(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 224 533 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
02.01.2004 Bulletin 2004/01

(21) Application number: **00968189.1**

(22) Date of filing: **18.10.2000**

(51) Int Cl.[7]: **G06F 7/72**

(86) International application number:
PCT/IE2000/000132

(87) International publication number:
WO 2001/029652 (26.04.2001 Gazette 2001/17)

(54) **A CRYPTOGRAPHIC ACCELERATOR**

KRYPTOGRAPHISCHER VERSCHNELLER

ACCELERATEUR CRYTOGRAPHIQUE

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **20.10.1999 IE 990878**

(43) Date of publication of application:
24.07.2002 Bulletin 2002/30

(73) Proprietor: **AEP Systems Limited**
Bray, County Wicklow (IE)

(72) Inventors:
- **FAIRCLOUGH, Christopher**
  Greystones, County Wicklow (IE)
- **FLANAGAN, Francis**
  County Dublin (IE)

(74) Representative: **Weldon, Michael James et al**
c/o John A. O'Brien & Associates,
Third Floor,
Duncairn House,
14 Carysfort Avenue
Blackrock, Co. Dublin (IE)

(56) References cited:
EP-A- 0 502 782
EP-A- 0 525 968
WO-A-99/14881
WO-A-99/39475
US-A- 5 289 397
US-A- 5 923 893

- **NAOFUMI TAKAGI: "A RADIX-4 MODULAR MULTIPLICATION HARDWARE ALGORITHM FOR MODULAR EXPONENTIATION" IEEE TRANSACTIONS ON COMPUTERS,US,IEEE INC. NEW YORK, vol. 41, no. 8, 1 August 1992 (1992-08-01), pages 949-956, XP000298588 ISSN: 0018-9340**
- **"IBM SYSTEM DIGITAL SIGNATURE DATA STRUCTURE FORMAT" IBM TECHNICAL DISCLOSURE BULLETIN,US,IBM CORP. NEW YORK, vol. 36, no. 5, 1 May 1993 (1993-05-01), pages 343-346, XP000409015 ISSN: 0018-8689**

EP 1 224 533 B1

Description

INTRODUCTION

Field of the Invention

[0001] The invention relates to a cryptographic accelerator.

Prior Art Discussion

[0002] In any electronic exchange of information between two or more participants, cryptography is intended to provide some or all of the following assurances.

- Confidentiality
  No one except the intended participant(s) will have access to the information exchanged
- Authentication
  Each participant is confident of the identities of the other participant(s)
- Integrity
  The information exchanged between the participants will have nothing added or removed without the participants being aware of the adulteration
- Non-Repudiation
  A sender of information cannot deny having sent the information, and a recipient cannot deny its reception.

[0003] These assurances are essential to the growth of secure electronic communications. The biggest problem associated with conventional (symmetric/single key) cryptography relates to the distribution of the secret keys used to encrypt and decrypt data in secure communication sessions. Modem public key encryption, which uses public/private key pairs, overcomes this problem. However, public key encryption carries a very large computational overhead in comparison to that associated with conventional encryption. As a way of limiting this overhead, many cryptographic protocols only use public key encryption as a mechanism to allow participants setting up a secure communication session to exchange secret keys. The exchanged keys are then used for conventional encryption to encrypt the bulk of data to be transmitted in the session.

[0004] Modem PC systems, with suitable software, are capable of implementing both conventional and public key encryption mechanisms in order to complete secure electronic transactions (for example Web shopping or Internet banking). The computing overheads and physical security required are not beyond the resources of a typical end-user PC provided that it does not need to carry out a large number of such transactions within a short period of time. However, this is not the case for the commercial server systems with which these transactions are conducted. E-commerce server systems are naturally expected to be able to conduct large numbers of transactions within short periods of time, and must be able to guarantee a high degree of physical security for this activity.

[0005] One of the emerging protocols used for electronic commercial transactions is SET (Secure Electronic Transactions). Depending on the nature of the transactions involved, a single electronic 'purchase' can involve a many as fourteen separate public key operations on different systems in up to four separate organisations. Clearly, since the computational requirements of public key encryption are high, this activity becomes a bottleneck orders of magnitude over and above the normal overheads of the administration and logistics of computer based commercial order-processing systems.

[0006] WO9939475 describes a cryptographic system and method for encrypting and decrypting data using public key cryptography comprising a host interface, a processor, memory units, and a cryptographic co-processor.

[0007] It is therefore an object of the invention to provide a cryptographic accelerator which provides the level of cryptographic computation required and has a high throughput.

SUMMARY OF THE INVENTION

[0008] According to the invention, there is provided a cryptographic accelerator system comprising a host interface comprising means for interfacing with a host system having applications requesting cryptographic operations, means for performing exponentiation, means in the host interface for routing request responses to the host system, a plurality of logical units, including an exponentiation sub-system, and a CPU connected between the host interface and the

logical units and comprising means for managing operation of the logical units,
characterised in that,

the exponentiation subsystem comprises individual modular exponentiators of a predetermined bit size, which can be operated alone, and means for dynamically forming a group of modular exponentiators chained together for providing exponentiations up to a multiple of said predetermined bit size.

