

METHOD AND SYSTEM FOR EFFICIENT DISTRIBUTION OF E-MAIL ATTACHMENT

Field of invention

5 This invention relates to a method and a system of sending file via e-mail. More particularly, this invention relates to a method and a system of allowing a sender to send one or more files via an e-mail to multiple recipients. Still more particularly, this invention relates to a method and a system for managing and distributing file among recipients and storage systems.

10

Background

 Using e-mail to send file attachment is a convenient way for sharing documents. However, most e-mail systems impose file attachment size limits in order to prevent overloading the server. In a known system, a cloud-based e-mail
15 attachment service is implemented to relieve the e-mail server load and to eliminate the need for a size limitation by shifting the storage of e-mail attachments to cloud storage. However, a bottleneck on the server occurs when multiple recipients download the same copies of attachments from the cloud simultaneously. This is a common problem since people in the office read their e-mails in the office about the
20 same time, causing heavy network traffic and thus slowing down the downloading process of retrieving e-mails with large attachment. This may occur, for example, when people arrives in the office in the morning and read their e-mails with the attachment, and starts to download the attachment for viewing around the same time.

25 Thus, those skilled in the art are constantly striving to provide an improved method and system for sending and receiving e-mails enclosing file attachments.

Summary of Invention

 The above and other problems are solved and an advance in the art is
30 provided by a method and a system of sending file via e-mail in accordance with this invention. A first advantage of a system and a method in accordance with

embodiments of this invention is the efficient downloading of the file from various sources. A second advantage of a system and a method in accordance with embodiments of this invention is that there is no limit on the file size of the file that may be sent via an e-mail. A third advantage of a system and a method in
5 accordance with embodiments of this invention is that it allows a sender to send the file which has a large file size to multiple recipients via an e-mail without overloading the server.

In accordance with embodiments of this invention, a system and a method for
10 sending and receiving file via e-mail among a number of e-mail clients via a server and a plurality of slice stores is provided in the following manner. A sender using an e-mail client creates an e-mail. The file is sliced using a secret matrix to a number of slices and transmitted to various slice stores. A proxy attachment containing information of said sliced file is generated. The file in the e-mail is replaced by the
15 proxy attachment and subsequently transmitted to the server. The e-mail is then transmitted to a number of recipients. Preferably, the file size of the proxy attachment is smaller than the file size of the file.

In accordance with embodiments of this invention, the attachment information
20 is generated to include an attachment ID, the secret matrix, e-mail address of the recipients, and a session key.

In accordance with embodiments of this invention, in response to receiving the attachment information from the sender, the server creates a new file entry in a
25 persistent storage in the server and stores the attachment information in the new file entry.

In accordance with embodiments of this invention, prior to slicing the file, the e-mail client may determine a file size of the file is above a predetermined file size.

In accordance with embodiment of this invention, the proxy attachment comprises an file attachment ID, a ; IDs of each of the slices, $S = \{S_1, S_2, \dots, S_n\}$ where S is the set of slices and n is the number of slices the document has been sliced into; a minimum number of slices, m , needed to re-combine the document
5 where $m < n$; a checksum for each of the slices; $R = \{r_1, r_2, r_3, \dots, r_i\}$, wherein R refers to the set of e-mail email addresses $r_1, r_2, r_3, \dots, r_i$ are the e-mail address of each of said recipient; a session key for authentication; and server ID, α , containing an IP address of said server.

10 In accordance with embodiment of this invention, in response to receiving the e-mail by one the e-mail clients used by one of the recipients, the e-mail client transmits a request to the server for retrieving the file, wherein the request includes the attachment ID, a , and a list of slice IDs, $S = \{S_1, S_2, \dots, S_n\}$.

15 In accordance with embodiment of this invention, in response to receiving the request for retrieving the file, the server retrieves the attachment information according to the request and generates a list of download information based on the retrieved attachment information. The server then transmits the list of download information to the e-mail client. Preferably, the list of download information
20 comprises, $D(a, s_i) = \{I_{i,r}\}$, where a is the attachment ID, s_i is the slice ID of i^{th} slice, and $I_{i,r}$ is the IP address of a recipient or server having i^{th} slice.

In accordance with embodiment of this invention, in response to receiving the list of download information from the server, one of the e-mail clients retrieves the
25 slices based on the list of download information in the following manner. The list of download information is scanned. Any download information that has already been retrieved will be removed from the list of download information. The e-mail client then selects a subset of download information and subsequently determines if each of the download information in the selected subset of download information has more than
30 one IP address. In response to determining that each of the download information remaining in the list of download information has more than one IP address, the IP

addresses are re-ordered. The IP addresses are re-ordered in the following manner. The download cost for each of the IP addresses in each of the download information is determined. The IP addresses in each of the download information are then arranged according to lowest download cost to highest download cost. The
5 determining of the download cost may be determined by pinging each of the IP addresses to determine a download speed of each of the IP addresses. Preferably, the pinging is a simple Internet Control Message Protocol (ICMP) ping.

In accordance with embodiment of this invention, the e-mail client performs in
10 the following manner when retrieving each of the slices of the selected subset of download information. The e-mail client monitors for a slice retrieved. The e-mail client then determines a validity of the slice being retrieved.

In accordance with embodiment of this invention, the e-mail client performs in
15 the following manner when determining the validity of the slice retrieved. A checksum for the slice retrieved is calculated. The checksum is verified to determine if valid based on a checksum in the proxy attachment. If valid, the e-mail client updates the server that the retrieved slice is available at the recipient. If not valid, the retrieved sliced is discarded.

