standard from *Microsoft* and *OpenDoc* from Component Integration Laboratory(CILabs) [25].

### 1.2 Document Storage and Retrieval

During the last two decades significant development has occurred in storage technology. In hardware, the development of high capacity storage devices like Microfilm, Microfiche, hard disks, optical disk etc. are the milestones for storing both operational as well as analytical data. In the software side, a milestone in storage and retrieval of information is the development of the concept of databases. The evolution of database systems is marked by three generations. *First generation* database systems included the *hierarchical* and *network* database systems . These systems are implementation dependent. *Second generation* database systems included the *Relational Database Management System (RDBMS)*. RDBMS introduced the concept of data inde-
pendence. Third generation database systems, consisting of Object-Oriented Database Management System (OODBMS) and Object-Relational Database Systems (ORDBMS), accommodate complex structures in the data model, thus improving the expressive power of the model. However the increased expressive power also implied an increased complexity of the query languages [35]. In addition to these three generations, consisting mainly of generic databases, emergence of domain specific database systems in recent years pave the way for fourth generation systems. This generation includes document, active, spatial, temporal, multi-media database systems. Out of them document or text database systems are of special interest for office automation. Document databases are deserving more and more attention, due to their diverse applications: World Wide Web, paper-less offices, digital libraries etc. Research on document database models emphasize for efficient storage and retrieval. The pertinent problem is to extract relevant information from these documents. Documents may be unstructured or semi-structured.

1. Unstructured: Unstructured documents are the plain texts or ASCII texts without any form of tagging or structural information. The unstructured document is assumed as a sequences of words or phrases including keywords and stop words - often referred as terms. In conventional Information Retrieval(IR) methods, documents are retrieved against a query by matching keywords belonging to the document with those belonging to the query, defined as a Boolean combination of keywords. Vector space model [16], signature file model, inverted file model [11] etc. are the standard works based on this approach.

2. Semi-Structured: Researchers have found semi-structured data to be different from fully structured data, like relational or object oriented
It is defined as irregular or incomplete data and whose structure change rapidly and unpredictably. The Lore project, started in Stanford University around 1995, (http://www-db.stanford.edu/lore), is a pioneering work in this direction. The major contributions of the project are: a complete database management system for semi-structured data, a schema-less self-describing data model, called Object Exchange Model (OEM) and a query language, called Lorel. The first public release of XML version of Lore was made in May 1999. Recent works, like SGML document databases[35] and XML document databases[40], also belong to this approach.

### 1.3 Data Warehousing and Data Mining

Another significant development is a conjugate pair of concepts: data warehousing and data mining. A data warehouse is a wider perspective of a database. Data warehousing is the process of integrating enterprise-wide corporate data into a single repository from which end-users can easily run queries, make reports and perform analysis. The basic criterion is that a data warehouse holds read-only data where as a normal database holds operational data. A data warehouse is important for heterogeneous database integration. Many organizations typically collect diverse kinds of data and maintain large databases from multiple, heterogeneous, autonomous, and distributed sources. To integrate such data, and provide easy and efficient access to it is sought to be done through Data Warehouses[28].

The term data mining refers to the finding of relevant and useful information from databases. Data mining in databases or data warehouses is a new interdisciplinary field with the merging of ideas from statistics, machine
learning, databases and parallel computing. The fundamental goals of data mining are prediction and description. Prediction makes use of existing variables in the databases to predict unknown or future values of interest and description focuses on finding patterns describing the data and the subsequent presentation for user interpretation.

