

# IBM Research Report

## An Automatic Electronic Contract Document Signing System in a Secure Environment

**Thomas Kwok, Thao Nguyen**  
IBM Research Division  
Thomas J. Watson Research Center  
P.O. Box 704  
Yorktown Heights, NY 10598



**Research Division**  
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

# An Automatic Electronic Contract Document Signing System in a Secure Environment

Thomas Kwok and Thao Nguyen  
IBM Research Division  
Thomas J. Watson Research Center  
19 Skyline Drive, Hawthorne, NY 10532  
[kwok@us.ibm.com](mailto:kwok@us.ibm.com)

## Abstract

*Superimposing electronic signing data on a signed electronic contract usually involves a large number of complex manual steps. These manual steps are tedious and may cause security faults. In this paper, we present an automatic electronic contract document signing system in a secure environment. The system comprises a document and signature profile module, a signing text template module and a signing information superimposed module. The document and signature profile module parses the contract document and signature information. Based on this information, the signing text template module creates a signing text template with empty text fields. The signing information superimposed module carry out a novel electronic document signing method. This novel method consists of four processing steps: inserting signing text data into empty text fields, converting the text template into an image template, adding signing image data onto the image template, and superimposing the signing image template on the contract document to produce a signed electronic contract document. This automation system has found to improve substantially the efficiency, and at the same time eliminate the potential for human errors and security breaches in signing electronic contract documents.*

## 1. Introduction

Today, most business between enterprises is conducted under contract. Contracts also constitute the binding relationship between a company and its customers or suppliers. Everyday, hundreds of contracts are created, executed and managed via paper-based manual processes in large enterprises. Automation of the contract lifecycle presents a substantial value creation opportunity for the enterprise. This value stems from improved productivity and security, effectively aggregated contract information, accelerated contract lifecycle processes, reduced contractual errors and risk, enabled revenue forecast and profit optimization, as well as better compliance enforcement [1]. With the advent of Internet technology

and electronic commerce, there are growing research activities and implementation efforts on establishing and standardizing an electronic contract management and processing system. Currently, the International Association of Contract and Commercial Managers has listed over twenty commercial available software products for electronic contract management and process [2]. Most of the research activities on signing electronic contracts reported is focus on digital signature [3-5]. However, none of these studies has provided an automated solution to replace a large number of tedious and manual steps involved in adding electronic signing information on signature blocks of a signed electronic contract document.

In a Web-based electronic contract management and processing system, a user signs an electronic contract on-line by clicking the "I accept", "Sign", or any other acceptance buttons. After all required parties have electronically signed an electronic contract on-line; the signed electronic contract will officially become a legally signed contract document even though none of the required signatures or signing data appeared on its signature blocks. However, there is still a need to add electronic signing data into a signed electronic contract document so that a user can distinguish between a signed and unsigned electronic contract document either as a soft or hard copy. Moreover, most contract administrators and customer representatives want electronic signing data added automatically into signature blocks of a signed electronic contract document to save time, to have a uniform format, and to avoid mistakes and security faults.

There are a number of difficulties in adding signing data automatically on a signed electronic contract document. First, it has to find out automatically what type of contract document as each contract document type can have different numbers of signature blocks. Second, it has to find out automatically what page number and location of a signature block because the same signature block can be in different pages and locations of different contract documents depending on the length of contract contents. Third, the signing data in text formats cannot be added into an electronic contract PDF document that is in an image format without the corresponding text fields. In

this paper, we present an automated system for superimposing electronic signing data onto a signed electronic contract document in a secure environment.

## 2. A Framework for an Automatic Electronic Contract Document Signing System

An architectural framework of an automatic electronic contract document signing system is shown in Figure 1. The system comprises of three individual modules: a document and signature profile module, a signing text template module, and a signing information superimposed module. The framework illustrates interactions between an electronic contract management and processing system and these three different modules. These two Web-based systems can be deployed on the same application server. They can also contain a link to a common remote database. All communications among a user's computer, these two systems, and a common remote database go through one or more secure wired or wireless networks.