20

In accordance with an embodiment of this invention, the e-mail client performs the following after retrieving a slice. The e-mail client determines if all slices of the subset of download information are retrieved. If all slices of the subset of the download information are retrieved, the e-mail client determines a total number of
25 slices retrieved and compare the total number with a minimum number of slices, m , in the proxy attachment. The e-mail client generates a request for the secret matrix in response to the total number of slices being equal to or more than the minimum number of slice in the proxy attachment. The e-mail client then transmits the request to the server. In response the request, the server retrieves the secret matrix
30 associated to the request and transmits the secret matrix to the e-mail client. In

response to receiving the secret matrix from the server, the e-mail client re-combines the slices based on the secret matrix to form the file.

Brief description of drawings

5 The above and other features and advantages of a method and a system in accordance with this invention are described in the following detailed description and are shown in the following drawings:

 Figure 1 illustrating devices and systems communicatively connected to a network to provide or perform a system and/or method in accordance with an
10 embodiment of this invention;

 Figure 2 illustrating a representative processing system in one or more of the devices shown in figure 1 in accordance with embodiments of this invention;

 Figure 3a illustrating a sequence diagram in respect to a first recipient sending file via an e-mail to a number of recipients in accordance with an
15 embodiment of this invention;

 Figure 3b illustrating a sequence diagram in respect to a second recipient retrieving the file in accordance with an embodiment of this invention;

 Figure 3c illustrating a sequence diagram in respect to a third recipient retrieving the file in accordance with an embodiment of this invention;

20 Figure 4 illustrating a flow diagram for sending file via an e-mail to a number of recipients in accordance with an embodiment of this invention;

 Figures 5 and 6 illustrating a flow diagram for retrieving the file in accordance with an embodiment this invention;

 Figure 7 illustrating a flow diagram of a process performed by a server in
25 accordance with an embodiment of this invention;

 Figure 8 illustrating a flow diagram of a process performed by the server to create a new file entry in accordance with an embodiment of this invention;

 Figure 9 illustrating a flow diagram of a process performed by the server to update an attachment information in accordance with an embodiment of this
30 invention; and

Figure 10 illustrating a flow diagram of a process performed by the server to transmit a secret matrix in accordance with an embodiment of this invention.

Detailed description

5 This invention relates to a method and a system of sending file via e-mail. More particularly, this invention relates to a method and a system of allowing a sender to send one or more files via an e-mail to multiple recipients. Still more particularly, this invention relates to a method and a system for managing and distributing file among recipients and storage systems

10

 It is envisioned that a system and/or method in accordance with embodiments of this invention may be used to allow a user to send file via e-mail to multiple recipients. In current system, a sender may send an e-mail enclosing one or more files to multiple recipients. A server upon receiving the e-mail with file transmits a
15 copy of the e-mail with the file to each of the intended recipients. This is inefficient as the server is required to transmit the e-mail with file to multiple recipients.

 In accordance with embodiments of this invention, instead of sending the file to all the recipients, the file is sliced and stored in one or more slice stores. For
20 purposes of this discussion, a file is any text documents; image files; audio files; or any other documents or files that can be attached to an e-mail. The file is then replaced by a proxy attachment which contains the necessary information to retrieve a required number of slices to re-combine the document. A recipient receiving the e-mail actively discovers other recipients who have possession of any of the slices and
25 retrieves the slices from the other recipients instead of downloading the slices from slice stores. This helps to lessen the load from the server as the file is not sent to recipients directly and also allows the recipients to download the sliced file faster and more efficiently using a local area network which is typically faster than downloading the sliced file from a wide area network. More details of the invention will be
30 described below.

One skilled in the art will recognise that more than one file can be sent via an e-mail without departing from the invention. Further, there is no restriction on the file size for each of the files. This means that files with small file size can also be sent without departing from the invention.

5

Figure 1 illustrates a system 100 that performs processes for providing the systems and methods in accordance with embodiments of this invention. The system 100 includes an attachment server 110, e-mail clients 121-124, and slice stores 131-133.

10

Attachment server 110 is a processing system that connects to e-mail clients 121-124 via a conventional wire connection, wireless connection or any other method. Although not shown, it is understood that e-mail clients 121-124 may also be connected to attachment server 110 indirectly via communications over wide area network 140 without departing from this invention. Further, attachment server 110 is connected to slice stores 131-133 via a conventional wire connection, wireless connection or any other method through wide area network 140. Attachment server 110 executes applications for managing the storage of file received from one of e-mail clients 121-124. Attachment server 110 manages the location of the file that is stored in e-mail clients 121-124 and slice stores 131-133 in accordance with embodiments of this invention. One skilled in the art will recognize that although only one attachment server 110 is shown, any number of servers may be connected without departing from this invention.

25

E-mail clients 121-124 are typical software such as Microsoft Outlook, Outlook Express, Eudora, Gmail, Hotmail, Yahoo, etc that are executed on processing systems such as a desktop computer, laptop computer, or other computer terminals that connect to attachment server 110 via a conventional wire connection, wireless connection or any other method. E-mail clients 121-124 are also connected to an e-mail server that manages the transmissions of e-mails among e-mail clients 121-124. Processing system of e-mail clients 121-124 are also interconnected to each other

30

via a conventional wire connection, wireless connection or any other method. Each e-mail client 121-124 executes applications that perform the processes in accordance with this invention. One skilled in the art will recognize that although only four e-mail clients 121-124 are shown, any number of e-mail clients may be connected without
5 departing from this invention.