1.4 CSCW and Groupware

In the next phase, we observe automation of group work. As a result, a new field of identifiable research in computer science has emerged, where role of computer in group work is focused. The new field is called Computer Supported Cooperative Work (CSCW). CSCW [17] is defined as computer-assisted coordinated activity, such as problem solving and communication carried out by a group of collaborating individuals. CSCW addresses the organizational issues in collaborative work done by a group of individuals and the multi-user software supporting such collaborative work is termed as groupware [6]. Groupware represent a paradigm shift for computer science, one in which human-human rather then human-machine communications and problem solving are emphasized. The paradigm shift has resulted from a number of converging phenomena, like, pervasive computer networking, workgroup computing, increasing interest in telecommuting, electronic mail etc. Groupware is distinguished from normal software by the basic assumption it makes: groupware makes the user aware that he or she is a part of a group, while most other software seeks to hide and protect users from each other. The major group work supported by groupware are coauthoring of documents, conferencing, meeting scheduling etc. The majority of CSCW applications are fundamentally distributed and are dependent on the
facilities provided by the existing distributed computing platforms. People cooperate synchronously and asynchronously. Synchronous cooperation requires the presence of all cooperating users, while asynchronous cooperation occurs over a longer period of time and does not require the simultaneous interaction of all users. A traditional problem with cooperation in distributed systems is the need to recognize the autonomy of individual sites in a network. Indeed, full cooperation and full autonomy are actually two extremes in a spectrum of possibilities. Increasing autonomy of a system decreases the support for cooperation and vice-versa [7].

1.5 Workflow

We have seen that the initial stage of automation of office work is to use computers for word processing. The next stage is to use a database to store information. The current trend is to move towards what is termed *work processing*. One of the major component of work processing is *workflow automation*. It is a major component of CSCW. An office work comprises of a set of tasks. The genesis of workflow automation is in accomplishing the tasks of an office work in a predefined order by routing the objects of work in the predefined routes following predefined rules by some roles. Rules define both the conditions, the workflow must meet to traverse to the next step and how to handle exceptions. Roles define job functions independent of the people who do it. A *Workflow Management System (WfMS)* is a software system that supports the specification, execution and management of workflows. A job in a workflow system is known as a *case*. Workflow coordinates user and system participants together with appropriate data resources, which may be held on- or off-line to achieve defined objectives. The coordination involves
passing tasks from participants to participants in correct sequence, ensuring that all fulfill their required contributions and take default actions when necessary [20].

A WfMS is a client/server application where the client is called a workflow client and the server is called the workflow engine. The workflow client contains a workflow description tool for designing a workflow template and a workflow activation tool for activating the workflow template. Handshaking between the client and the workflow engine when starting a workflow, terminating a workflow and suspending a workflow, is done by the activation tool. The workflow tracking tool displays the status of various active workflows including the time taken to complete various tasks. The workflow engine provides workflow management services like interpretation of workflow templates, route management, rule management, workflow tracking management, user and role management etc. It interacts with workflow databases through interfaces and uses persistence and concurrency control capabilities of relational DBMS or object-oriented DBMS to allow workflow objects to be defined, created, searched and updated [20]. Workflow is today considered as the heart of E-Business. Some of the common workflow software are Office.IQ, WorkMAN, Visual WorkFlo, CabinetNG etc. Lotus Notes/ Domino and Oracle contain a WfMS as an important component. Standardization of WfMSs is done by the Workflow Management Coalition (WMC) formed in 1993. It is a field of active research now.

1.6 Security

Security is a major concern in an office environment in which the computing is usually distributed. The security aspects in an office are not limited to secure
transmission and reception of documents. Documents have to be signed and they have to be stored securely. The proof of receipt and the proof of sending documents have also to be stored securely. With documents having a long life time, the issue of repudiation of signatures has to be handled. There are significant advancements made in cryptography during the last few decades and office software today uses these cryptographic concepts. Office related security concepts are discussed below. The security in a digital office can be discussed under the following heads:

- **User Authentication:** User authentication establishes a level of confidence about the user’s identity. The main objective of authentication mechanism, in general, is to identify an entity uniquely and unforgeably.

- **Document Security:** Once a user has been authenticated, how does a recipient of a document know that it originated where it says it? How to ensure that the content of the document has not been tampered with? How to resolve the cases of repudiation of signing, sending and receiving the documents? How to authenticate the time of signing? How to maintain the confidentiality of the document? Document security addresses these issues.

