

A METHOD TO TRANSMIT AND UPDATE A TRANSMITTED ELECTRONIC DOCUMENT

5 BACKGROUND OF THE INVENTION

Today's commercial environment is moving towards a paperless system. In this system, commercial documents or data are usually electronically stored on a local database environment. However, it is increasingly important to provide a backup system at a remote location. This is advantageous so that in the event of a disaster where the locally stored data may become unreadable or un-retrievable, (for example due to a fire, a flood, or a computer virus attack on the local database, etc) electronic information will not be lost, and an organization can proceed to recover its data from the remote location.

Traditionally, a method of storing data at a remote location and keeping it readily accessible, is to transmit the data through a network and to have it stored in a computer's hard disk at a remote site.

It is often critical for businesses to maintain the confidentiality and integrity of their electronic data. Therefore, the data needs to remain encrypted both during transmission and when it resides at the remote site, as the transmission network, for example the internet, wide area network (WAN), local area network (LAN) etc, and even the remote site (for example a 3rd party hosting centre) itself, may not be itself secure.

The problem faced with the traditional method of storing data, is that when a transmitted document is amended or modified, the entire document will have to be encrypted and re-transmitted to the remote site. This is regardless of the extent of amendment or modification made on the transmitted document. This is time consuming. As some businesses rely on timely data, this method of storing data is unacceptable.

In another method, existing tools are used to synchronize unencrypted documents over a network. In this method the unencrypted amendments are synchronized with the transmitted documents to update the amendments. As the existing tools can only

synchronize unencrypted documents, the transmitted document is first decrypted, before synchronization with the amended document. Once synchronization is complete, the updated documents are then encrypted again for security. In this method, it will be necessary to set up a secure network route or infrastructure prior to synchronization, as the amendments are not encrypted during transmission and therefore experiences a threat of security. Again, this is time consuming and resource intensive, especially for large electronic documents like e-mails. More importantly, there is a compromise to data security during transmission, as the documents are not encrypted prior and during transmission.

10

It is an object of the present invention to overcome or at least ameliorate one or more of the above problems in the prior art.

15

Discussion of any one of the prior art mentioned above is not to be taken as an admission of the state of common general knowledge of the skilled addressee.

SUMMARY OF THE INVENTION

20 According to the invention, there is provided a method to securely transmit a document to a remote storage location, including partitioning the document into a plurality of distinct data blocks, said data block having a pre-determined block length, categorizing each distinct data block into one of two categories – incomplete and new, said incomplete data block having a data length less than a pre-determined block length, and said new block
25 having a data length equal to the pre-determined block length, assigning a digital signature to each classified data block, preparing a signature table to tabulate the assigned digital signatures, storing the assigned digital signatures to the prepared signature table, preparing a delta file to receive the data of the data blocks and finally transmitting the data blocks to the remote storage location.

30

Preferably, the method further includes encrypting the data blocks prior to transmitting the data blocks to the remote storage location. This is to provide further security to the documents when transmitting through a network.

According to another aspect of the invention, there is provided a method to securely incorporate amendments to a transmitted document including partitioning the amended document into a plurality of distinct data blocks, said data block having a pre-determined block length, preparing a subsequent signature table, taking signatures assigned and
5 store into the subsequent signature table, identifying the data blocks that have been amended by identifying identical signatures in the subsequent signature table with the signature table of the original document, identifying complete blocks of data in the amended document, classifying the identified complete blocks of data as new, computing and assigning signatures to the classified new data blocks, placing the assigned
10 signatures into the subsequent signature table, classifying the remaining data blocks as incomplete, preparing an amendment delta file, preparing a delta file to receive the data of the data blocks, transmitting only the amended data blocks to the remote storage location and integrating the transmitted amended data blocks to the transmitted documents thereby incorporating the amendments.

15

Preferably, the method further includes encrypting the data prior to transmission of the data blocks to the remote storage location.

Still preferably the method of classifying the data blocks includes classifying a data block
20 as processed where the data block has an assigned digital signature.

Preferably, the method of classifying the data blocks includes classifying a data block as new where the data block has no assigned digital signature, and has a data length equal to the pre-determined data length.

25

Preferably, the method of classifying the data blocks includes classifying a data block as incomplete where the data block has no assigned digital signature, and has a data length less than the pre-determined data length.