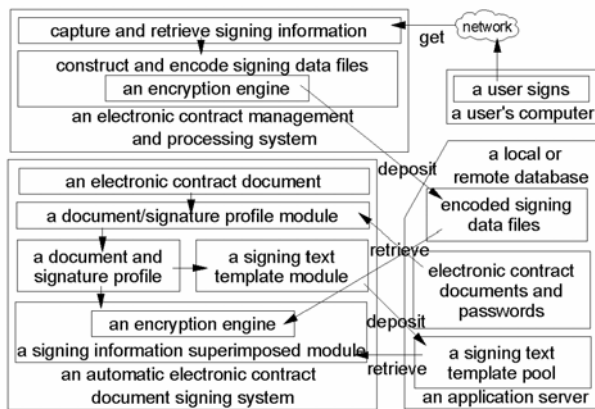


Figure.1 An architecture framework of an automatic electronic contract document signing system.

A typical Web-based electronic contract management and processing system shown in Figure 1 has been described in a previous publication [6]. This particular system merges all related documents of an electronic contract into one PDF document such that all parties involved only need to review, approve and sign a single electronic contract PDF document. When a user signs an electronic contract on-line by clicking the "I agree", "I accept", "Sign", or any similar buttons representing acceptance, the system captures or gathered most of the required signing information from the signer's computer using the URL get or post method through the Internet or other on-line electronic processes. The system also retrieves the rest of the required signing information from his or her records stored in the system's database. The signing information comprises at least the signer's name, the signer's company, and the date or timestamp of the signing. It may also include the signer's system uniform resource locator (URL), the signer's handwritten

signature, and other legally required signing information. Personal biometric information can also be included. For example, a user can sign on a handwriting pad, the captured signature data is the user's signature strokes and the timestamps of each signature stroke. Moreover, the signing information can include the signature of a single party or of a plurality of parties. Thus, the electronic signing information consists of many signing data entries, each of them can be in either text or image format. The system has to construct at least one signing data file. The first signing data file contains all signing data entries in text format. For each signing data entry in image format, the system creates an additional signing data image file.

Once a signing data file has been generated, an encryption engine immediately and automatically encodes it. The encoded signing data file is then stored in a local or remote database of the system for future retrieval. The encryption engine includes a built-in private key to encode the signing data file such that only the corresponding encryption engine with the same built-in private key in the signing information superimposed module can decode the encoded signing data file. The signing information superimposed module implicitly calls the encryption engine to decode an encoded signing data file in order to parse and add signing data entries onto a signing text or image template. Only software codes running on the signing information superimposed module can utilize or apply the private key and decoding methods. As a result, an administrator or a user has no access to and cannot reveal the private key and decoding methods of the encryption engine. This ensures that no one can alter or modify the encoded signing data file. However, an administrator or authorized user of the electronic contract management and processing system is still able to retrieve and decode the encoded signing data file to view the electronic signing information using its encryption engine.

## 3. The Algorithm of a Document and Signature Profile Module

The document and signature profile module automatically reads and parses the required document and signature information from document properties, metadata or barcodes of an electronic contract document into a temporary document and signature profile. The required document information includes the type, format and orientation. The required signature information includes the number of signature blocks, the signature block type and the number of signing data fields for each signature block, the signing data type, the page number and page location for each signing data field. When a user creates an electronic contract document in most text editors or legacy systems, he or she can enter and store

this document and signature information as document properties, metadata or barcodes. If the document properties, metadata or barcodes are not available, the user can manually select or enter the required document and signature information from GUIs or templates provided by this module.

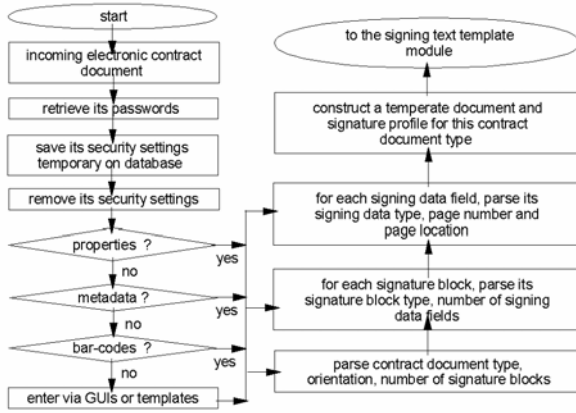


Figure 2. A flow diagram illustrating the algorithm of a document and signature profile module.