- **Storage Security:** The office documents normally have a long life time. The security of the persistent office document and the evidences of occurrence of events on office documents like signing, sending, receiving etc. during storage is provided by storage security using encryption and proper authorization mechanisms.

- **Transport Security:** Since documents will be transported from one point to other geographically distant place through computer networks,
it should be resilient to network attacks. Hence we need a secure channel for document flow across a network.

1.6.1 Digital Signature

In case of paper documents, handwritten signatures on the document resolve the issues of document security to a legally acceptable level. From the difference of handwriting, ink used etc., forgery cases can be detected. Confidentiality can also be provided using sealed envelopes. Handwritten signatures provide only an imperfect solution to these requirements. It has several weaknesses. The weaknesses are: forged signatures are very hard to detect without genuine samples to compare with; it does little to prevent the alteration of a document: that is, it cannot maintain content integrity. Witness signatures are often added to a document to authenticate the main signature, but they suffer from similar weaknesses. Despite these imperfections, handwritten signatures are widely used as an authentication technique for paper documents. Equivalently, for digital documents we have several digital signature schemes [34, 39, 26, 22]. Digital signatures are analogs of handwritten signatures. The ability to provide a digital signature depends on there being something that the principal, who is the original signatory can do that others cannot. Confidentiality can also be provided by using a sealed digital envelope, created by encrypting the document using suitable encryption keys. A digital signature of a message is a number, dependent on some secret, known only to the signer, and, additionally, on the content of the message being signed. Signatures must be verifiable. If a dispute arises as to whether a party signed a document, caused by a lying signer trying to repudiate a signature it did create, or a fraudulent claimant, an unbiased third party should be able to resolve the matter equitably, without requiring access
to the signer's secret information. Digital signatures addresses the issues of user authentication, content integrity, non-repudiation and certification.

Public-key cryptography is generally used for digital signatures. Each user has a pair of conjugate keys: a secret key and a public key. The secret key is known only to the user concerned and the public key is public, that is, known to all. A message encrypted with the secret key can be decrypted with the conjugate public key or vice-versa. Let $A$ and $B$ be the two communicating users. $s_A$ and $p_A$ are the secret key and public key of $A$ respectively. Similarly, $s_B$ and $p_B$ are the pair of keys of $B$. $B$ knows $p_A$ of $A$ and $A$ knows $p_B$ of $B$. Also, $\{m\}_k$ denotes message $m$ is encrypted with a key $k$ and $\{m\}_{s_A} p_A = \{m\}_{p_A} s_A = m$. $A$ can send the signed message to $B$ by transmitting $\{m\}_{s_A}$. On receipt $B$ can verify the signature of $A$ by decrypting the signed message with $p_A$ to get $m = \{m\}_{s_A} p_A$. Confidentiality can also be incorporated with the signed message by transmitting $\{m\}_{s_B} p_B$ so that only $B$ can read the message. Since $s_B$ is known only to $B$, $B$ can decrypt the transmitted message with $s_B$ and then verify the signature. Now, so far as efficiency is concerned, encryption in public key cryptography is much slower than the encryption of symmetric cryptography. Therefore, instead of the entire message $m$, a digest $\delta_m$ is encrypted with the secret key of $s_A$ of $A$. This serves both the purposes of origin and content integrity of the message. A message digest is a fixed-length bit string computed from the arbitrarily long message using a one-way hash function. It is discussed in detail in [34, 22].