30

DESCRIPTION OF FIGURES

In order that the invention might be more fully understood, embodiments of the invention will be described by way of example only, with reference to the accompanying drawings,
5 in which:

Figure 1 shows the setup for performing the method of transmitting a document;

Figure 2 shows a flowchart illustrating a method of transmitting a document securely to a
10 remote storage location for the first time;

Figures 3A, 3B and 4 show the partitioning of the data blocks, the assigning of signatures and the transmission of data of the document;

15 Figure 5 shows a flowchart illustrating a method of incorporating amendments to a transmitted document at a remote storage location;

Figure 6 shows how the data blocks are identified and classified;

20 Figures 7A, 7B show the respective signature tables of the original document and the amended document;

Figure 8 shows the format of the amendment delta file 700 to be transmitted by the user;

25 Figure 9 shows a flowchart illustrating the method of generating a delta file from an amended document;

Figure 10 shows the construction of a new encoded file from the existing encoded; and

30 Figure 11 is a flowchart illustrating a method of constructing a new encoded file from the delta file.

The figures are not necessarily drawn to scale.

DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. The preferred
5 embodiments of the invention are not intended to limit the invention in its broadest aspect to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims. Furthermore, in the following detailed
description of the embodiments, numerous specific details are set forth in order to provide
10 an understanding of the present embodiments.

Referring to the drawings, Figure 1 shows an information network 105 suitable for implementing the methods according to the present invention. The information network 105 includes a user computer 101 connected to a remote storage location shown in
15 Figure 1 as a server computer 103 through a network 102. Documents prepared at a user computer 101 will be subsequently transmitted through the network 102 to the server computer 103. The user computer 101 may be any programmable device, for example, a general-purpose computer, a programmable hand phone, a personal digital assistant (PDA) or any other programmable digital appliance. The server computer 103 as shown
20 in Figure 1, is a general-purpose computer. The network 102 may be a Local Area Network, Wide Area Network, Intranet, Internet, etc.

According to the method of the invention, the document is first partitioned and categorized before encrypting and transmitting to the server 103.

25 Figure 2 depicts a flow chart illustrating a method according to the present invention for processing a document for its secure transmission to the server 103. A pre-determined block size of L bytes is first determined. Once determined, this block size L is maintained throughout the method of the invention. The pre-determined block size L may be
30 determined based on the electronic size of each individual document which is to be transmitted, so that, for example, a larger sized document may have longer data blocks, and vice versa. In another variation, the pre-determined block size L may be determined based on the type of document, for example, information intensive documents like spreadsheets, graphics or documents with text. Once the block size L is determined, the

document is partitioned into equal blocks of L bytes. The programme then proceeds to classify these data blocks as “new” or “incomplete”. A data block is classified as “new” if its partitioned block size is equal to the pre-determined block size L, and classified as “incomplete” if its partitioned block size is less than the pre-determined block size L. As it is the first time these data blocks are processed, the programme classifies these data blocks as “new”. In the situation where the size of the document is not an exact multiple of L bytes, the last block will be shorter than the pre-determined block length L and hence classified as an “incomplete” block.

A signature table 210 is then prepared so that digital signatures computed and assigned, may be tabulated.

A digital signature 20 is computed and assigned for each data block 10 classified as “new”, using, for example, the algorithm such as a 128-bit MD5 (message digest). The digital signature 20 is computed for all “new” blocks and these signatures are stored in the prepared signature table 210. There is no digital signature computed and assigned for an “incomplete” block. Next, the data contained in all the data blocks, i.e., blocks classified as either “new” or “incomplete” are encoded using any method, and appended to a delta file and finally, the delta file is transmitted to the server computer.

The method as described in Figure 2 will be illustrated in the following figures 3A, 3B and 4.

Applying the method as described above, Figure 3A shows a document 200 that has been partitioned into data blocks arbitrarily termed A1 through to A6 using any known conventional data partitioning methods. The document is partitioned into the pre-determined block size L, for example, 8192 bytes.

It is quite likely that the document size is not an exact multiple of the pre-determined block size. In this example, the final data block A6 may fall short of the pre-determined block size. In this situation, data block A6 is classified as an “incomplete” block. All the other complete data blocks 10, arbitrarily termed as A1 through to A5 is of the pre-determined block size, L. As the document 200 has not been previously processed, data blocks A1 through A5 are classified as “new”.