Figure 2 illustrates the algorithm of a document and signature profile module in a flow diagram. There are a number of automatic flow steps for the module to construct a temporary document and signature profile. When the system receives a request for adding electronic signing information on an incoming electronic contract document, the module first retrieves its passwords from a local or remote database. Second, the module uses the retrieved passwords to remove or disable one or more security settings so that the electronic contract document can be modified with the addition of the electronic signing information. Third, the module reads and parses the type, format and orientation of the contract document. Fourth, the module also reads and parses the number of signature blocks. Fifth, for each signature block, the module reads and parses the signature block type and the number of signing data fields. Sixth, for each signing data field, the module reads and parses the signing data type, the page number and page location of this signing data field. Seventh, a document and signature profile is generated based on this document and signature information. Finally, this profile is saved and stored as a temporary file in a local or remote database of the system for later use by the signing text template and signing information superimposed modules.

#### 4. The Flow Steps in the Generation of a Signing Text Template

There are two different ways to generate a signing text template corresponding to an electronic contract document type as shown in Figure 3. The first way is to construct a large number of temporary document and

signature profiles corresponding to all electronic contract document types that the system needs to support at the system setup stage. For each new electronic contract document type, the document and signature profile module can automatically construct a temporary document and signature profile using a standard electronic contract document of the same type. Alternative, a user can manually enter the document and signature information for a new electronic contract document type to build its temporary document and signature profile using GUIs or templates provided by the signing text template module. As a result, the signing text template module can generate a large number of signing text templates corresponding to all known electronic contract document types.

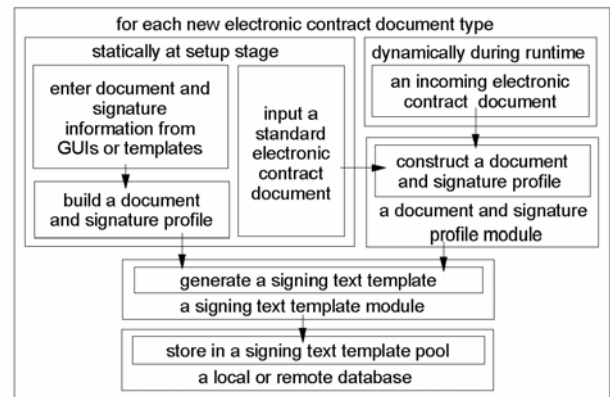


Figure 3. A schematic diagram illustrating the flow steps in the generation of a signing text template.

The second way is to generate a new signing text template corresponding to a new electronic contract document type during the system runtime. Whenever an incoming electronic contract requests the adding of electronic signing information, the document and signature profile module will automatically construct its temporary document and signature profile. If the incoming electronic contract is a new contract document type to the system, this dynamic event will also trickle the signing text template module to generate a new signing text template based on the document and signature profile. Otherwise, this dynamic event will just trickle the automatic retrieval of an existing signing text template corresponding to the contract document type from the signing text template pool.

#### 5. The Design and Architecture of a Signing Text Template Module

All these statically or dynamically generated new signing text templates are stored on the local or remote database of the system for the signing information superimposed module to retrieve, make a copy, and then use the copy for filling the signing data. Figure 4 is a

block diagram showing the design and architecture of a signing text template module. The following steps describe the generation of a new signing text template. First, the module creates a new empty text template for each new electronic contract document type. From the corresponding temporary document and signature profile, the module reads and parses the number of signature blocks. For each signature block, the module reads and parses its signature block type and the number of signing data fields. For each signing data field, the module also reads and parses its signing data type, page number and page location. Then, the module adds an empty text field and an associated hidden label named after the signing data type on the same page number and page location of the newly created text template. After a new signing text template has been generated, it is stored in a signing text template pool in a local or remote database of the system for future retrieval and later use. The signing text template is a contract document type-specific. There is an empty text field and an associated hidden label in a signing text template for every signing data field that exists in a corresponding electronic contract document. These empty text fields are to be filled with signing data text entries from a signing data text file by the signing information superimposed module.

