

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Invention Title: Scalable Authentication Process and Apparatus

The invention is described in the following statement:

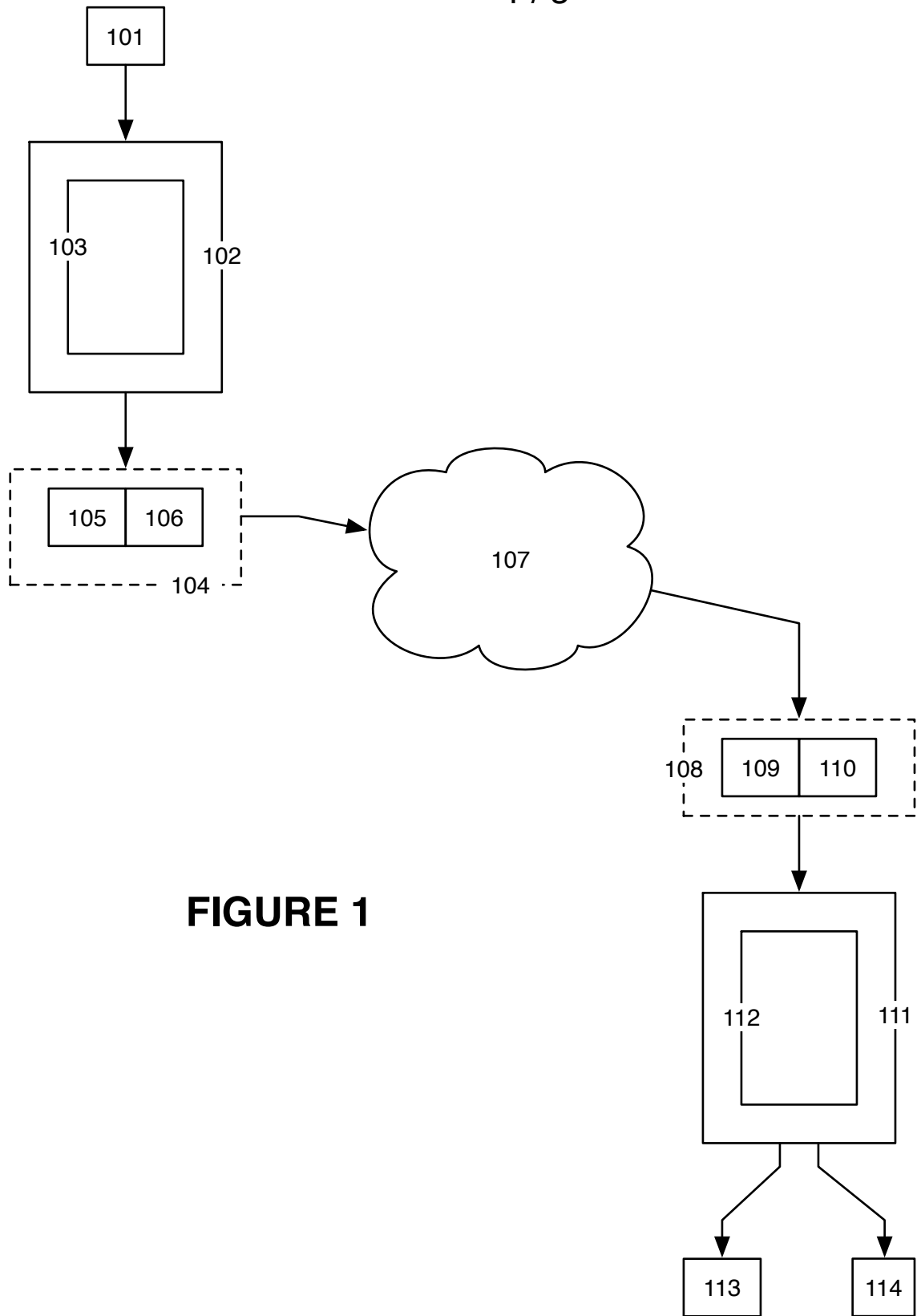


FIGURE 1