Slice stores 131-133 are processing systems that provide resources to server 110 and e-mail clients 121-124 and are connected to a wide area network 140 via a conventional wire connection, wireless connection or any other method. Resources
10 may include but are not limited to storage, processing time, and applications. One skilled in the art will recognize that although only three slice stores 131-133 are shown, any number of slice stores may be connected without departing from this invention.

Figure 2 illustrates exemplary processing system 200 that represents the processing systems in attachment server 110, e-mail clients 121-124 and slice stores 131-133 that execute instructions to perform the processes described below in accordance with this invention. One skilled in the art will recognize that the instructions may be stored and/or performed as hardware, firmware, or software
15 without departing from this invention. One skilled in the art will recognize that the exact configuration of each processing system may be different and the exact configuration executing processes in accordance with this invention may vary and processing system 200 shown in figure 2 is provided by way of example only.

Processing system 200 includes Central Processing Unit (CPU) 205. CPU 205 is a processor, microprocessor, or any combination of processors and microprocessors that execute instructions to perform the processes in accordance with the present invention. CPU 205 connects to memory bus 210 and Input/Output (I/O) bus 215. Memory bus 210 connects CPU 205 to memories 220 and 225 to
25 transmit data and instructions between the memories and CPU 205. I/O bus 215 connects CPU 205 to peripheral devices to transmit and receive data between CPU
30

205 and the peripheral devices. One skilled in the art will recognize that I/O bus 215 and memory bus 210 may be combined into one bus or subdivided into many other busses and the exact configuration is left to those skilled in the art.

5 A non-volatile memory 220, such as a Read Only Memory (ROM), is connected to memory bus 210. Non-volatile memory 220 stores instructions and data needed to operate various sub-systems of processing system 200 and to boot the system at start-up. One skilled in the art will recognize that any number of types of memory may be used to perform this function.

10

 A volatile memory 225, such as Random Access Memory (RAM), is also connected to memory bus 210. Volatile memory 225 stores the instructions and data needed by CPU 205 to perform software instructions for processes such as the processes for providing a system in accordance with this invention. One skilled in the art will recognize that any number of types of memory may be used to provide volatile memory and the exact type used is left as a design choice to those skilled in the art.

 I/O device 230, keyboard 235, display 240, memory 245, network device 250 and any number of other peripheral devices connect to I/O bus 215 to exchange data with CPU 205 for use in applications being executed by CPU 205. I/O device 230 is any device that transmits and/or receives data from CPU 205. Keyboard 235 is a specific type of I/O that receives user input and transmits the input to CPU 205. Display 240 receives display data from CPU 205 and display images on a screen for a user to see. Memory 245 is a device that transmits and receives data to and from CPU 205 for storing data to a media. Network device 250 connects CPU 205 to a network for transmission of data to and from other processing systems. Other devices such as a persistent storage may also be provided in processing system 200. A persistent storage is a storage device that preserves data when its power is off.

30 A system and method for sending file via e-mail in accordance with an embodiment of the invention is described. Figures 3a, 3b and 3c illustrate sequence

diagrams for transmissions between e-mail clients 121-124, attachment server 110 and slice stores 131-133 in accordance with an embodiment of the invention. Each of the remainder of the figures illustrates flow diagrams of processes performed in the components to send or retrieve file being sent via an e-mail to multiple recipients.

5

Figure 3a illustrates a sequence diagram of sequence 300a in which a sender 320 sends a file via an e-mail to multiple recipients and a first recipient 310 receives the e-mail in accordance with an embodiment of this invention. For purpose of this discussion, sender 320 sends e-mail using any of e-mail clients 121-124 and recipients receives the e-mail using any of e-mail clients 121-124. For brevity, when referring to sender 320 or recipients, it is taken to mean that the e-mail clients that the sender 320 or recipients used are the devices performing the processes.

Sender 320 creates an e-mail with a file attached to the e-mail. Prior to sending the e-mail to the recipients, the system will detect if any file is attached to the e-mail. If a file is attached in the e-mail, the system will slice the file to smaller pieces with a matrix and store the sliced file in slice stores 131-133 in 3105.

In 3110, sender 320 sends attachment information to attachment server 110. At the same time, the file is replaced by a proxy attachment and the e-mail is sent to the recipients via an e-mail server. In response to receiving the attachment information from the sender, attachment server 110 stores the attachment information in a register of attachment information in its persistent storage. The attachment information can also be stored in memory for faster access. Further details of a sender sending file via an e-mail will be described below with reference to figure 4. Further details of attachment server 110 receiving the attachment information will be described below with reference to figure 7.

One skilled in the art will recognise that slicing of the document may be performed by the attachment server 110 instead of the sender 320 without departing from the invention.

In 3115, first recipient 310 receives the e-mail containing the proxy attachment and transmits a request to the attachment server 110 for a download information. In response to the request, the attachment server 110 performs a lookup
5 in the register for the attachment information based on the request. The attachment server 110 then generates a download information based on the retrieved attachment information and transmits the download information to the first recipient 310 in 3120. The download information is stored in memory for future requests from clients. Since the first recipient 310 is the first person who receives or opens the e-mail at this
10 juncture, the download information will contain information that the slices are only available at various slice stores 131-133. With the download information, the first recipient transmits a request for a portion of the slices to the slice stores 131-133 in 3125. The requested portion of slices is then retrieved in 3130 from the slice stores 131-133. After the requested portion of slices is retrieved in 3130, first recipient 310
15 updates the attachment servers 110 on the slices it has obtained and repeats from 3115 to request for download information until a required number of slices are retrieved. This allows first recipient 310 to determine if other portions of the slices are available from the other recipients. If other portions of the slices are not available from other recipients, the first recipient 310 continues to request for another portion of
20 the slices from the slice stores 131-133. If other portions of the slices are available from other recipients, the first recipient 310 requests for other portions of the slices from one of the recipients.

