Title: A SYSTEM AND METHOD FOR STREAMING SECURED DATA

Abstract: A system and method for encrypting or decrypting a plurality of data packets from a data stream using an identifier that associates a data packet from a data stream with a key. This allows encryption or decryption of a portion of a data stream using a key. The system and method also provide permissions that allow a recipient access to a portion of the data stream based on the permissions.
Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

— with international search report (Art. 21(3))
A SYSTEM AND METHOD FOR STREAMING SECURED DATA

Technical Field

The present invention relates to a system and method for streaming secured data, and particularly, although not exclusively to a system and method for streaming secured data objects which are encrypted.

Background

Transferring information electronically through the Internet or another public telecommunication network (such as wired or wireless telephone services) is a cost effective solution for distributing information. However, as much of the Internet operates on public infrastructure, sensitive or confidential information sent through the Internet may be accessible to unauthorised parties.

To address these security concerns, corporations and other users may choose to encrypt the information before transmitting the data over a public network. One approach is to use encryption software, such as "Zip" programs which offer an encryption routine to encrypt the data before it is transmitted over the public network.

Although such encryption software provides some level of security, all such software has a fundamental flaw, in that the encryption process embeds the decryption key within the encrypted data object itself. As such, it is possible for a hacker to use brute force or other suitable methods to decrypt the data object since the necessary components to decrypt the data object are all integrated within the encrypted object.
Moreover, where encrypted data is streamed (when a media file, such as an audio or video file, is constantly received by and presented to a user while being delivered by a server), the packets of data that make up the stream all share the same decryption key. As such, if a third party is able to access the key for one packet, then they can access the data in all packets of the stream.

**Summary of the Invention**

In accordance with a first aspect of the present invention, there is provided a method for decrypting a plurality of data packets comprising the steps of:

- receiving a data packet from a sender;
- utilising an identifier to identify the data packet and retrieving a key associated with the identifier to decrypt the encrypted data into decrypted data.

In an embodiment, the method comprises the further step of, iterating the method steps for each data packet received from a sender.

In an embodiment, the method comprises the further step of buffering a plurality of data packets before display.

In an embodiment, the method comprises the further step of displaying at least one of the plurality of decrypted data packets to a user.

In an embodiment, the method comprises the further step of storing at least one of the plurality of decrypted data packets.

In an embodiment, the identifier uniquely identifies the data packet.
In an embodiment, the identifier associates the key and the data packet. In an embodiment the identifier can be way of associating a key with a particular data packet, including a rule or rules to associate a key with a particular data packet. The identifier may be a permission or rule that is associated with a recipient that governs access to one or more data packets and associated keys. The identifier may be variable depending on the type of data being streamed and the resources available. In one example the identifier can be a rule to sequentially encrypt every packet, each sequentially encrypted data packet being associated with a different key. In another example an identifier may be generated for a plurality of packets such that one key is used to encrypt or decrypt a plurality of data packets. Other identifiers and identifier rules can be utilised in the system.

An advantage of at least an embodiment of the present invention is that when data is streamed (in packets or data blocks, for example), different parts of the data stream (e.g. different data packets) may be associated with separate keys such that the data parts of the stream may be accessed separately. For example, different recipients may have different access permissions for different portions of the data stream, such that recipients can access parts of the data stream (e.g. a particular set of data packets) based on access permissions. Different data portions can have different access permissions or rules.

In accordance with a second aspect, the present invention provides a method of encrypting data packets to be streamed, comprising the steps of:
- receiving at least one data packet;
utilising a key to encrypt the at least one data packet; and

generating an identifier that identifies the key and the data packet;

wherein the identifier is utilisable to associate the key and the data packet.

In an embodiment, the key is saved in a first location.

In an embodiment, the encrypted data packet is stored in a second location.

In accordance with a third aspect, the present invention provides a system for decrypting a plurality of data packets comprising:

a receiving module arranged to receive a data packet from a sender;

a module arranged to utilise an identifier to identify the data packet; and

a module arranged to retrieve a key associated with the identifier to decrypt the encrypted data into decrypted data.