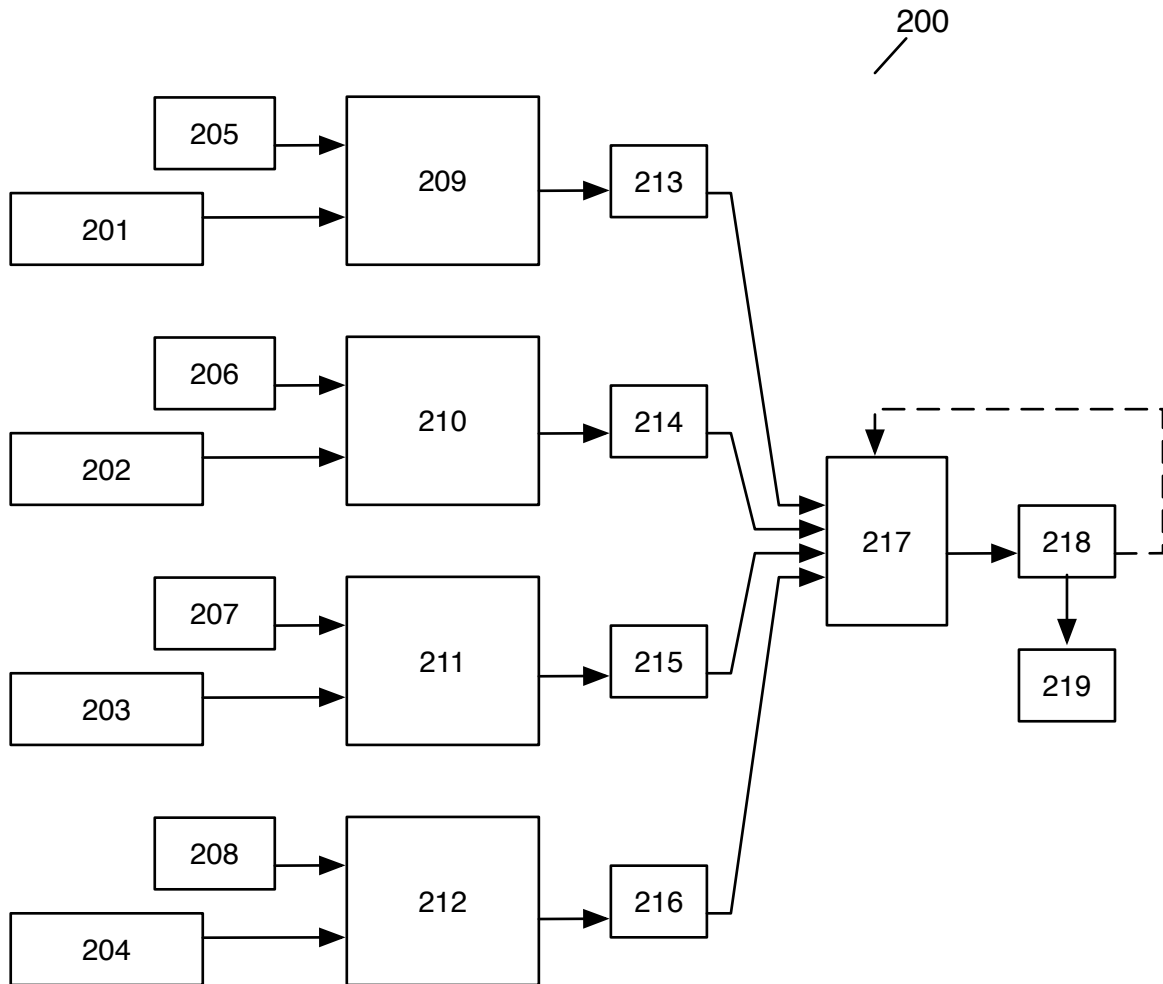


FIGURE 2

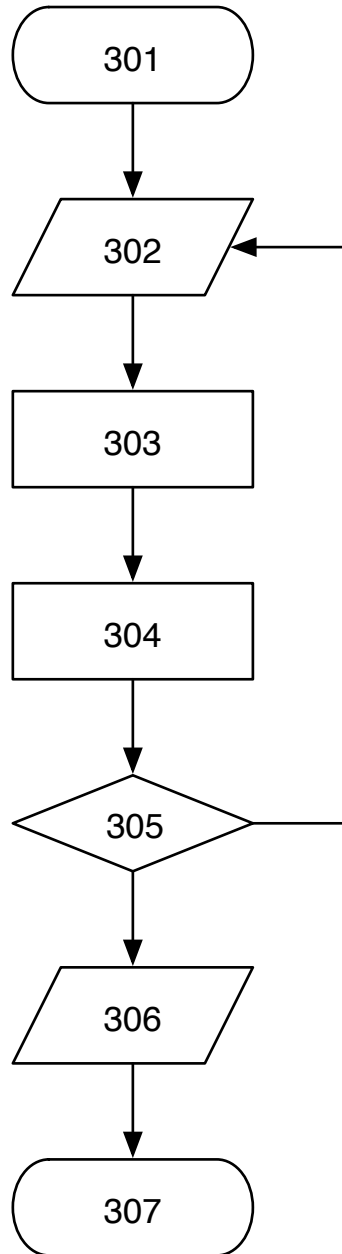


FIGURE 3

Title

Scalable authentication process and apparatus

5 Field of the invention

The present invention relates to counter-assisted data authentication.