Next, following the method described in Figure 2, a digital signature 20, for example of 128 bit, is computed for each classified “new” block. Therefore, digital signatures 20 are computed and assigned to data blocks A1 through to A5.

5

The digital signature 20 is computed and assigned such that they are unique. This may be achieved by adopting several commonly known algorithms used in cryptography. An example of one such algorithm is the 128-bit MD5 message digest. The digital signatures 20 and their assigned corresponding data blocks 10 are stored in a signature table 210, as seen in Figure 3B. This signature table 210 functions as a digital fingerprint or as an electronic identity of the document 200. Digital signatures 20 are computed and assigned only for data blocks categorized as “new”, and therefore a digital signature is not computed for block A6 categorized as an “incomplete” block as seen in Figure 3B. Next, as seen in Figure 4, the data contained in the “new” and “incomplete” blocks are encoded and placed in a delta file 301 to be transmitted to the server 103 through the network 102. The encoding of data is optional. To enhance security of transmission, encoding may be performed by encrypting the contents with any of the encryption algorithms found commonly in the market.

20 The form of encoding will depend on several factors. For instance, if security of the document 200 is of concern, then the encoding takes the form of an encryption. If the document is not previously in a compressed format, then encoding takes the form of data compression. Any other form of encoding can be applied provided that any chosen conventional algorithm used to encode the data, will not result in a loss in data integrity when the document is subsequently decoded. The next step is to transmit the delta file 301 to the server computer 103 where the encoded data will be extracted and stored.

30 If network bandwidth is of concern, encoding may involve compressing the document with any commonly known compression algorithm. Encoding may also involve first compressing followed by encrypting of each data block content. It is to be understood that the present invention is not limited to the type or choice of algorithm for compression and/or encryption, and the invention may be performed with the use of any known encoding method, with a requirement that documents or data encoded may similarly be decoded with no loss in data integrity.

It is further to be understood that the terms adopted in this specification – “incomplete”, “new” and “processed” are purely arbitrary, and have been termed as such, as an example. Other terms may also be adopted, for example, the use of the term “partial” instead of “incomplete”, “unprocessed” instead of “new”, and “executed” instead of “processed”, without departing from the spirit of the invention.

Figure 4 shows the layout of the delta file 301 prepared for the document as described in the method of Figure 2.

10

The delta file 301 is a file prepared that includes instructions to the system, to prepare the data blocks prior to transmission to the remote location, or server 103

This delta file 301 is prepared and contains instructions to insert all the data contents of the document 200 carried by the delta file 301, and transmitted by a user to the server 103. The instructions contained in the delta file 301 includes instructions necessary to enable the server 103 to reconstruct the encoded document 202 correspondingly to form data blocks B1 to B6, arbitrarily named , representing the encoded, transmitted form of data blocks A1 to A6. The data blocks B1 to B6 for the encoded document 202 forms the encoded data file 202 at the server constructed from the contents of the document 200. Therefore, the data blocks A1 to A6 of the document 200 is encoded and transmitted to form the encoded data file formed by corresponding data blocks B1 to B6, arbitrarily named. This is done sequentially from the first block A1 to the last block A6.

Figure 5 shows a method where amendments made to the original document 200 are identified and prepared prior to transmission to the server 103. A signature table (seen in Figure 7B) is prepared for the amended document 205, known as an amendment signature table 602.

Data blocks of the amended document 205 with computed signatures found in the signature table 210 of the original document 200 are first identified (“processed” data blocks), and their assigned signatures are stored in the subsequent signature table 602. These data blocks do not need to be prepared for transmission, as they have previously been transmitted. This saves time, and is an advantage of the present invention.

Next, complete data blocks of block size L are identified in the amended document 205. These data blocks are classified as “new”. Again, digital signatures are computed for data blocks classified as “new”. A digital signature is computed and assigned to each “new” data block, and these signatures are stored in the subsequent signature table 602. The remaining data blocks are classified as “incomplete”.

Figures 3, 6, 7A and 7B applies the method as described in Figure 5.