In accordance with a fourth aspect, the present invention provides a system for encrypting a plurality of data packets comprising:

a module arranged to receive one or more data packets;

a module arranged to utilise a key to encrypt one or more data packets; and

a module arranged to generate an identifier that identifies the key and the data packet;

wherein the identifier is utilisable to associate the key and the data packet.
In an embodiment, the present invention can be used to encrypt or decrypt a plurality of data packets in a data stream.

In accordance with a fifth aspect, the present invention provides a computer program comprising at least one instruction for controlling a computer system to implement a method in accordance with any one of any one of the embodiments of the first aspect of the present invention.

In accordance with a sixth aspect, the present invention provides a computer readable medium providing a computer program in accordance with the fourth aspect of the present invention.

In accordance with a seventh aspect, the present invention provides a data signal comprising a computer program in accordance with the fifth aspect of the invention.

In accordance with an eighth aspect, the present invention provides a computer program comprising at least one instruction for controlling a computer system to implement a method in accordance with any one of the embodiments of the second aspect of the present invention.

In accordance with a ninth aspect, the present invention provides a computer readable medium providing a computer program in accordance with the ninth aspect of the present invention.

In accordance with a tenth aspect, the present invention provides a data signal comprising a computer program in accordance with the eighth aspect of the present invention.

In accordance with an eleventh aspect, the present invention provides a method of processing data in a data stream comprising the steps of:
associating a plurality of decryption keys with a respective plurality of portions of the data; whereby each decryption key is arranged to enable decryption of the associated portion of data of the data stream.

In accordance with a twelfth aspect, the present invention provides a system for processing data in a data stream comprising:

- a module arranged to associate a plurality of decryption keys with a respective plurality of portions of the data stream; and
- a module arranged to decrypt a portion of data of the data stream, whereby each decryption key is arranged to enable decryption of the associated portion of data of the data stream.

Brief Description of the Drawings

Embodiments of the present invention will now be described, by way of example, with reference to the accompanying drawings in which:

- Figure 1 is a schematic block diagram of a system for distributing secured data in accordance with one embodiment of the present invention;
- Figure 2 is a block diagram of a system for securing data in accordance with one embodiment of the present invention;
- Figure 3 is a block diagram of a system for streaming secured data in accordance with another embodiment of the present invention; and
- Figure 4 is a flow diagram of a system in accordance with Figure 3.
Detailed Description of the Preferred Embodiment

Referring to Figure 1, there is illustrated an embodiment of the present invention. This embodiment is arranged to provide a system for distributing secured data comprising:

- a module arranged to receive a request from a data recipient to access the encrypted data;
- an authentication routine arranged to authenticate the request and whereupon the request is authenticated;
- a decrypting processor arranged to retrieve a key to decrypt the encrypted data into decrypted data; and
- a communication interface arranged to distribute the decrypt data to the data recipient.

In this example embodiment, the module, authentication routine, decrypting processor may be implemented by one or more electronics circuits, computers or computing devices having an appropriate logic, software, hardware or any combination thereof programmed to operate with the computing devices. The computer may be implemented by any computing architecture, including stand-alone PC, client/server architecture "dumb" terminal/mainframe architecture, mobile computing devices such as smart phones, mobile phones or the like or any other appropriate architecture. In some embodiments, the computing device may also be appropriately programmed to implement the invention.

Referring to Figure 1 there is a shown a schematic diagram of a system for accessing secured data which in this embodiment comprises a computing device which can include any client or server device. In this embodiment, the computing device is a server 100. The server 100, or any suitable computing device comprises suitable
components necessary to receive, store and execute appropriate computer instructions. The components may include a processing unit 102, read-only memory (ROM) 104, random access memory (RAM) 106, and input/output devices such as disk drives 108, input devices 110 such as an Ethernet port, a USB port, etc. Display 112 such as a liquid crystal display, a light emitting display or any other suitable display and communications links 114. The server 100 includes instructions that may be included in ROM 104, RAM 106 or disk drives 108 and may be executed by the processing unit 102. There may be provided a plurality of communication links 114 which may variously connect to one or more computing devices such as a server, personal computers, terminals, wireless or handheld computing devices. At least one of a plurality of communications link may be connected to an external computing network through a telephone line, optical fibre, wireless connection or other type of communications link.