After the required number of slices are retrieved, the first recipient 310
25 transmits a request for a matrix from the attachment server 110 in 3135. In response to the request, the attachment server 110 transmits the matrix to the first recipient 310 in 3140. The attachment server 110 further updates the attachment information to indicate the slices available at the first recipient 310. With the matrix, the first recipient 310 re-combines the sliced file to generate the original file. Further details
30 of a first recipient receiving the document will be described below with reference to

figures 5 and 6. Further details of the attachment server 110 receiving the request will be described below with reference to figure 7.

One skilled in the art will recognise that more than one document can be sent via an e-mail without departing from the invention. Further, there is no restriction on the file size for each of the files. This means that document with small file size can also be sent without departing from the invention.

Figure 3b illustrates a sequence diagram of sequence 300b in which a second recipient 350 receives the e-mail. When a second recipient 350 receives the e-mail, the second recipient 350 transmits a request for the download information in 3205. The attachment server 110 performs a lookup in the register for the attachment information based on the request. If a download information associated to the attachment information has been generated, the attachment server 110 will retrieve the download information associated to the attachment information. Otherwise, the attachment server 110 generates a download information based on the retrieved attachment information and transmits the download information to second recipient 350 in 3210. Assuming the first recipient 310 has obtained some or all the slices, the download information will contain information that the slices are available at the first recipient 310 and various slice stores. With the download information, the second recipient 350 attempts to retrieve a portion of the slices from the first recipient 310 by transmitting a request for the slices in 3215. In response to the request from the second recipient 350, the first recipient transmits the requested slices in 3220. After the requested portion of slices is retrieved in 3220, second recipient 350 updates the attachment server 110 on the slices it has obtained and repeats from 3205 to request for download information until a required number of slices are retrieved.

In the event the first recipient 310 is not online or the slices are no longer available, the second recipient 350 then transmits a request for a portion of the slices to the slice stores 131-133 in 3225. In response to the request, the slice store 131-133 transmits the requested portion of slices to the second recipient 350 in 3230.

After the requested portion of slices is retrieved in 3230, second recipient 350 updates the attachment servers 110 on the slices it has obtained and repeats from 3205 to request for download information until a required number of slices are retrieved.

5

After the required number of slices is obtained, the second recipient 3235 transmits a request for a matrix to re-combine the sliced file. In response to the second recipient 350, the attachment server 110 transmits the matrix and updates the attachment information to indicate the slices available at the second recipient 350.

10 With the matrix, the second recipient 350 combines the sliced file to generate the original file. Further details of the second recipient receiving file will be described below with reference to figures 5 and 6. Further details of the server receiving the request will be described below with reference to figures 7-10.

15 One skilled in the art will recognise that more than one file can be sent via an e-mail without departing from the invention. Further, there is no restriction on the file size for each of the files. This means that file with small file size can also be sent without departing from the invention.

20 Figure 3c illustrates a sequence diagram of sequence 300c in which a third recipient 360 receives the e-mail. When the third recipient 360 receives the e-mail, the third recipient 360 transmits a request for the download information in 3305. The attachment server 110 will look up the register for the attachment information based on the request. If a download information associated to the attachment information
25 has been generated, the attachment server 110 will retrieve the download information associated to the attachment information. Otherwise, the attachment server 110 generates a download information based on the retrieved attachment information and transmits the download information in 3310 to the third recipient 360. Assuming the first recipient 310 and the second recipient 350 have obtained some or all the slices,
30 the download information will contain information that the slices are available at the first recipient 310, second recipient 350 and various slice stores.

With the download information, the third recipient 360 will attempt to retrieve the slices from either the first recipient 310 or the second recipient 350. In order to speed up the process of retrieving all the slices, the third recipient 360 may request a portion of the slices from the first recipient 310 and another portion of the slices from the second recipient 350. Hence, the third recipient 360 transmits a request for a portion of the slices from the second recipient and another portion of the slices from the first recipient in 3315. In response to the request from the third recipient 360, the first recipient 310 and the second recipient 350 transmit the requested portions of slices in 3320. After the requested portions of slices are retrieved in 3330, third recipient 360 updates the attachment servers 110 on the slices it has obtained and repeats from 3305 to request for download information until a required number of slices is retrieved.

In the event the first recipient 310 is not online or the slices are no longer available with the first recipient 310, the third recipient 360 transmits a request for another portion of the slices from the second recipient 350 instead. Further, if both first and second recipients 310 and 350 are not online or the slices are no longer available with the first and second recipients 310 and 350, the third recipient 360 transmits a request for a portion of the slices from the slice stores 131-133 in 3335. In response to the request, the slice stores 131-133 transmit the requested portion of slices to the third recipient 360 in 3340. After the requested portion of slices is retrieved in 3340, third recipient 360 updates the attachment servers 110 on the slices it has obtained and repeats from 3305 to request for download information until a required number of slices is retrieved.

After the required number of slices is obtained, the third recipient 360 transmits a request for a matrix to re-combine the sliced file to the server 330 in 3345. In response to the request from the third recipient 360, the server 330 transmits the matrix and updates the attachment information to indicate the slices available at the third recipient 360 in 3350. With the matrix, the third recipient 360 re-combines the

sliced file to generate the original file. Further details of the third recipient receiving file will be described below with reference to figures 5 and 6. Further details of the attachment server 110 receiving the request will be described below with reference to figures 7-10.