The original document 200 is partitioned into blocks A1 through to A6 and the signatures assigned to each block are taken and stored into a signature table 210, as shown in Figure 3, in accordance with the steps outlined in the flow chart depicted Figure 2.

The programme first identifies which blocks in the amended document 205 relates to the original document 200. As certain data blocks of the original document 200 may be common with the amended document 205, its digital signature can be found in the signature table 210. In the present example shown in Figures 3, 6 and 7, it can be seen that there are 3 blocks common to both the original document 200 and amended document 205 as they have identical digital signatures, i.e. blocks A3 and D2 have an identical digital signature S(A3), blocks A2 and D5 have an identical digital signature S(A2), and, blocks A4 and D6 have identical digital signature S(A4). Data blocks D2, D5 and D6 of the amended document 205 are classified as “processed” blocks. It is classified thus as they have been previously processed and their encoded data can be found in the server computer 103 therefore making it unnecessary to transmit this information again.

The subsequent signature table 602 of the amended document 205 is updated with the digital signatures of the “processed” blocks, i.e. S(A3) for block D2, S(A2) for block D5, and S(A4) for block D6. Having identified the blocks common to both the original 200 and amended documents 205, the next step is to identify complete blocks of data in the amended document 205, i.e. blocks with previously determined data size of L bytes. Block D3 508 is one such block, and its computed and assigned digital signature S(D3) is computed and placed in the subsequent signature table 602. This data block is classified as a “new” block as its signature is not found in the signature table 210 and its contents is recognised to have not previously been transmitted to the server 103 through the

amendment delta file 700. The remaining blocks D1, D4 and D7 are classified as “incomplete” blocks, as their data sizes are less than the pre-determined block size of L bytes. These “incomplete” blocks have not been previously assigned digital signatures and therefore, these signatures are not present in the signature table 602. Nevertheless,
 5 the content of the “incomplete” blocks represents new information that must be transmitted to the server 103 via the amendment delta file 700.

Figure 8 shows the format of the amendment delta file 700 to be transmitted by the user to the server 103 to update the existing encoded document at the server 103 to reflect the
 10 amendments made to the document 200 at the client computer 101. The amendment delta file 700 comprises a sequence of instructions for the server computer 103 to execute. There are basically two instructions – either (i) to insert data carried by the delta amendment file 700 or (ii) to copy data that is already on record in the server computer 103 in order to update the encoded document at the server end.

15 The amendment delta file 700 is constructed after having identified the types of blocks present in the amended document 205. D1 is an “incomplete” block with no assigned digital signature. Its data is encoded and appended to the delta file 701 with instruction (i) for the server computer 103 to insert this data into the updated encoded document file.
 20 D2 is a “processed” block and its data is common to A3 (block 3) of the original document 200. Hence, instruction (ii) is called, and its entry in the amendment delta file 702 is an instruction for the server computer 103 to use block 3 of the current encoded document to update the new encoded document. D3 is a “new” block, and therefore instruction (i) is called. Its data is encoded and appended to the amendment delta file 703 with the
 25 instruction for the server computer to insert this data into the updated encoded document file. D4 and D7 are “incomplete” blocks and are processed in the same fashion as block D1, calling instruction (i). D5 and D6 are “processed” blocks and are processed in the same fashion as block D2, i.e. calling instruction (ii).

30 Figure 9 is a flowchart illustrating the method of generating a delta file from an amended document 205. The data blocks of the amended document are first classified or identified as “processed”, “new” and “incomplete” blocks of the amended document. A new delta file and a signature table is first created for the amended document. The data blocks classified as “processed” blocks are the first blocks to be identified, as their signatures will

match those in the signature table belonging to the original document 200. From the remaining data, the blocks are partitioned according to the basic pre-determined block length of L bytes of data. These data blocks are classified as “new” blocks, as they have not previously been processed. The remaining blocks are classified as “incomplete” as
 5 their data sizes are smaller than the basic predetermined block size of L bytes.