The server 100 may include storage devices such as a disk drive 108 which may encompass solid state drives, hard disk drives, optical drives or magnetic tape drives. The server 100 may also use a single disk drive or multiple disk drives. The server 100 may also have a suitable operating system 116 which resides on the disk drive or in the ROM of the server 100.

The system has a database 120 residing on a disk or other storage device which is arranged to store at least one data record relating to data used by the server 100 to provide the function of the system for accessing secured data. The database 120 is in communication with an interface 202, which is implemented by computer software residing on the server 100. The interface 202 provides a means by which a user may input commands, instructions or
requests to the server 100 for execution or processing. The interface 202 may be implemented with input devices such as keyboards, mouse or, in another example embodiment the interface 202 may be arranged to receive inputs, requests or data through a network connection, including Ethernet, Wi-Fi, Fire-wire, USB or the like.

With reference to Figure 2, there is illustrated a block diagram of an embodiment of a system for securing data. In this embodiment, the system may be implemented with a computer server 200 arranged to be connected to a communication network such as the Internet, Intranet, VPN etc, wherein the computer server 200 is arranged to communicate with other computing or communication devices 204, 206 via the communication network to provide an encryption service.

As shown, this embodiment comprises a server 200 which is arranged to receive an encryption request 202 from a sender computing device 204 operated by a user, data sender, processor or controller wanting to encrypt a data object for transmission to another recipient user 206, computer, processor or controller. In this example embodiment, the encryption request 202 may contain information relating to the data object that is to be encrypted by the sending computing device 204. This information may include, but not limited to:

1. The identity of a sender 204;
2. The identity of a recipient 206;
3. Filenames of any files to be encrypted;
4. File size, dates, properties, permissions settings and other attributes;
5. The access permissions of the recipient 206;
6. The address or reference of the recipient 206;
7. Any other information relating to the security settings or the data object that is to be encrypted which may be required to encrypt the file.

Once the encryption request 202 is received by the server 200, the server 200 is arranged to generate one or more keys which can be used to encrypt the data object. The one or more keys 208 may then be sent to the sender computing device 204 which has sent the encryption request 202 to the server 200. Once received, the one or more keys 208 may then be used by the computing device 204 to encrypt the data object such that an encrypted data object 210 is generated.

Preferably, the encryption process on the computing device operates by encrypting the data object 210 such that the key 208 is not in any way integrated into the encrypted data object 210. As a result, the encrypted data object 210 cannot be decrypted by a hacker or malicious party who is able to obtain an authorized copy of the encrypted data object 210 since the encrypted data object 210 itself is unable to provide the necessary information (e.g. the key 208) for the hacker to decrypt the file. This embodiment is advantageous in that the encrypted data object 210 is highly secured since the key 210 needed to decrypt the file is not incorporated within the object 210 itself.

After the data object is encrypted, the sender computing device 204 may then be operated by its user, processor or controller to send the encrypted data object 210 to a recipient 206. As the encrypted data object 210 is now secured, it may be sent through a computer network email, virtual storage servers or provided to the recipient in the form of digital media such as CDs, DVDs,
Blu-Rays, USB storage or the like. Preferably, in some situations, some form of security consideration is still put into practice with the transmission of the encrypted data object 210 for best practice.

Once the recipient user 206 receives the encrypted data object 210, the recipient user 206 may then contact the server 200 with a request to retrieve the necessary keys to decrypt the data object 210. In one embodiment, the server 200 may enforce an authentication process (212) on the recipient 206 by checking and validating the identity of the recipient 206 prior to providing a key 214 to the recipient. The authentication process (212) may include a login/password check, a biometric check, a time delayed validation process, a telephone code check, a pass key check, an IP address check or a combination of one or more of the systems described thereof.

After the recipient user 206 is authenticated by the server 200 and is authorized to decrypt the data object 210, a key 214 may be provided to the recipient user 206 to decrypt the file. In one example embodiment, the recipient user 206 may be given a key 214 which only decrypts certain portions of the encrypted data object 210 such that only portions of the data may be released to the recipient user 206. In another embodiment, the decryption of the data is restrictive such that certain usage permissions are enforced on the recipient 206. In these examples, it may be necessary to encrypt the data object with necessary information for third party software to control and enforce these permission settings. Examples of these third party software includes Secure Word or Adobe Acrobat reader which have permission controls capable of limited the manipulation of a data file.
Alternative embodiments of a system for securing data are also described in WO 2009/079708 which is incorporated herein by reference. These embodiments are advantageous in that the encryption key 208 which can be used to decrypt an encrypted object is removed from the encrypted data object 210. As such the encrypted data object may be transmitted in a less secure, whilst more convenient channel since even in the event that the encrypted data object 210 is copied by an unauthorised user, the object cannot be easily decrypted with known methods of decryption since the key is not within the encrypted object.