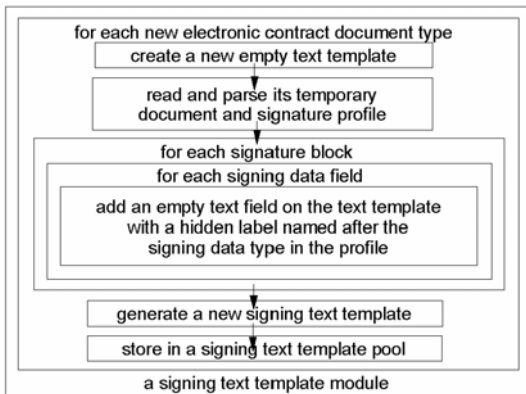


Figure 4. A block diagram showing the design and architecture of a signing text template module.

## 6. The Algorithm of a Signing Information Superimposed Module

Documents related to an electronic contract can be created in text format by any text editors. They can also be generated statically or dynamically in either text or image format by a number of legacy tools. In addition, some of these documents can contain images in image format even though they themselves are in text format. Moreover, some of these documents can be received or reproduced in image format from facsimile machines even they are originally created in text format. An electronic contract management and processing system merges all these related documents in either text or image

format into a single electronic contract PDF document. Thus, an electronic contract PDF document can consist of contents in both text and image format. It is rather difficult to determine automatically the format of an incoming electronic contract PDF document requesting the superimposing of the electronic signing information during the system runtime.

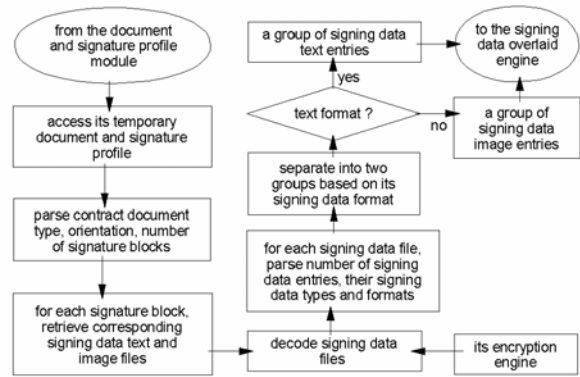


Figure 5. A flow diagram illustrating the algorithm of a signing information superimposed module.

As described in previous sections, the signing information consists of signing data entries in both text and image formats. A signing data entry in text format can only be added into an electronic contract PDF document in text format by text insertion technique while a signing data entry in image format can be added into an electronic contract PDF document in either text or image format by overlaying image technique. However, an electronic contract PDF document can be in both text or image format. Figure 5 is a flow diagram illustrating the algorithm of a signing information superimposed module. Whenever an incoming electronic contract requests the superimposing of the electronic signing information on its electronic contract PDF document, the signing information superimposed module parses the contract document type and the number of signature blocks by accessing and reading this temporary document and signature profile. For each signature block, the module has to retrieve and decode at least one corresponding signing data file using the module's encryption engine. The first retrieved signing data file contains all the signing data entries in text format. Any additional retrieved data file contains subsequent signing data entry in image format. The module also reads and parses the number of signing data entries, their signing data types and formats for each signing data file. Then, the module separates these signing data entries into two groups depending on their signing data formats. The first group consists of all the signing data entries in text format while the second group consists of the remaining signing data entries in image format. These two groups of signing data

entries will be automatically feed into the signing data overlaid module for process.