Background of the invention

10 Data authentication is a cryptographic process that accepts an arbitrarily long stream of data and a key known to both parties and produces an output that must uniquely match the supplied data and key and must resist all possible attacks up to the required security level.

The apparatus 100 shown in figure 1 illustrates known processes of authenticated
15 transmission and reception of data. Data 101 is input to a hash function 103 that is executing on a computer 102. The hash function 103 generates a hash value 106. The output from the computer 102 is data 104 which includes the hash value 106 and a copy 105 of the data 101.

20 The data 104 is transmitted over a public network 107, eventually resulting in the receipt of data 108 at a receiving computer 111. The data 108 includes a component 109 and a component 110. If the data 104 has been corrupted or forged during transmission over the public network 107, then there is a high probability that the hash value of the component 109 will not be the same as the value of the component 110.

25 At a receiving computer 111, a hash function 112 calculates a hash value (which is not illustrated in the drawing) of the received component 109. If that calculated hash value of component 109 is the same as the received component 110, then the computer generates an output 113 that that has a high probability of being the same as the original input data
30 101. If the calculated hash value of component 109 is not the same as the received component 110, then the computer generates an indicator 114 to show that there is a high probability that data 104 has been either garbled or forged in transmission.

Transmission of data takes place over networks that have limitations on packet size and

authentication takes place using microchips that have limited data storage. Accordingly, long data streams must be divided into smaller blocks for encryption and authentication. The authentication can use collision-resistant one-way hash functions, MAC (message authentication code) processes combined with a block or stream cipher, or stream cipher processes that provide authenticated encryption with infinite error propagation.

Summary of the invention

In contrast, the present invention provides an authentication process, which process receives as input a key and a data stream, the authentication process comprising:

dividing the data stream into a set of data blocks, each data block being of the same size;

for each data block, performing an authentication process in an authenticator, each of which authenticators takes as separate inputs:

the block of data; and

a counter value that is related to the position of that data block in the data stream, and

producing an output from the authentication process by adding the outputs of the authenticators.

It is preferred that the data stream is padded by the authentication process so that message may be divided into equal sized data blocks.

According to a preferred form of the invention, at least two of the authentication processes are performed in parallel.

According to another preferred form of the invention, at least two of the authentication processes are performed in series.

It is preferred that each authenticator implements a function that is chosen from the group consisting of:

a collision-resistant one-way hash function;

a MAC (message authentication code) process combined with a block cipher;

a MAC (message authentication code) process combined with a stream cipher; and

a stream cipher process that provides authenticated encryption with infinite error propagation.

It is preferred that the adding is chosen from the group consisting of:

- 5 a binary XOR operation; and
- a word-based carry add operation.

The present invention also provides apparatus that implements such processes.

- 10 It will be seen that the present invention offers a scalable (parallelizable) method of data authentication by linearly combining outputs of counter-assisted authenticators processing small blocks of data. Each authenticator receives as its inputs a data block and a counter value corresponding to the position of that data block in the stream. In this way any number of authenticators can be executed in parallel while being fully compatible with
- 15 other implementations of the same process executing a larger or a smaller number of simultaneous authenticators. Security of this method relies entirely on the security of the underlying authenticators. A cryptographer skilled in the art can also combine our counter mode authentication method described herein with a sufficiently secure counter mode encryption process. Since our method does not require large additional storage or use of
- 20 complex multiplication operations, it is very efficient in hardware allowing larger microchips to provide data authentication at higher speeds with minimal overhead for both large and small implementations.

25 **Brief description of the drawings**

In order that the invention may be more readily understood, preferred embodiments of it are described with reference to the drawings in which:

- figure 1 is a block schematic drawing generally illustrating the known technique of data authentication;
- 30 figure 2 is a block schematic drawing illustrating apparatus according to preferred embodiments of the present invention; and
- figure 3 illustrates processes of operation of the apparatus of figure 2.

Description of preferred embodiments of the invention

Figure 2 illustrates a preferred embodiment of apparatus 200 according to the present invention. A consecutive stream of data (which is not illustrated in the drawings) is divided into a series of data blocks, the blocks all being of the same size. Figure 2 shows the data stream as being divided into four data blocks 201, 202, 203 and 204, all of the same size. The four data blocks and four counter values 205, 206, 207 and 208 are inputs into authenticators 209, 210, 211 and 212. (The generation of counter values is described below.) Resulting authenticator outputs 213, 214, 215 and 216 are added together by adder 217 to accumulator state 218. Output of the adder 217 is saved as updated accumulator state 218 and is returned as the process output at the end of the process. Preferred forms of counter to generate the counter values 105, 106, 107, and 108 are such as are disclosed in our co-pending International Patent Applications PCT/AU2006/000527 entitled *Process of and Apparatus for Counting*; ; and PCT/AU2006/000528 entitled *Process of and Apparatus for Counting*, the contents of each of which is incorporated herein by reference.