In another embodiment, the server 200 is arranged to provide dummy keys to the sender computing device 204 and the recipient computing device 206. By transmitting and utilising dummy keys in the encryption process, hackers or other malicious parties listening to the transmissions from the server 200 may receive a plurality of keys without any reference or knowledge as to which of the dummy keys can in fact be used to decrypt the data object. The dummy keys may also be integrated with the genuine key such that the permutations between the dummy keys and the genuine keys render it unfeasible or impractical for a hacker to use the data for any meaningful purpose.

With reference to Figure 3, there is illustrated another embodiment of a system for streaming data. In this embodiment, system 300 comprises an encryption module 306, a decryption module 304 and a security module 302. These three modules 302, 304, 306 may be implemented with computer software, hardware or a combination of software and hardware operating on a single computing device or multiple computing devices to stream encrypted data over a communication network. In certain security or operating
environments, it is preferable that each of the modules 302, 304, 306 are implemented on individual computing devices, such as servers or banks of servers and deployed at one or more geographical locations. The modules 302, 304 and 306 are logically or physically separated. In one example, the modules 203, 204 and 306 are implemented on individual computing devices and each of the computing devices is geographically separated from the other devices. In another example, the modules 302, 304 and 306 are implemented on a single computing device but are logically separated from each other. The modules 302, 304 and 306 are arranged to communicate with each other.

In this example, the security module 302 is arranged to generate and store a key which can be utilised to encrypt or decrypt a data object. The security module 302 may be a server arranged to receive a request for a key to encrypt a data file, after which when the file is encrypted and is required to be decrypted, the key is then provided to a recipient 308 of the file after the recipient 308 has been authenticated. The security module 302 may be a server 302 connected to a network arranged to allow other computers 308 operated by users, routines, processors or the like to connect to the server with requests to generate or obtain a key to encrypt or decrypt a data object. In one embodiment, the security module 302 is implemented based on the server 200 described above, or in another embodiment, the security module 302 is implemented based on a system for securing data described with reference to WO 2009/079708 which was previous incorporated herein.

As illustrated in Figure 3, the encryption module 306 is arranged to receive an input data object for encryption and streaming to a recipient. The encryption
module 306 operates on a sender computing device. Preferably, the data stream received by the encryption module may be a large file which may not be easily transmitted through email (e.g. 20 megabytes or larger) or data streams which are generated or transmitted via a feed of unknown size or duration. Typical examples of these "data objects may include multimedia files such as movies or videos, market stock data, surveillance feeds, news feeds, communication feeds or the like.

Once the data stream is received by the encryption module 306, the module proceeds to divide the data stream into individual blocks or packets for transmission. The block size may be dependent on the expected size of the data stream and network resources which are available to the system 300. Once the data stream received by the encryption module 306 is sufficient to fill an individual block or packet, the encryption module 306 may then proceed to contact the security module 302 to obtain a key which can be used to encrypt the individual block, although in some implementations, a plurality of keys which can be used to encrypt one or more individual block may be requested in a single request to the encryption module 306.

The encryption module 306 can generate one or more identifiers that uniquely identify the key and the data packet, wherein the identifier is utilisable to associate the key and the data packet. The identifier can be stored at the security module or server 302. Alternatively the identifier is stored at a remote location such as a database, with the location information of the identifier being stored at the security module 302.

The identifier can be any suitable way of associating a data packet with a key or any rules that
relate the key to a particular data packet. The identifier is variable depending on the type of data being streamed and the resources available. In one example the identifier can be a rule to sequentially encrypt every packet. In another example an identifier may be generated for a plurality of packets such that one key is used to encrypt or decrypt a plurality of data packets. Other identifiers and identifier rules can be utilised in the system 300. In a further example the identifier may comprise a counter that keeps a track of the data packet and which key is used. The counter is updated every time a data packet is associated with a key.