Let $\delta_m = h(m)$ be the digest of message $m$, where $h()$ is a one-way hash function. $A$ can now transmit the signed message $\{m, \{\delta_m\}_{s_A}, k\}_{p_B}$, where $k$ is a randomly generated symmetric key. $B$ first gets $k = \{k\}_{p_B} s_A$ and then decrypts $\{m, \{\delta_m\}_{s_A}, k\}$ with $k$. $B$ can then verify the signature.
of $A$ and the content integrity of the message. Content integrity is verified by recomputing a digest $\delta'_m = h(m)$ and then checking the equality $\delta'_m = \delta_m$. In principle, any public-key cryptographic scheme can be used for digital signatures. More details on digital signatures can be found in [39]

### 1.6.2 Multi-Signature

In addition to the originator's signature on the digital document, supervisors are often required to sign office documents for verifying and approving an originator's message. In such cases several persons sign the same document. This is referred to as a multi-signature. Proprietary digital signature schemes cannot resolve the issues related to such multi-signature documents. As a result, the literature contain different multi-signature schemes. Signature schemes originally developed for single signatures, are also extendable to the multi-signature case. However, because of the increase in signature length, they are not satisfactory for use. Itakura and Nakamura [19] proposed a solution based on extended RSA scheme and resolved the problem of signature length. But their schemes needs to predetermine a hierarchical relationship among users. In some offices the hierarchical relationship either does not exist or cannot be predetermined in advance. Okamoto proposed a scheme [24] that overcomes these problems. In this scheme the signature length of a multi-signature in nearly equal to that for a single signature and the order of signing is not restricted. But it introduces the problem of key distribution. All the persons who are communicating among themselves should know one another's public key. This leads to the distribution on $n^2$ keys, which keeps on increasing exponentially. It puts a lot of processing overhead on the users. A user who is $n^{th}$ on the list of persons reviewing the document then $s/he will have to decrypt the document $n - 1$ times with the public key of all
the previous persons. The verification process is time consuming because a recipient must check the multi-signature by the reverse order of signing. Similarly Harn and Kaisler [18] also proposed a scheme of multi-signature. The issues addressed in the above schemes are mainly related to the length and order of signatures.

1.7 The Problem

From the above discussion, it is evident that, with the development of information technology different directions of office automation have received considerable attention. The technological infrastructure for a paper-less office is almost ready and there is rapid progress in both software as well as hardware aspects. But a vital and central problem, common to almost all offices, is missing from the research agenda on office automation. The problem is the production of Multi-Part Multi-Signature Documents (MPMSD): It is different from the multi-signature problem and the multi-part compound document problem discussed in the literature. In case of multi-signature or group-signature, a group of users can sign a single message. Here, content of the same message is authenticated by multiple people. For example, the minutes of a meeting is signed by all the participants present in the meeting. But in case of multi-part multi-signature each member of the group of signatories contributes a different message which is authenticated by the member by signing the message.

In this section an outline of the problem for the present research work is provided. The issues of the problem to be addressed are also identified and discussed. It is not simple to describe all the work performed in an office in a common framework without referring to the specific organization. But
document production and storage is a common work in almost all offices. The scope of our discussion is limited to this common work within an office. Document production in an office is based on a request-reaction-response paradigm. When a document containing a request is received in an office, the office reacts to the request. The reactions are recorded in the form of comments on the document and finally a response document is dispatched. We can term the process as Document Production Workflow (DPW) [37]. The resultant document of a DPW is termed as a Multi-Part Multi-Signature Document (MPMSD) [36]. Therefore, a MPMSD is a case of the DPW. The first part of a MPMSD is the request document and the last part is the response document and the other parts in between are the comments of other reviewers, that means, the reactions. Each part of a MPMSD is signed by the corresponding reviewer. The first reviewer is also termed as the originator of the request.