## 7. The Algorithm of a Signing Data Overlaid Engine

A signing data overlaid engine involves a novel method to superimpose the signing information on an electronic contract PDF document in a secure environment. It utilizes a signing text template with empty text fields as a dummy document in text format, and a signing image template as a dummy image document in image format. First, it inserts each signing data text entry in text format as a text string into an empty text field in a signing text template. In addition, it converts the signing text template with signing data text entries into a signing image template. Then, it overlays each signing data image entry in image format as an image format file on the signing image template. By doing that, the signing image template with the signing information can be superimposed on any incoming electronic contract PDF document in both text and image format in a manner similar to adding a watermark. Moreover, it has the capability of adjusting the position of a signing data text in a signing text template or a signing data image in a signing image template. Thus, it is not necessary to assume that electronic contracts of the same contract document type always have the same signing data field on the same page and in the same page location.

Figure 6 is a flow diagram illustrating the algorithm of a signing data overlaid engine. After parsing the contract document type of the incoming electronic contract from its temporary document and signature profile, the engine retrieves a signing text template of the same contract document type from a pool of signing text templates in the local or remote database of its system. For each signing text data entry, the engine inserts the signing data text entry into an empty text field with the same label in the signing text template. In addition, the engine adjusts the page location of this signing data text entry in the signing text template according to the page position of its corresponding signing data field in the electronic contract document it these two page positions so not match each other. After all the signing data text entries have been inserted into the signing text template, the engine converts the signing text template with all the signing data text entries into a signing image template. Then, for each signing data image entry, the engine adds the signing data image entry on the signing image template at a page location of its corresponding signing data field with the same label in the contract document. Moreover, the module superimposes the signing image template filled with all the signing data entries on the electronic contract PDF document as a watermark. Finally, secure

protections from alternations or modifications are added on the signed electronic contract PDF document with the signing information to ensure that its contents and signing data entries cannot be altered or modified. The secured and signed electronic contract PDF document is deposited back to the local or remote database of the system for future reference or retrieval.

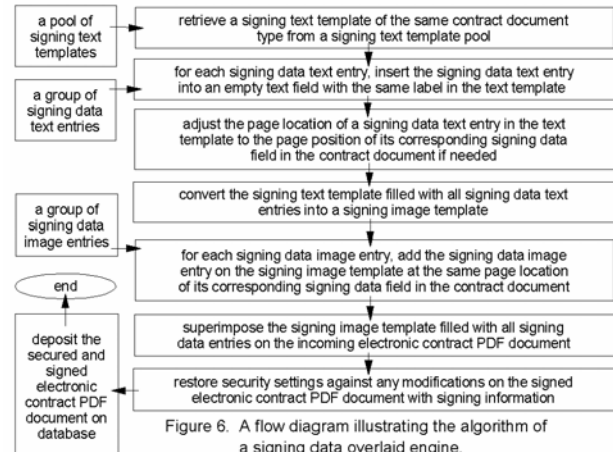


Figure 6. A flow diagram illustrating the algorithm of a signing data overlaid engine.

All processes in this novel method can be carried out automatically in a tight secure environment using any scripts or application programming interfaces (APIs) packaged together as functions or subroutines for all the predefined tasks with input parameters parsed from the configuration file. Moreover, task execution may be performed one-by-one, in a specific or random order, or simultaneously. Once all the necessary tasks have been executed, the system restores the security settings that were disabled or removed from the electronic contract document when it entered the system to prevent its contents from future modifications before re-files or deposits back on the local or remote database of the system. Thus, this novel method enables the superimposing of signing information on any electronic contract PDF document without the need to determine its format (*e.g.*, text, image, etc.). It also eliminates the need to manually format the signing information or locate the page and the page position of each signing data field in the electronic contract document. Since all processes or steps in this novel method are carried out automatically in a secure environment, it provides security to the signing information and their signing data files.