5

One skilled in the art will recognise that more than one document can be sent via an e-mail without departing from the invention. Further, there is no restriction on the file size for each of the files. This means that document with small file size can also be sent without departing from the invention.

10

Figure 4 illustrates a flow diagram of process 400 performed by a system in one of e-mail clients 121-124 used by a sender 320 attempting to send an e-mail attaching at least one file to multiple recipients in accordance with an embodiment of this invention. The system is provided by processes stored in software, firmware, or hardware. Process 400 begins in step 405 by slicing the file with a secret matrix into a number of slices. The slices are then transmitted to the slice stores 131-133 in step 410.

One skilled in the art will recognise that prior to slicing the file with the secret matrix, process 400 may verify if the file is of certain file size. If the file is below the certain file size, no slicing of the file is required. If the file is above the certain file size, slicing of the file will commence.

Process 400 then generates a "proxy attachment" file for the sliced file in step 415. The proxy attachment is typically a small text file that contains information regarding the attachment. Further the file size of the proxy attachment is smaller than the file size of the file. The proxy attachment may optionally be encrypted for added security. The proxy attachment comprises:

1. Attachment ID, a ;

2. The IDs of the all the slices that make up the file $S = \{S_1, S_2, \dots, S_n\}$ where S is the set of slices of the sliced file and n is the number of slices the file has been sliced into;
3. The minimum number of slices needed to re-combine the attachment, m where $m < n$;
4. Checksum of each slice;
5. The recipients' email addresses, $R = \{r_1, r_2, r_3, \dots, r_i\}$, wherein R is the set of recipient e-mail addresses; and $r_1, r_2, r_3, \dots, r_i$ are the e-mail addresses of the each of the recipients;
6. Session key, which can be used by recipient e-mail clients to authenticate each other; and
7. Attachment server ID, α .

Attachment Server ID, α , may include a domain name or an IP address of the attachment server 110.

In step 420, the file in the e-mail is replaced by the proxy attachment. Process 400 then transmits an attachment information to the attachment server 110 in step 425. The attachment information includes, but is not limited to, the attachment ID, the secret matrix used to slice the file, a recipient list, and the session key. Subsequently, the e-mail containing the proxy attachment is transmitted to the recipients in step 430 via an e-mail server. One skilled in the art will recognise that the slicing of the file may be performed by the attachment server 110 instead of the e-mail clients 121-124 used by a sender 320 without departing from the invention.

25

In an embodiment of the invention, it is assumed that when an e-mail client starts up, the e-mail client will register its e-mail address and local IP address mapping to the server by sending the server the mapping information (r_i, I_{r_i}) where I_{r_i} is the IP address of r_i . One skilled in the art will recognise that in order to prevent

spoofing, the server may use whatever method to check that the IP address is real (by DNS query, reverse IP lookup, etc) without departing from the invention.

Figures 5 and 6 illustrate flow diagrams of processes 500 and 600 performed by a system in one of e-mail clients 121-124 used by a recipient attempting to receive an e-mail containing proxy attachment that has been sent to more than one recipient in accordance with an embodiment of this invention. The system is provided by processes stored in software, firmware, or hardware.

Process 500 begins in step 505 by generating a request for download information based on the proxy attachment and transmitting the request for download information. The request comprises attachment id, a , and list of slice IDs it requires, $S = \{S_1, S_2, \dots, S_n\}$.

Based on the request, the server will find the attachment information associated to the request. The server then generates and transmits a list of download information, $D(a, s_i) = \{I_i\}$, where a is the attachment ID, s_i is the slice ID and $i = \{1, 2, 3, \dots, n\}$ is the number of slices available for download. Each download information contains minimum of one address that has the slice s_i of a . Assuming a first recipient, r_1 , requests for download information, the server will send $D(a, s_i) = \{I_{i,svr}\}$ for $i = \{1, 2, 3, \dots, n\}$, where $I_{i,svr}$ is the IP address of the slice store for the respective slices i . This is because the slices may be stored in different slice stores. Assuming a second recipient, r_2 , request for download information after r_1 downloaded some or all the slices, the server will send $D(a, s_i) = \{I_{i,svr}, I_{i,r1}\}$ for $i = \{1, 2, 3, \dots, n\}$, where $I_{i,r1}$ is the IP address of r_1 . Further details of generating of the list of download information will be described below with reference to figures 7 and 8.

In step 510, the e-mail client receives the list of download information. Process 500 then scans the list of download information in step 515 and removes any of download information that the e-mail client already retrieved. After step 515,

process 500 randomly selects a subset of download information in step 520. The subset of download information is a portion of the remaining download information. Process 500 then determines if each of the download information in the selected subset of download information contains more than one IP address in step 522. If
5 there is only one IP address, it means that the e-mail client is a first recipient and process 500 proceeds to step 540 to retrieve the slices from the slice stores based on the IP address indicated in the selected subset of download information. The random selection of a subset of download information for retrievable of slices helps to achieve some parallel downloads to improve performance. The random selection of a subset
10 of download information may also be controlled to cater to the possibility that more than one e-mail client may start the download at the same time. At this juncture, none of the slices are available locally on any of the e-mail clients and the slice stores may be inundated with too many requests for the same slices. After step 540, process 500 repeats from step 505.