1.7.1 A Scenario

To understand the salient features of a DPW in an office let us consider the following scenario of a DPW in a University. An employee, $A$ submits an application, $m_A$ regarding her travel plans for approval to the head, $B$ of the department. $B$ verifies the travel plans in the context of previous cases of employees from the department already in travel, type of leave to be granted for $A$ during travel, resolutions on travel taken in departmental advisory committee, standing rules, etc. and adds her comment, $m_B$ and forwards it to the finance officer, $C$. $C$ also examines the case by verifying the budget allocation status under the head of account for travel, TA/DA rules in such cases, circulars from University Grants Commission on travel expenditure, and adds her comment, $m_C$ on the amount that may be granted and forwards
it to the director, D. D also justifies the previous comments, approves the travel plans and adds the note of approval, may be in the form of office order, \( m_D \). A copy of the whole multi-part document or only the office order \( m_D \) may finally go back to the originator, A and the original multi-part document is stored in a folder. The flow of the document is recorded in log-books. This is a case of the travel plan workflow. It is shown in figure 1.1

![Figure 1.1: Travel Plan DPW](image)

1.7.2 The Components of the Problem

The DPW has three components to study:

- **MPMSD**: It is the major component of a DPW. Since in our framework, an office document is produced as a case of a DPW, therefore, all documents are MPMSDs. It is a generic framework. Different parts belonging to different cases of different DPWs may also be integrated
to form a MPMSD subject to satisfaction of certain criteria. For example, office orders, circulars, meeting resolutions produced as parts of cases of different DPWs but containing rules on a certain topic may form a MPMSD. Moreover, a document may have multiple versions produced at different points of time. Such multi-version documents are also MPMSDs. In a paper document system, it is the same paper document that is passed around and the proof that it has come through the proper channel is the series of comments followed by the signatures of the reviewers. In a digital system, there are several issues to be addressed for secure production and storage of MPMSDs. The issues to be addressed are identified and discussed in detail in the following sections. A single part document is a special MPMSD, where the total number of parts in the document is equal to one. Henceforth, in the rest of the discussion, a document means a MPMSD.

- **Context:** Just as a human being can develop amnesia and forget past experiences, an office can also experience loss of memory unless there is a proper framework to maintain organizational memory[15]. Huge collection of documents in an office is the major constituent of its organizational memory. Contemporary offices have only a weak ability to remember and learn from the past. What is missing from organizational memory is the context or rationale that lay behind these documents when they were created. In an office a new document is produced in the context of a set of existing documents constituted of rules, precedents and other support documents. In a formal office rules are framed almost on all topics to prevent the possibility of arbitrary decisions. Rules are generally well defined. When rules are either not defined or not well-defined we look for similar cases handled earlier,
that is, precedents. Here, rules include regulations, office orders, meeting proceedings etc. and the precedents are the already produced cases following concerned rules. Certain decisions require support documents. For example, a purchase indent to sanction a purchase. Rules, precedents and support documents constitute a reference space. A reviewer navigates through a subspace of the reference space before producing a new document and draws citations wherever necessary to substantiate the rules position, the precedent position etc of the new document. This subspace is called the context of the document. The process of navigation through the context is called the case examination.

- **Linking:** Links establish relationships among different documents as well as different parts of a document. As soon as a document is created in an office, the new document may be implicitly linked to many documents. Moreover, a document may also be explicitly linked to many more documents at a future point of time. For example, documents on the same topic or of the same type or created within a certain period are implicitly linked, whereas document cited in another document are explicitly linked. Therefore, linking is an important component to be studied in an e-office.

### 1.8 The Security Issues

There are several security issues related to secure production and storage of digital documents. Some of the issues are general in nature and some are specific to DPW system. The security issues are discussed below.
1.8.1 Principal Authentication Issues

A fundamental concern for a secure office system is the authentication of the principals involved in the system. Authentications of the principals are usually done by using their credentials. A credential is a piece of information that is used to prove the identity of a principal. Passwords, digital certificates, secret keys etc. of the entities are the important credentials. In the office system the principals are the users and the processes. Therefore, the issues are:

1. *User Authentication:* All the office workers (users) of the system need to be authenticated through standard challenge-response protocols during session set up using the credentials of the users.

2. *Process Authentication:* Processes are the entities who speak for users during run-time[21]. A set of processes who speaks for a user may share the credentials of the user. Processes are to be authenticated when they try to access any object.