The identifier can be generated by the sender utilising appropriate application or software program. Preferably the identifier can be generated by the encryption module 306 during the encryption process. The encryption module divides the data stream into individual data blocks or packets and uses a key to encrypt one or more blocks. During this process the encryption module can track which data block is associated with which key. The identifier is based on this tracking. The identifier is then transmitted to the security server 302 for storage. Alternatively the identifier may be transmitted to the recipient for use by the decryption module 304 during decryption of the data stream.

After the individual block has been encrypted by the encryption module 306, the encrypted block is then transmitted as an object to the decryption module 304, which may be a module 304 operating on a recipient's computing or communication device. In examples where the data stream concerned is for a video file, the decryption module 304 may be implemented as a piece of middleware between the communication interface of the recipient's
computing device and the video player or integrally within the video player itself. Once the encrypted data block or data packet is successfully received by the decryption module 304, the decryption module will then contact the security server 302 to obtain a key to decrypt the block. In some examples, the decryption module 304 must firstly authenticate the identity of the recipient before the key is provided by the security server 302.

Once the identity of the recipient is authenticated, the security module 302 can also provide the decryption module the identifier or provide the recipient location information of the identifier such that the recipient can access the identifier. The identifier is used to uniquely identify the data packet such that the decryption module 306 can access the correct key associated with a particular data block or data packet. The identifier allows the decryption module 306 to select the correct key and use it to decrypt the data packet. Such an operation occurs for the entire data stream until at least part or the whole data stream is decrypted.

Once the key is transmitted to the decryption module 304, the decryption module may then proceed to decrypt the encrypted block into a decrypted block. This decrypted block is then processed by another processor or routine or signal processing module, arranged to process the decrypted data. This may be a video player arranged to process individually received data blocks to broadcast a video, or a signal processing module arranged to process and/or aggregate each block it receives to produce an output stream for the recipient. In some instances, a plurality of encrypted blocks are sent or expected, and as such, the signal processing module is arranged to process, wait or query for each block it has received or is
expected to receive to generate an output data stream for the recipient.

In another embodiment, the system includes a key selector module arranged to select and determine the number of keys which can be used to encrypt or decrypt the data stream. In one example, the key selector module is arranged such that for each data block of the data stream that is encrypted or decrypted by the encryption module 306 or decryption module 304, a different key is requested from the security server 302. This is advantageous in that the security for the data stream is significantly enhanced since a hacker or authorised parties listening to the data stream being transmitted between the encryption module and the decryption module will be required to obtain a plurality of keys to decrypt the data stream. As it can be appreciated, if the data stream is of a large length, then the number of keys which must be obtained or otherwise "guessed" by a hacker or unauthorised party will render the malicious process of obtaining the data stream unfeasible.

In example illustrated in Figure 3, the system comprises an encryption key selector module 320 and a decryption key selection module 321. The encryption key selection module 320 is arranged to select and determine the number of keys required for encryption and the decryption key selector module 321 is arranged to select and determine the number of keys required for decryption. The number of keys required for decryption. The two key selection modules 320, 321 may be arranged to communicate with other. Alternatively the two modules 320, 321 communicate each other via the security server 302. The encryption key selection module 320 transmits the information regarding selected keys to the security server
302, such that the decryption key selector 321 can access this key information from the server 302. The encryption key module 320 is implemented on the same computing device or as part of the encryption module 306. The decryption key selection 321 is arranged to be implemented on the same computing device or as part of the decryption module 304. In this example, the decryption key selector 321 receives key information such as the number of keys from decryption key section. The key information allows the module 321 to select the correct keys for use by the decryption module 304. This key information regarding selected keys forms part of the identifier.

In another example, the key selector module may use a single key to encrypt a cluster of blocks. This may be a more feasible approach in certain circumstances depending on the resources available to the system 300.

In another alternative embodiment of the system 300, includes a dummy key integrator 312 and a dummy key separator 314. In this embodiment, the dummy key integrator 312 and the dummy key separator 314 is arranged to utilise dummy keys, or fake keys as part of the encryption and decryption process.