## 8. Implementation and Industrial Experience

Some of the features and functions of this automatic electronic contract document signing system have been used and implemented in a set of electronic contract managing and processing tools in IBM. Among all the functions and features, the signing data overlaid engine is found to be the most difficult one to implement. As a

result, we have tried several methods to implement the execution of predefined tasks in this engine. We have tried to use application programming interface (API) calls as execution instructions for a predefined task. We have also tried to use COM objects from Microsoft Visual Studio 6.0 and Visual Studio .Net for integration with other applications or as a communication interface between other applications and this tool set. We have found that Microsoft Visual Studio .Net has more COM objects available and provides more interfaces for the integration.

This tool set is being used and tested as a pilot program by several dozens of Customer Sales Representatives (CSRs) working on electronic contracts in IBM. Our industrial experience has confirmed the stability and usefulness of this tool set based on the initial feedbacks from CSRs using this tool set. It has found to save CSRs' time, increases their efficiencies and productivities, such as automatically adding the electronic signing information on the signed electronic contract PDF document. Security faults caused by missing or adding the wrong security settings on electronic contracts have been eliminated because the required manual steps for CSRs are replaced with an automatically task.. It has also found to improve customer satisfaction by using in a standardized format on superimposing the electronic signing information on the signed electronic contract PDF document.

## 9. Conclusion and Discussion

The electronic contract document signing system described in this paper provides an automated and efficient way for a user to superimpose the electronic signing information on a signed electronic contract PDF document in a secure environment. The system has solved a number of difficult problems involving the adding of signing data fully automatic to all types of electronic contract documents in all kinds of formats including text and image. The system allows only authorized users, such as administrator, to access the signing data files, thereby preserving the integrity of the signing information. Moreover, the system substantially eliminates the potential for human errors and security breaches in the performance of on-line signing electronic document tasks. Many contract administrators may also want to create a new document of one to two pages long containing all electronic signing information. This new document is then merged as a part of a signed electronic contract document.

The novel method used in the signing data overlaid engine to superimpose the signing data on the electronic contract document can also be used to cover any signature blocks, text or image blocks that are duplicated or simply

need to be removed from the contract document. The system can also include a business process for resale of an electronic contract document. For example, the system can limit the original copy to just one copy by adding "copy" or "final copy" or other watermarks to all other copies to avoid third party or business partners for resale.

However, not all necessary or required predefined tasks can be represented solely by API calls. In this case, we have to use a combination of API calls and sending a sequence of keystrokes as execution instructions for some predefined tasks. These keystrokes are the emulations of some manual steps performed by a user on the application with which this system integrated. As a result, this system may be interrupted occasionally by other applications such as timely events or email alerts, when those predefined tasks represented partly by a sequence of keystrokes are being executed. By turning off some timely events or alerts temporary, we can minimize the interruptions. Currently, we are investigating other methods to represent execution instructions or to eliminate any interruption caused by timely events or alerts without temporary turning them off.

## 10. Acknowledgment

The authors would like to thank M. Guimond, C. Berryman, C. Myers, L. Banks, M. Howze and M. McMillin for useful discussions. The authors would also like to acknowledge the review of the manuscripts, suggestions for improvement and support from D. Dias and N. Halim.

## 11. References

- [1] W.M. McGovern and L. Lawrence, *Contracts and Sales: Cases and Problems*, Matthew Bender, 1986.
- [2] International Association of Contract and Commercial Managers, <http://www.iaccm.com>
- [3] M-C. Pong, EC-SignGate: electronic contract signing gateway, in *Proc. of 25<sup>th</sup> Annual Int'l Computer Software and Applications Conf.* pages 245-248, 2001.
- [4] C.G. Kang, New Digital Multisignature Scheme in Electronic Contract System, in *Proc. of IEEE Int'l Symposium on Information Theory*, page 486, 1995.
- [5] S. Fischmeister, G. Hagleitner and W. Pree, Hermes - A Learn M-Commerce Software Platform Utilizing Electronic Signatures, in *Proc of 35th Hawaii Int'l Conf. on System Science*, pages 4034-4041, 2002.
- [6] T. Kwok and T. Nguyen, A Secure Electronic Contract Management and Process System Automated with Predefined Tasks, in *Proc. of IEEE Int'l Conf. on e-Technology, e-Commerce and e-Service*, pages 276-281, 2005.