1.8.2 The Production Security Issues of a Part

For every individual part of a MPMSD, the security issues are as follows:

Let \( A \) and \( B \) be legitimate principals and let \( A \) send a signed message \( m_A \) to \( B \). The issues are-

1. *Proof of Origin:* It should be verifiable by \( B \) or any third party that the message \( m_A \) was really signed by \( A \) and not forged by an intruder.

2. *Content Integrity:* It should be verifiable by \( B \) or any third party that the content of the message \( m_A \) was not illegally modified by a intruder. Even the originator of the document, \( A \), is not allowed to modify its
content after it is dispatched to $B$. The first part of this issue can be taken care of by digital signatures using fixed length message digests generated by one-way hash functions. The second part is the more interesting point to look into in office automation.

3. **Confidentiality**: It is required that the message $m_A$ be accessible for reading only to the authorized principal $B$, to whom it is addressed, and not to any eavesdropper.

4. **Repudiation of Signing**: The success of a digital signature scheme using public-key cryptography pivotally depends on the secrecy of the secret key. Even if the message $m_A$ is signed by the secret key of $A$ and successfully verified as in issue 1, $A$ can repudiate the signature with the pretext of compromise of the secret key of $A$ and can thus disown the responsibility for $m_A$.

5. **Repudiation of sending and receiving**: If $A$ or $B$ repudiates the sending or receiving respectively of the message $m_A$, then it should be verifiable by any third party from the stored evidences of the flow of the message. The evidences may be the proof of sending and the proof of receipt. The evidences should be acceptable as legal and irrefutable proofs. The repudiation may be on the time of sending or of receiving the message. Moreover, $A$ and $B$ may collude to remove the evidences of transmission of the message. Therefore, the records should also be tamper-proof.

6. **Signature Replacement**: If another principal $X$, with the cooperation of $B$, tries to replace the digital signature of $A$ on $m_A$ by its own signature and claim the ownership of $m_A$ then such an issue should be resolvable. In digital signature schemes, based on public-key cryptography with
one-way hash functions, this issue is not addressed. In online communication this can be taken care of by a digital envelope which ensures the secured transmission and reception of message $m_A$ and digital signature $\sigma_A$ on it but in the problem domain with persistent storage, it is not sufficient. Even the recipient $B$ can replace the signature with its own since there is no certification of association of $m_A$ and $\sigma_A$. The message digest $m_A$ of $m_A$ included in $\sigma_A$ can be created by anybody having $m_A$ since the digest function is public.

1.8.3 The Production Security issues of the Whole MPMSD

Apart from the security issues of each individual parts of a MPMSD as mentioned in section 1.8.2, the following special security issues related to a MPMSD as a whole are to be addressed additionally.

1. **Part integrity of a MPMSD** : Apart from the content integrity of each part, we also need the total part integrity of the whole document. The content integrity of all parts individually does not necessarily imply the integrity of the whole document. All parts must remain in order in which they were added to the document. Removal of some parts and reordering of parts should not be allowed.

2. **Reuse of Parts** : Reuse of parts should not be allowed. Suppose, the ordered list of reviewers of a MPMSD is $(A, B, C, D)$ as in the example given in section 1.7.1. If $D$ does not like what $C$ has written on the document $(m_C)$ then $D$ may cooperate with $B$ to have $B$ mark the document directly to $D$, bypassing $C$. $B$ can do this by using the document $m_A || m_B$ passed to it by $A$ and reusing it.
1.8.4 The Storage Security Issues

1. **Authorization Flow:** During the production of MPMSDs, the authorization also flows synchronously with the document flow. Only the latest reviewer of a document can add a part to the document and only she can read the previous parts but cannot modify or reorder the previous parts. After signing and forwarding the document to the next reviewer of the document the current reviewer loses the privileges of being the latest reviewer. The author of a part is not allowed to modify her previous parts, if any, even if she is marked again as the latest reviewer. Moreover, a user has privileges as long as he/she is assigned to the role of a reviewer of the DPW.