In one example, the dummy key integrator 312 is arranged to contact the security server 302 to obtain a real key which can be used to encrypt the data stream and in addition to obtaining the real key, the dummy key integrator 312 is arranged to receive from the security server 302 one or more dummy keys which are then also integrated with the real key when the real key is transmitted to the dummy key integrator 303. After the dummy key integrator 303 receives this cluster of keys, the dummy key integrator 303 is then able to determine the
real key and provide this to the encryption module 306 for encrypting portions of the data stream.

In this example, the decrypting module 304 is arranged to communicate with the dummy key separator 314, which has retrieved the combination of real and dummy keys from the security server 302. Once the dummy key separator 314 processes this combination received from the security module 304, the real key is then provided to the decryption module 304 to decrypt the encrypted data stream.

In some embodiments, the determination of the real key and the dummy keys can be made by the security module 302 or the integrator 312 or separator 314 by using specific algorithms depending on the data stream, user preferences, time or date stamps or any other information which can be used to identify the real key from the dummy keys.

These embodiments are advantageous in that keys which are transmitted between the security server 302 and the encryption or decryption module 306, 304 can be further secured since a hacker or unauthorised persons listening to the communications between the encryption or decryption module 306, 304 will also have to identity the real key from the plurality of dummy keys which are transmitted with the real key.

In an embodiment the key data (i.e. keys) for encryption/decryption may be stored at a remote location such as a separate database or separate server 330. This remote server or database is arranged to communicate with the security server 302. The security module 302 stores the location of the key data. The security module 302 is arranged to receive a request for key data for encryption/decryption from a user (e.g. a sender or
recipient) . The user is authenticated by the security module 302. Once the user is authenticated the security server 302 transmits the location of the keys to the user device such that the user can access the key data from the remote location. This embodiment of the system provides an additional layer of security because the key data is stored at a remote location that is separate from the security server (i.e. security module) 302. A hacker who hacks the security server 302 does not have access to the key data.

In yet another example the data may be stored in a separate storage media such as a smart card or USB key or CD ROM, in which case the security server 302 sends a direction to the decryption module 304 to direct the recipient to find the relevant storage media housing the key.

In one example the sender (i.e. encryption module 306) uses one or more keys to encrypt one or more data packets. The encryption module 304 is arranged to generate an identifier for each data packet that is encrypted such that the identifier can be used to associate each data packet with a unique key. The identifier can be transmitted to the security server 302, but is preferably stored at a remote location. To access the identifier the decryption module 304 must be authenticated. Once authenticated the server 302 transmits location information regarding the location of the identifier.

As part of the encryption process, the sender (via the encryption module 306) can assign permissions or rules to describe the data stream. The permissions or rules are arranged to control the manner in which the data within
the data stream (i.e. data blocks) can be interacted with or manipulated. In one example the permissions or rules include what level of users within any specific IP address range using specific type of computer software can access the data stream and what data within the stream they will have access to. These rules include, but are not limited to:

- the view, edit permission of the data within the data stream;
- the copy permission of the data within the data stream;
- the share permission of the data within the data stream;
- the redistribution of the data within the data stream;
- specific time periods allowed to access the data within the data stream;
- the parts of the data stream that can be viewed by a particular user;
- the location, both network or geographical location of the computer allowed to access the data within the data stream.

In one embodiment the decryption module 304 can comprise a receiver application that is used to stream the secured data and access the security server 302. The receiver application enforces the rules associated with a data stream that is accessed by the recipient. These permissions may form the identifier or be part of the identifier.

The sender computing device may include an application that can be used to establish an access control list which provides a list of recipient users authorised to receive and interact with the encrypted
(i.e. secured) data stream. The access control list is
stored at the security module 302. The access control
list can in one example also define the authentication
scheme necessary for the recipient to be authenticated.

In one example the data stream may comprise a stream
of video data. The video data stream may comprise a
plurality of channels. The different channels within the
stream may be associated with different keys. The
different channels in the stream may also be associated
with different permissions or rules for access by a
recipient. For example, one recipient may have
permissions to access all the channels. Another recipient
may be only limited to access a restricted number of
channels based on the permissions e.g. access the first
two channels.