Preferred forms of the adder 217 are chosen from the group consisting of:

- a binary XOR; and
- a word-based carry adder.

Preferred forms of the authenticators 209, 210, 211 and 212 are such as are described in our International Patent Application PCT/AU2006/000557, *Process of and Apparatus for Hashing*, the contents of which is incorporated herein by reference.

Another preferred form of the authenticators 209, 210, 211 and 212 uses counter-assisted authenticated encryption according to our VEST cipher as its authentication processes. That cipher is described in detail in our paper *Authenticated Encryption Mode of VEST Ciphers* by Sean O'Neil and Benjamin Gittins, available in the Cryptology ePrint Archive: Report 2005/414 that is published at <http://eprint.iacr.org/2005/414> a copy of which is Annex A to this specification.

Figure 3 illustrates a preferred process 300 of operation of the embodiments of figure 2. The process 300 initializes internal states of all its authenticators and the counters 205, 206, 207 and 208 are set to their initial values corresponding to indexes of the first four

blocks of data on initialization step 301. The process receives the four subsequent data blocks 201, 202, 203 and 204 and as its inputs on step 302. On step 303 the four data blocks are hashed by counter-assisted authenticators 209, 210, 211 and 212, and all four counter values are simultaneously incremented to their values corresponding to indexes of the next four data blocks. On step 304 the outputs of the four authenticators are added by the adder 217 to accumulator state 218. On decision step 305 the process evaluates the amount of processed data and if more data is available, the process returns to the step 302, otherwise the process returns accumulator state 218 as its output on step 306 and terminates on step 307.

5
10

It will be appreciated that the authentication of a multi-block message may be achieved according to various embodiments of the present invention by performing the authentication of all of the blocks in parallel, by performing the authentication of all of the blocks sequentially, or by performing a mixture of parallel and sequential authentication.

15

While the present invention has been described with reference to a few specific embodiments, the description is illustrative of the invention and is not to be construed as limiting the invention. Various modifications may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims.

20

“Comprises/comprising” when used in this specification is taken to specify the presence of stated features, integers, steps or components but does not preclude the presence or addition of one or more other features, integers, steps, components or groups thereof.

25

The claims defining the invention are as follows:

1. An authentication process, which process receives as input a key and a data stream,
the authentication process comprising:
5 dividing the data stream into a set of data blocks, each data block being of
 the same size;
 for each data block, performing an authentication sub-process in an
 authenticator, each of which authenticators takes as separate inputs:
 the block of data; and
 a counter value that is related to the position of that data block in
10 the data stream, and
 producing an output from the authentication process by adding the outputs
 of the authenticators.

2. An authentication process as claimed in claim 1, in which each of the
15 authenticators additionally takes as separate inputs at least one of:
 a key; and
 an initialization vector (IV).

3. An authentication process as claimed in claim 1 or claim 2, in which the data
20 stream is padded by the authentication process so that message may be divided into
 equal sized data blocks.

43. An authentication process as claimed in any one of the preceding claims, in which
 at least two of the authentication sub-processes are performed in parallel.
25

5. An authentication process as claimed in any one of the preceding claims, in which
 at least two of the authentication sub-processes are performed in series.

6. An authentication process as claimed in any one of the preceding claims, in which
30 each authenticator implements a function that is chosen from the group consisting
 of:
 a collision-resistant one-way hash function;
 a MAC (message authentication code) process combined with a block
 cipher;

a MAC (message authentication code) process combined with a stream cipher; and
a stream cipher process that provides authenticated encryption with infinite error propagation.

5

7. An authentication process as claimed in any one of the preceding claims, in which the adding is chosen from the group consisting of:

a binary XOR operation; and
a word-based carry add operation.

10

8. Apparatus that is adapted to implement an authentication process as claimed in any one of the preceding claims.

Synaptic Laboratories Limited

15

Dated: 27 July 2006

Abstract

An authentication process (300) divides a data stream into blocks (201, 202, 203, 204) of equal size. An authenticator (209, 210, 211, 212) each receives a data block (201, 202, 203, 204) and a counter output (205, 206, 207, 208). Each authenticator (209, 210, 211, 212) performs an authentication sub-process. The outputs of the authenticators (209, 210, 211, 212) are added by an accumulator (217).