2. **Authorization Constraints:** During case examination, a reviewer accesses the documents belonging to the context. But the accesses are subject to some authorization constraints. A reviewer may be allowed to read the precedent cases but may not be allowed to read all the parts of it. For example, the originator, that means, $A$ in the scenario given may be allowed to see only the last part, the office order $m_D$ and her application, $m_A$, but not the intermediate comments. Some reviewers may not be allowed to see all the comments given by the higher authorities. Similarly, all rules and other support documents may not be accessible to all reviewers. It depends on the security policy of the office concerned. But the system should have provisions for such authorization mechanism.

3. **Secure storage of session keys:** Session keys play an important role in our system both for digital signatures and secure transport of MPMSDs. Generation of a session key should be unique for a particular user, be-
cause a session key speaks for a user both during the session and in future for persistent documents. Signed copies of the session keys should be stored securely.

4. **Secure storage of document flow records:** The records of the flow of documents along with the evidences of non-repudiation are to be protected from unauthorized access. The records need to be well structured to enable efficient tracing of a document with its latest state, that is, at a particular time where the document is lying and at what state. The tracing of documents is to be allowed only to authorized users.

5. **Security of program codes:** The program codes designed to implement the system are to be stored securely in storage. They are to be authenticated using the credentials during loading.

6. **System administrator threat:** The system is on top of the OS of a host system. Even though the operating system of the host system is assumed to be trusted the administrator of the host system should not be allowed to access the objects of the system without proper authorizations.

7. **Validation of Old Documents:** If the principal $A$ feels that her secret key has been compromised and the key is consequently changed then all the documents signed by $A$ using the earlier key will be invalid. Now, the issue is how to validate such old documents and how to disallow the use of the old key of $A$.

8. **Archival of Old Documents:** The documents no longer active and not referred frequently are to be archived as normally done in the record rooms of an office. The archived documents will have life tags attached
signifying the permanence of the information stored in the document. After the expiry of the life, the documents may be deleted from the system. Only archived documents may be deleted from an office.

1.9 The Management Issues

Apart from the security issues there are important management issues to be addressed.

1.9.1 The Context Management Issues

1. **DPW Context:** With every DPW, a default initial context is attached. A reviewer can add more items to it during case examination, but cannot remove any item from the existing context. As a result, the context may grow as the document flows from the current reviewer to the next. Moreover, during case examination, a new relevant rule or a new version of the existing rule, a new case or a support document may come up, which should be automatically added to the concerned contexts. Therefore, a DPW context consists of all the documents relevant to the DPW in general.

2. **Drawing Citations:** During composing a part, a reviewer should be able to draw citations directly from the context, so that a hyper linked address of the cited document is automatically included in the comment.

3. **Case Context:** The DPW context provides the documents relevant to all the cases of the DPW. The DPW context attached to a DPW changes with time. The context of a case is the context specific to the case. It consists of the DPW context and some more documents relevant to the
specific case. Moreover, a case is a MPMSD. Different parts of the case may be produced at different points of time. Therefore, the context of a part of a case consists of the state of the DPW context and the other documents included in the case context at the time of production of the part. During case examination, when a reviewer peruses a part of a precedent case, he/she should be able to retrieve the state of the case context at the time of creation of the part of the case. The state of the case context is defined by which rules and which versions of the rules, which precedents and support documents were available at the time of production of the part.

1.9.2 Storage Management Issues

Apart from the security issues of stored documents there are a few more management issues to be addressed.