With reference to Figure 4, there is illustrated a
flow diagram arranged to show each process of the
operation of the system of Figure 3. In one embodiment,
the encryption module 306 is arranged to receive a data
stream (402). After which, the encryption module 306
divides the stream into individual data blocks (404) and
communicates with the security module 302 to obtain a key
for a specific block of data (406). Once the key is
obtained, the key is used to encrypt the data block (408)
and is prepared for transmission to the recipient.

Once a recipient is authenticated, a connection is
then established with the authenticated recipient (409).
After which, the connection may then be used to transmit
the encrypted data blocks to the recipient's decryption
module 304 (410).

The recipient may then obtain a key (414) from the
security server 302 and proceeds to decrypt the received
encrypted data block (416). The decrypted data block is
then provided to the recipient, and may be processed by a recipient processor, such as the processor unit of a recipient computing device. The decrypted data blocks may be buffered prior to displaying to the recipient, by the recipient processor.

In some embodiments, the system 300 may be used in different streaming systems and applications to stream data from a source to one or more recipients. These streaming systems and applications may include, without limitation, video or sound streaming applications, telephony applications, communication systems or the like. These embodiments may be advantageous in that partial blocks of data which belongs to a complete data file may be encrypted and decrypted without having to transmit the entire complete data file, and thereby enhancing the security of some streaming systems and applications.

In a further example the sender can encrypt a data stream and make the data stream available for data streaming. The sender may implement the encryption process to create a secured data stream without a known end point. The secured data stream may always be available for streaming by an authorised recipient. A recipient who wants to stream the secured data stream may receive information regarding the data stream from the sender. The recipient can access and decrypt the data stream once authenticated. Once authenticated a connection is formed between the sender and recipient such that the recipient can stream the secured data stream from the sender. The recipient can use the decryption process described above.

Although not required, the embodiments described with reference to the Figures can be implemented as an application programming interface (API) or as a series of
libraries for use by a developer or can be included within another software application, such as a terminal or personal computer operating system or a portable computing device operating system. Generally, as program modules include routines, programs, objects, components and data files assisting in the performance of particular functions, the skilled person will understand that the functionality of the software application may be distributed across a number of routines, objects or components to achieve the same functionality desired herein.

In a further alternate embodiment the identifier could be a rule or permission associated with the person receiving the data stream. The identifier could be an access code associated with the recipient. The recipient could automatically have the identifier e.g. a password or smart card. The recipient can access parts of the data stream (i.e. particular data packets) that the recipient is authorised to receive based on the identifier.

A further advantage of at least one of these embodiments is that by decrypting the encrypted data objects on the remote client module 306 is that a recipient user does not need to have any specialised decryption software to be installed on the recipient computer system or device. As such, the costs to use the system for distributing secured data is reduced as the cost of additional software and its associated maintenance and training is avoided.

It will also be appreciated that where the methods and systems of the present invention are either wholly implemented by computing system or partly implemented by computing systems then any appropriate computing system architecture may be utilised. This will include stand
alone computers, network computers and dedicated hardware devices. Where the terms "computing system" and "computing device" are used, these terms are intended to cover any appropriate arrangement of computer hardware capable of implementing the function described.

It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

Any reference to prior art contained herein is not to be taken as an admission that the information is common general knowledge, unless otherwise indicated.
CLAIMS:

1. A method for decrypting a plurality of data packets comprising the steps of:
   5 receiving a data packet from a sender;
   utilising an identifier to identify the data packet and;
   retrieving a key associated with the identifier to decrypt the encrypted data into decrypted data.

2. A method in accordance with Claim 1, the method comprises the further step of, iterating the method steps for each data packet received from a sender.

3. A method in accordance with Claim 1 or 2, the method comprises the further step of buffering a plurality of data packets before display.

4. A method in accordance with Claims 1, 2 or 3, the method comprises the further step of displaying at least one of the plurality of decrypted data packets to a user.

5. A method in accordance with any one of Claims 1 to 4, the method comprises the further step of storing at least one of the plurality of decrypted data packets.

6. A method of encrypting data packets to be streamed, comprising the steps of:
   receiving at least one data packet;
   utilising a key to encrypt the at least one data packet; and
   generating an identifier that identifies the key and the data packet;
wherein the identifier is utilisable to associate the key and the data packet.