After identifying the 3 types of blocks, each block is considered in turn. If the block is classified as a “processed” block, then the newly created signature table is updated with the data block’s digital signature (which can be found in the old signature table). The delta
 10 file is appended with the instruction to copy data from the block number found in the old signature table. There is no need to insert the block’s data into the delta file for transmission as its data is already on record in the server computer 103. If the data block is classified as a “new” block, then the block’s signature is computed and tabulated into the new signature table. The block’s data is then encoded and the data appended to the
 15 delta file for transmission to the server computer where it will be stored. Note that if this block is found in subsequent modifications to the document, then it will be classified as a “processed” block and its data need not be re-transmitted again to the server computer. If the block under consideration is neither a “processed” or “new” block, then it must be an “incomplete” block. The block’s data is then encoded and the data appended to the delta
 20 file for transmission to the server computer where it will be stored. Note that the signature of an “incomplete” block is neither computed nor recorded and as such can never be identified in subsequent modifications to the document. Having processed the final block of the amended document 205, the current signature file is then discarded and replaced by the new signature file. The construction of the delta file is now complete and can be
 25 transmitted to the server computer for processing.

Figure 10 shows the construction of the new encoded file 902 from the existing encoded file 903 found in the server and the delta file 901 received from the client. The delta file, reproduced from step 700 comprises a set of instructions. New data (comprising contents
 30 of “new” and “incomplete” blocks) is inserted into the new data file as illustrated in steps 904, 906, 907 and 910. Data that has been previously processed (contents of “processed” blocks”) are copied from the existing data file to the new data file as illustrated in steps 905, 908 and 909.

Figure 11 is a flowchart illustrating a method according to the present invention of constructing a new encoded file from the delta file and existing encoded file by the server computer 103. Each block in the delta file is processed sequentially starting from the first data block to the last data block. If the block under consideration contains a “copy”
5 instruction (ii), then, this is a “processed” block and data is extracted from the existing encoded data file and appended to the new encoded data file 1005. If the block under consideration contains an “insert” instruction (i), then, this is either a “new” or “incomplete” block and its data is appended to the new encoded data file 1006. After the last block of the delta file has been processed, the newly encoded data file replaces the existing
10 encoded data file 1008.

CLAIMS

1. A method to securely transmit a document to a remote storage location, including partitioning the document into a plurality of distinct data blocks, said data
5 block having a pre-determined block length;
categorizing each distinct data block into one of two categories – incomplete and new, said incomplete data block having a data length less than a pre-determined block length, and said new block having a data length equal to the pre-determined block length;
10 assigning a digital signature to each classified data block;
preparing a signature table to tabulate the assigned digital signatures;
storing the assigned digital signatures to the prepared signature table;
preparing a delta file to receive the data of the data blocks; and
transmitting the data blocks to the remote storage location.
15
2. A method to securely transmit a document to a remote storage location according to claim 1 wherein the method further includes encrypting the data blocks prior to transmitting the data blocks to the remote storage location.
- 20 3. A method to securely incorporate amendments to a transmitted document, said document transmitted in a method according to claim 1 including
partitioning the amended document into a plurality of distinct data blocks, said data block having a pre-determined block length;
preparing a subsequent signature table;
25 taking signatures assigned and store into the subsequent signature table;
identifying the data blocks that have been amended by identifying identical signatures in the subsequent signature table with the signature table of the original document;
identifying complete blocks of data in the amended document;
30 classifying the identified complete blocks of data as new;
computing and assigning signatures to the classified new data blocks;
placing the assigned signatures into the subsequent signature table;
classifying the remaining data blocks as incomplete;
preparing an amendment delta file;

preparing a delta file to receive the data of the data blocks;
transmitting only the amended data blocks to the remote storage location;
and
integrating the transmitted amended data blocks to the transmitted
5 documents thereby incorporating the amendments.

4. A method to securely incorporate amendments to a transmitted document
according to claim 3, wherein the method further includes encrypting the data prior to
transmission of the data blocks to the remote storage location.

10

5. A method to securely incorporate amendments to a transmitted document
according to any one of claims 3 or 4 wherein the method of classifying the data blocks
includes classifying a data block as processed where the data block has an assigned
digital signature.

15

6. A method to securely incorporate amendments to a transmitted document
according to any one of claims 3 to 5 wherein the method of classifying the data blocks
includes classifying a data block as new where the data block has no assigned digital
signature, and has a data length equal to the pre-determined data length.

20

7. A method to securely incorporate amendments to a transmitted document
according to claim 3 wherein the method of classifying the data blocks includes
classifying a data block as incomplete where the data block has no assigned digital
signature, and has a data length less than the pre-determined data length.

25