15

In step 522, if there is more than one IP address in the download information, process 500 proceeds to step 525 to ping the list of IP address in the download information to determine the download speed for each of the IP addresses in the list. The pinging of the IP address is via a simple Internet Control Message Protocol
20 (ICMP) ping. This means that the e-mail client is not a first recipient and that some or all the slices can be downloaded locally from one of the e-mail clients who have completed downloading some or all the slices. The process of determining the download speed may also determine the round trip delay to each IP addresses in the list of download information.

25

In step 530, process 500 re-orders the download information to a sorted list based on the download speed from the fastest download speed to the slowest download speed. By re-ordering the list of download information, the e-mail client may download from one of the recipients rather than from the slice stores. Taking the

example from the above for the second recipient, the re-ordered download information for slice i may be $D(a, s_i) = \{I_{i,rl}, I_{i,svr}\}$ for $i = \{1, 2, 3, \dots, n\}$.

One skilled in the art will recognise that besides determining the download speed, the system may also be extended to include calculation of the network speed/throughput and charges incurred for a particular network link for re-ordering the list in the download information without departing from the invention. Essentially, the re-ordering of the list of download information may be modified based on considering the cost for downloading the slices. For example, downloading of slices from slice stores may be calculated based on per download. Hence, in order to reduce cost, the system may impose that downloading from the slice stores will not be permitted after a certain number of recipients have obtained some or all slices from the slice stores.

Based on the re-ordered list of download information, the e-mail client retrieves at least one slice accordingly in step 535. Process 600 continues from step 535 of process 500.

Process 600 begins with step 605 by monitoring for a slice retrieved. After a slice is retrieved, a checksum is calculated for the retrieved slice in step 610. Process 600 then compares the checksum with the checksum in the proxy attachment in the e-mail that is associated to the slice. This is to prevent slices with incorrect data being injected in the system. If the checksum is not correct, process 600 proceeds to step 620 by removing the slice from memory. Process 600 then proceeds to step 625. If checksum is correct, process 600 update the server on the slice the recipient has obtained in step 622. Further details regarding the updating of the server on the slices obtained by recipients will be described below with reference to figures 7 and 9. Process 600 then proceeds to step 625 to determine if all slices of the selected subset of download information has been retrieved. Assuming the subset of download information contains five download information, process 600 will

determine if five slices have been retrieved. If all slices of the subset of download information are not retrieved, process 600 repeats from step 605. Otherwise, process 600 proceeds to step 630. For brevity, process 600 illustrates a process of retrieving one slice of the sliced file only. Hence, one skilled in the art will recognise
5 that multiple iterations of steps 605-625 in process 600 may be performed at one time if step 535 of process 500 retrieves more than one slice from different sources.

In step 630, process 600 determines if a minimum number of slices, m , needed to re-combine the file have been received. The determination is performed in
10 the following manner. The total number of slices retrieved is determined. This total number is compared with the m from the proxy attachment. If total number is equal to or more than m , the minimum number of slices is achieved. If the recipient does not have the required minimum of slices, process 600 proceeds to step 505 of process 500. If the recipient has the minimum m slices it requires, process 600
15 generates a request for the secret matrix to the server in step 635. For added security, the e-mail client needs to use the session key attached in the proxy attachment to request for the secret matrix. Process 600 then transmit the request for matrix to the server in step 640.

20 In response to the request for secret matrix, the server transmits a secret matrix for re-combining the slices to form the file. Further details of the server sending the secret matrix will be described below with reference to figures 7 and 10.

In step 645, process 600 receives the secret matrix. Process 600 then re-
25 combines the slices to form the file using the secret matrix. Process 600 ends after step 645.

After combining the slices to form the original file, the individual slices may be cached for a period of time in the memory of the e-mail client so that it is available for
30 download by other recipients. When the e-mail client purges the slices, the client

updates the server so that server will remove its IP addresses from the download information.

5 In accordance with one embodiment of the invention, an e-mail client may delete the slices but fail to update the server. In such circumstances, the e-mail client may receive stale download information. When a recipient tries to download the slices from the e-mail client who has deleted the slices, the recipient will not be able to download the intended slices. The recipient attempts to download the slices from the next IP address available in the list. The recipient may also inform the server that
10 the slices are no longer available on the e-mail client who has deleted the slices so that the server can purge IP address of the e-mail client who has deleted the slices from the download information.

During the processes of 500 and 600, the system may be optimised by
15 selecting a subset of download information to retrieve some of the slices from some of the recipients in step 635. Further the e-mail client may randomise the selection of the subset of download information. By doing so, it increases the chances that different e-mail clients may download different slices. This is to allow the e-mail clients to download different slices from each other. Each client may periodically
20 refreshes its copy of list of download information, $D(a, s_i)$, from the server, to determine if there are better, lower-cost download sources. By continuously re-ordering the list in the download information, it chooses the optimum download source, based on its perspective.

25 Figure 7 illustrates a flow diagram of process 700 performed by a system in attachment server 110 in accordance with an embodiment of this invention. The system is provided by processes stored in software, firmware, or hardware. Process 700 begins in step 705 by monitoring and receiving data from e-mail clients. Process 700 then determines the type of data received from e-mail clients in step 710. If the
30 data is an attachment information, process 700 proceeds to step 715 to create a new

file entry in the register containing attachment information and store the attachment information in the new file entry. The attachment information is stored in a persistent storage of the attachment server 110. A persistent storage is a storage device that preserves data when its power is off. The attachment information may also be stored
5 in the memory of the attachment server for faster retrieval. The exact configuration on the usage of both persistent storage and memory is left as a design choice to those skilled in the art.