1. **Organization of Documents**: An office has a huge volume of documents. Tracing a particular document and then retrieving it for perusal in an efficient way is a major issue in any office. Therefore organization of documents is a very important aspect. In a paper-based system office documents are organized in folders and folders in cabinets etc.. Within a folder documents are organized either as a stack or as a queue. As a result we get a linear organizational structure. Related documents are stored in the same folder. But, as the relations among the office documents are non-linear and complex in nature we find unnecessary duplication of documents and inefficient retrieval of documents. For efficiency, the organizational issue is to be addressed.
2. **Invariant Document Address**: During the life-cycle of an office document, it has certain degrees of mobility. The mobility may be attributed to the documents due to document flow, archival activities and system management. Therefore, address of the document may change from time to time. Since many documents may be hyper linked to a particular document, change of all such hyperlinks as soon as the location of the document changes, will be an inefficient proposition, if not impossible. Therefore, an invariant document address is required for mobile office documents.

3. **Performance**: In course of time, the volume of office documents will be large which may lead to performance degradation of document storage and retrieval. Therefore, measures are to be taken for enhancing efficiency.

4. **Reliability**: The reliability of the storage system is a pivotal issue. The entire system depends on it.

5. **Reverse Linking**: An explicit link comprises of a conjugate pair of directed links: one forward and one reverse. As soon as a forward link is established, the reverse link is to be established automatically. For example, an office document may be cited in many different documents at different points of time. If we want to study the effects or the reactions on the office order within the office then such reverse linking is a fundamental requirement.
1.10 Scopes and Goals

Almost all the work done in different directions of office automation discussed above have failed to address the DPW problem. With the advent of workflow solutions, similar problems are coming to focus. But the genesis of workflow automation is on the automatic routing of documents and the automatic execution of the tasks based on the rules and roles defined in the workflow design. It is rigid in nature. As a result we see success of workflow systems in practical application areas like manufacturing, where the flow of work is almost static. But what is required in a real life office today is a flexible, co-operative workflow tool which assists the office workers in reviewing the documents by providing facilities for secure production, storage, case examination and flexible routing. Our study aims towards incorporation of such a tool in future office automation software. A similar problem was studied in the POLITeam project [27]. It addressed the problem of multiple versions during the production of a speech in a German ministry. We found that this multi-version speech production workflow is a special case of our more generic framework of MPMSD where each part is a version of the speech. Single-part documents discussed in the literature for office automation are special cases of multi-part documents. Secure production and storage of persistent multi-part document constitute the major part of office work. But almost no effort has been made to solve this problem.

As the above discussion shows, digital signature schemes developed for general digital documents addresses the security issues of single-part documents but do not address the security issues of multi-part documents. The multi-signature schemes developed till date also do not address even the basic issues of MPMSD. Most successful, widely used, and a robust groupware product Lotus Notes/Domino also does not address the issues of MPMSD.
Therefore, there is sufficient scope for further research in this field.

The main goal of our research is to identify the issues of persistent multi-part digital documents in an office environment during production and storage and provide solutions to address the issues. The output of our research may be the input to future commercial software for paper-less office. We provide a design framework for multi-part document system and protocols to address the issues of multi-part documents using a trusted third party, called an arbiter. We also provide a storage model for automatic generation of context of a workflow. The present work does not provide a complete solution to a paper-less office. The study is limited only to the problem of persistent multi-part office documents, their production and storage within a single office and under a single arbiter. Moreover, by document in this work, we mean, text only documents, excluding multimedia documents and compound documents.

In the present chapter, we reviewed the directions of office automation, outlined the problem of DPW and identified the different issues of the problem. The rest of the thesis is organized as follows. In Chapter 2 we discuss a production and storage framework and a conceptual architecture for DPW. In Chapter 3 a model, named Page Cube, for storage and retrieval of documents is presented. In Chapter 4, we discuss the security framework and propose a protocol for secure production of MPMSD using a neutral arbiter as an in-line TTP. In Chapter 5, we discuss production of contexts. In Chapter 6, we discuss the aspects of authorization of pages in a page cube. Some implementation issues in relational as well as in XML database models are discussed in Chapter 7. In Chapter 8, we try to answer the question, where does our research work stand in the space populated by the current academic as well as commercial solutions for office automation, like the POLITeam project,
Lotus Notes etc. Some conclusive remarks and scope for further work are presented in Chapter 9.