7. A method in accordance with Claim 6, wherein the key is saved in a first location.

8. A method in accordance with Claim 6 or 7, wherein the encrypted data packet is stored in a second location.

9. A computer program comprising at least one instruction for controlling a computer system to implement a method in accordance with any one of Claims 1 to 8.

10. A computer readable medium providing a computer program in accordance with Claim 9.

11. A data signal transmitted by an electronic system implementing a method in accordance with any one of Claims 1 to 8.

12. A data signal transmitted by a computer system executing a computer program in accordance with Claim 9.

13. A system for encrypting one or more data packets comprising:

   a module arranged to receive one or more data packets;

   a module arranged to utilise a key to encrypt one or more data packets; and

   a module arranged to generate an identifier that identifies the key and the data packet;

   wherein the identifier is utilisable to associate the key and the data packet.
14. A system for decrypting a plurality of data packets comprising:
   a receiving module arranged to receive a data packet from a sender;
   a module arranged to utilise an identifier to identify the data packet; and
   a module arranged to retrieve a key associated with the identifier to decrypt the encrypted data into decrypted data.

15. A method of processing data in a data stream comprising the steps of:
   associating a plurality of decryption keys with a respective plurality of portions of the data stream;
   whereby each decryption key is arranged to enable decryption of the associated portion of data of the data stream.

16. A system for processing data in a data stream comprising:
   a module arranged to associate a plurality of decryption keys with a respective plurality of portions of the data stream; and
   a module arranged to decrypt a portion of data of the data stream, whereby each decryption key is arranged to enable decryption of the associated portion of data of the data stream.
Figure 2

Encrypt Object (210)

Recipient (206)

Key (214)

Authentication (212)

Server

Request to encrypt (202)

Sender (204)

Email, virtual server, disk, storage devices
START

Encryption module receives data stream

Divide stream into individual blocks

Obtain key for each block

Encrypt each block

Establish connection with reception

Transmit each block to recipient

Additional authentication

Obtain a key from secure module

Decrypt block

Authenticate Recipient

YES

NO

END

Figure 4
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

H04L 9/00 (OCT 2005)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Google, Google Scholar, Google Patents, www.freepatentsonline.com, EPOQUE: decrypting, encrypting, data, packet, frame, identifier, key, display and other similar terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Documents are listed in the continuation of Box C</td>
<td></td>
</tr>
</tbody>
</table>

[X] Further documents are listed in the continuation of Box C [X] See patent family annex

* Special categories of cited documents:
  'A' document defining the general state of the art which is not considered to be of particular relevance
  'E' earlier application or patent but published on or after the international filing date
  'L' document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  'O' document referring to an oral disclosure, use, exhibition or other means
  'P' document published prior to the international filing date but later than the priority date claimed
  'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
  'X' document of particular relevance, the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
  'Y' document of particular relevance, the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art or the same patent family

Date of the actual completion of the international search
01 August 2012

Date of mailing of the international search report
02 August 2012

Name and mailing address of the ISA/AU
AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
Email address: pct@ipaaustralia.gov.au
Facsimile No.: +61 2 6283 7999

Authorized officer
Andrew Munro
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No. 0399359620

Form PCT/ISA/210 (fifth sheet) (July 2009)
<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 78143 16 B1 (HUGHES et al) 12 October 2010 see, for example; claims 1-2 and 6, figure 6, column 2 lines 41-45, column 5 lines 35-49</td>
<td>1-2, 5-16</td>
</tr>
<tr>
<td>X</td>
<td>US 7243202 B2 (THOMAS) 10 July 2007 column 1 lines 13-26 and claim 1</td>
<td>1-6, 8-16</td>
</tr>
<tr>
<td>A</td>
<td>US 2011/0145573 A1 (NAKAI et al) 16 June 2011 see, for example, the abstract and claim 1</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>US 7499545 B1 (BAGSHAW) 03 March 2009 see, for example, the abstract and claims 1, 4 and 13</td>
<td></td>
</tr>
</tbody>
</table>
This Annex lists known patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Publication Number</td>
<td>Publication Date</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>US 7499545 B1</td>
<td>03 Mar 2009</td>
</tr>
</tbody>
</table>

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.

Form PCT/ISA/210 (Family Annex)(July 2009)