If the data is a request for download information, process 700 proceeds to
10 step 720 to retrieve the attachment information according to the request and based on the attachment information, generates and transmit a list of download information to the e-mail client. Further details of creating a generating and transmitting the list of download information will be described below with reference to figure 8.

15 If the data is an update on the slices retrieved, process 700 proceeds to step 725 to update the attachment information by including or removing the IP address of the e-mail client. Further details of creating a updating the attachment information will be described below with reference to figure 9.

20 If the data is a request for secret matrix, process 700 proceeds to step 730 to retrieve the secret matrix according to the request and transmit the secret matrix. Further details of retrieving and transmitting the secret matrix will be described below with reference to figure 10. Process 700 ends after step 715, 720, 725 or 730.

25 Figure 8 illustrates a flow diagram of process 800 performed by a system in attachment server 110 in accordance with an embodiment of this invention. The system is provided by processes stored in software, firmware, or hardware. Process 800 begins in step 805 by reading the download request received from an e-mail client. The request includes attachment id, a , and list of slice IDs it requires, $S = \{S_1,$
30 $S_2, \dots, S_n\}$. Based on the information in the request, an attachment information that correspond to the request is retrieved in step 810.

In step 815, process 800 reads the IP address for each slice and subsequently generates a download information, $D(a, s_i)$, for each slice in step 820. A download information contains an address that has the slice s_i of a . For example, if
5 slice number 3, s_3 , is only available on slice store number 2, $l_{3,svr2}$, then the download information for s_3 is $D(a, s_3) = \{l_{3,svr2}\}$. If slice number 3, s_3 , is now available on slice store number 2 and recipient number 5, r_5 , then the download information for s_3 is $D(a, s_3) = \{l_{3,svr2}, l_{3,r5}\}$.

10 After a download information has been generated for each slice, process 800 transmits the list of download information to the recipient who requested for the download information in step 825. The download information is also stored in the attachment server 110. Process 800 ends after step 825.

15 Figure 9 illustrates a flow diagram of process 900 performed by a system in attachment server 110 in accordance with an embodiment of this invention. The system is provided by processes stored in software, firmware, or hardware. Process 900 begins in step 905 by reading the updated information received from an e-mail client. The updated information contains the attachment ID, a , and slice number, s_i ,
20 that the recipient has obtained. Based on the updated information, an attachment information that correspond to the updated information is retrieved in step 910. Process 900 then append the attachment information to include the IP address of the recipient to the slices that is available from the recipient. The previously generated download information for the particular attachment information can also be updated
25 directly. Process 900 ends after step 915.

Figure 10 illustrates a flow diagram of process 1000 performed by a system in attachment server 110 in accordance with an embodiment of this invention. The system is provided by processes stored in software, firmware, or hardware. Process
30 1000 begins in step 1005 by reading the request. The request should at least contain

the attachment ID, *a*. Based on the request, an attachment information that correspond to the request is retrieved in step 1010.

5 The secret matrix is then read from the attachment information and transmitted to the recipient in step 1015. Optionally, process 900 may first determine if the recipient requesting for the secret matrix has obtained the minimum number of slices required for combining the slices to generate the original document prior to transmitting the secret matrix to the recipient. The determination may be performed in the following manner. The number of slices available for download from the
10 recipient is determined. The number of slices is then compared with m in the attachment information. If the number of slices is equal to or more than m , process 1000 transmits the secret matrix. Otherwise, process 1000 may request for all slices available on the recipient prior to sending the secret matrix.

15 The above is a description of embodiments of a system in accordance with the invention as set forth below. It is envisioned that those skilled in the art can and will design alternative embodiments of this invention based upon this invention that infringe on this invention as set forth in the following claims.

Claims

1. A method for sending and receiving file via e-mail among a plurality of e-mail clients via a server and a plurality of slice stores comprising:
 - creating an e-mail on a first of said plurality of e-mail clients;
 - 5 slicing file in said created e-mail to form a plurality of slices using a secret matrix;
 - transmitting each of said plurality of slices to various ones of said plurality of slice stores;
 - generating a proxy attachment containing information of said sliced file;
 - 10 replacing said file in said e-mail with said proxy attachment;
 - transmitting an attachment information to said server; and
 - transmitting said e-mail to a plurality of recipients from said plurality of e-mail clients, wherein a file size of said proxy attachment is smaller than a file size of said file.
- 15 2. The method according to claim 1 further comprising generating said attachment information wherein said attachment information comprises: an attachment ID; said secret matrix, e-mail address of said plurality of e-mail clients, and a session key.
- 20 3. The method according to claim 1 wherein in response to receiving said attachment information from said first of said plurality of e-mail clients, said server performs the following steps:
 - creating a new file entry in a persistent storage of said server.
- 25 4. The method according to claim 3 further comprising storing said attachment information in said new file entry in said persistent storage of said server.
5. The method according to claim 1 further comprising:
 - 30 determining said file size of said file is above a predetermined file size.

6. The method according to claim 1 wherein said proxy attachment comprises an attachment ID, a ; IDs of each of said plurality of slices, $S = \{S_1, S_2, \dots, S_n\}$ where S is said plurality of slices and n is the number of slices said file has been sliced into; a minimum number of slices, m , needed to re-combine the file where $m < n$; a checksum for each of said plurality of slices; e-mail address of each of said plurality of e-mail clients, $R = \{r_1, r_2, r_3, \dots, r_i\}$, wherein R is the set of recipient e-mail address, $r_1, r_2, r_3, \dots, r_i$ are the e-mail address of each of said recipient; a session key for authentication; and server ID, α , containing an IP address of said server.
7. The method according to claim 6 wherein in response to receiving said e-mail by a one of said plurality of e-mail clients used by one of said plurality of recipients, said one of said plurality of e-mail clients performs the following step:
transmitting a request to said server for retrieving said file, wherein said request includes said attachment ID, a , and a list of slice IDs, $S = \{S_1, S_2, \dots, S_n\}$.
8. The method according to claim 5 wherein in response to receiving said request for retrieving said file, said server performs the following steps:
retrieving said attachment information according to said request;
generating a list of download information based on the retrieved attachment information; and
transmitting said list of download information to said one of said plurality of e-mail clients.
9. The method according to claim 8 wherein said list of download information comprises:
 $D(a, s_i) = \{I_{i,r}\}$, where a is the attachment ID, s_i is the slice ID, $I_{i,r}$ is the IP address of recipient, r , having i^{th} slice, and $i = \{1, 2, 3, \dots, n\}$ is the number of slices available for download.

10. The method according to claim 8 wherein in response to receiving said download information from said server, said one of said plurality of e-mail clients performs the following step:

retrieving said plurality of slices based on said list of download information.

5

11. The method according to claim 10 wherein said step of retrieving said plurality of slices comprises:

scanning said list of download information;

removing download information from said list of download information which

10 has already been retrieved;

selecting a subset of download information;

determining if each of said download information in said selected subset of download information has more than one IP address; and

15 re-ordering said IP addresses in response to determining that each of said download information remaining in said list of download information has more than one IP address.

12. The method according to claim 11 wherein said step of re-ordering said IP addresses comprises:

20 determining a download cost for said IP addresses in each of said download information in said selected subset of download information;

arranging said IP addresses in each of said download information according to lowest download cost to highest download cost.

25 13. The method according to claim 12 wherein said step of determining said download cost comprises:

pinging each of said IP address to determine a download speed of each of the IP address.

30 14. The method according to claim 13 wherein said step of ping is a simple Internet Control Message Protocol (ICMP) ping.

15. The method according to claim 12 wherein said step of retrieving said file comprises:
- retrieving each of said slices of said selected subset of download information
5 from an IP address having a lowest download cost.
16. The method according to claim 15 further comprising:
- monitoring for one of said slices in said subset of download information being retrieved; and
10 determining a validity of said slice being retrieved.
17. The method according to claim 16 wherein said step of determining said validity of said slice retrieved comprising:
- calculating a checksum for said slice retrieved;
15 determining said checksum is valid based on a checksum in said proxy attachment; and
removing said retrieve slice in response to said slice being not valid.
18. The method according to claim 16 wherein said step of determining said validity of said slice retrieved comprising:
- calculating a checksum for said slice retrieved;
determining said checksum is valid based on a checksum in said proxy attachment; and
updating said server on retrieval of said retrieve slice in response to said slice
25 being valid.
19. The method according to any one of claims 17 and 18 wherein said step of retrieving said file further comprises:
- determining all slices of said subset of download information are retrieved
30 determining a total number of slices retrieved in response to said determination that all slices of said subset of download information are retrieved;

comparing said total number of slices with said m in said proxy attachment;
generating a request for said secret matrix in response to said total number of
slices being equal to or more than said m in said proxy attachment; and
transmitting said request to said server.

5

20. The method according to claim 19 wherein in response to receiving said
request for said secret matrix, said server performs the following steps:

retrieving said secret matrix associated to said request from a persistent
storage of said server;

10 transmitting said secret matrix to said one of said plurality of e-mail clients.

21. The method according to claim 20 wherein in response to receiving said
secret matrix from said server, said one of said plurality of e-mail clients performs the
following step:

15 re-combining said slices based on said secret matrix to form said file.

22. The method according to claim 11 wherein said step of retrieving said file
further comprises:

retrieving said slices of said selected subset of download information from
20 said at least one slice store in response to determining that only one IP address is
available.

23. The method according to claim 23 further comprising:

monitoring for one of said slices in said subset of download information being
25 retrieved; and
determining a validity of said slice being retrieved.

24. The method according to claim 23 wherein said step of determining said
validity of said slice retrieved comprising:

30 calculating a checksum for said slice retrieved;

determining said checksum is valid based on a checksum in said proxy attachment; and

removing said retrieve slice in response to said slice being not valid.

5 25. The method according to claim 23 wherein said step of determining said validity of said slice retrieved comprising:

calculating a checksum for said slice retrieved;

determining said checksum is valid based on a checksum in said proxy attachment; and

10 updating said server on retrieval of said retrieve slice in response to said slice being valid.

26. The method according to any one of claims 24 and 25 wherein said step of retrieving said file further comprises:

15 determining all slices of said subset of download information are retrieved;

determining a total number of slices retrieved in response to said

determination that all slices of said subset of download information are retrieved;

comparing said total number of slices with said m in said proxy attachment;

20 generating a request for said secret matrix in response to said total number of slices being equal to or more than said m in said proxy attachment; and

transmitting said request to said server.

27. The method according to claim 26 wherein in response to receiving said request for said secret matrix, said server performs the following steps:

25 retrieving said secret matrix from said persistent storage;

transmitting said secret matrix to said one of said plurality of e-mail clients.

28. The method according to claim 27 wherein in response to receiving said secret matrix from said server, said one of said plurality of e-mail clients performs the following step:

30

re-combining said slices based on said secret matrix to form said file.