[0009] Such an arrangement addresses the problems of the prior art and provides an improved cryptographic system with the required levels of computation of and a high throughput.

[0010] In one embodiment, the exponentiation subsystem comprises an ASIC.

[0011] In one embodiment, the exponentiation subsystem comprises means for chaining modular exponentiators within a group, wherein all chains within a group are of the same length.

[0012] In one embodiment, the exponentiation subsystem comprises means for executing exponentiation based on the Montgomery algorithm.

[0013] In one embodiment, each modular exponentiator has a size of 544 bits.

[0014] In one embodiment, the exponentiation sub-system comprises a scheduler, an exponentiator input buffer, and an exponentiator output buffer, and the scheduler comprises means for routing scheduling instructions to the exponentiators via the input buffer.

[0015] In one embodiment, the instructions include a status field for insertion of an error in the output buffer if a result should be discarded.

[0016] In one embodiment, the instructions include a control field with a group mode instruction for a chaining configuration.

[0017] In one embodiment, the control field instruction is associated with a particular group.

[0018] In one embodiment, the instructions include a block identifier field for insertion in the output buffer of an identifier of the block which generated the result.

[0019] In one embodiment, the instructions include a group identifier field for insertion in the output buffer of an identifier of the group which generated the result.

[0020] In one embodiment, the exponentiation sub-system comprises means for accessing control registers, including a register for an instruction causing the scheduler to commence initialisation of groups with exclusion of certain error-prone groups.

[0021] In one embodiment, a control register stores linear feedback shift register contents.

[0022] In one embodiment, the scheduler and the input buffer comprises means for transferring dummy data to exponentiators in the absence of real data.

[0023] In one embodiment, the host interface comprises a daemon and a plurality of APIs for a host system, and said daemon comprises means for managing request queues on a per-logical unit basis.

[0024] In one embodiment, the CPU comprises a parser comprising means for breaking each request into commands, for automatically determining a required response data space, and for allocating said space.

[0025] In one embodiment, each parser is associated with a particular logical unit and comprises means for breaking the commands into strings of a desired format and size for the associated logical unit.

[0026] In one embodiment, the CPU comprises a plurality of micro sequencers, each comprising means for either routing parsed command strings to the destination logical unit or for performing the requested operation itself.

[0027] In one embodiment, the logical units comprise a block cipher unit comprising means for implementing bulk and/or symmetric cipher operations.

[0028] In one embodiment, the logical units comprise a random number generator comprising means for generating a random number bit stream, and for performing a statistical analysis to ensure that the bits are random.

[0029] In one embodiment, the CPU comprises means for using the random number bit stream to generate prime numbers and for storing the prime numbers in configurable pools.

[0030] In one embodiment, the accelerator further comprises a bus for communication of the CPU with the logical units.

DETAILED DESCRIPTION OF THE INVENTION

Brief Description of the Drawings

[0031] The invention will be more clearly understood from the following description of some embodiments thereof, given by way of example only with reference to the accompanying drawings in which:-

Fig. 1 is a block diagram illustrating a cryptographic accelerator of the invention;

Fig. 2 is a diagram illustrating a modular exponentiation subsystem of the accelerator at a high level; and

Fig. 3 is a diagram showing the structure of a group of the subsystem of Fig. 2.

Description of the Embodiments

[0032] Referring to Fig. 1, a cryptographic accelerator 1 of the invention is illustrated. The accelerator 1 comprises a host interface 2 for interfacing with a host server such as a server performing on-line secure transactions. A CPU 3 handles host interfacing, device drivers, and authentication. It also implements some cryptography algorithms. An access control block 4 provides tamper resistance and includes components ranging from physical tamper-detection devices such as microswitches to intelligent access control functions. An internal bus 5 supports DMA transfer between the logical units within the accelerator 1. A block cipher function 6 is a PLD to implement encryption and decryption. It is particularly suitable for encryption of large blocks of data. The accelerator 1 also comprises a modular exponentiation subsystem 7, a random number generator 8, and a key storage function 9.

[0033] In more detail, the host interface 2 comprises a daemon and APIs 15 executing on a host server and also a PCI interface 16 comprising hardware and software within an accelerator circuit physically separate from the host system, shown by interrupted lines.

[0034] The host server has multiple applications. Each application is multi-threaded and interfaces to an instance of a library in the server, the library being associated with the accelerator 1 via sockets which are managed by a single daemon.

[0035] The applications route requests to the accelerator 1, and each request is either:

(a) a synchronous request in which the application waits for a response, or

(b) an asynchronous request in which the application does not wait and must be reactivated to receive the response.

[0036] The daemon manages the request via the sockets and a device driver connected to the PCI interface 16. The daemon is programmed with attributes of the logical units 6 - 9 of the accelerator 1 and it manages the requests in queues for the logical units.

[0037] The CPU 3 comprises:-

- six micro engines (micro sequencers),
- two high speed memory interfaces,
- an Advanced RISC Microprocessor (ARM) with real time multitasking capability,

[0038] The ARM has a message parser for each logical unit, and each parser parses signals for a logical unit, as set by the daemon. Each parser breaks requests into commands, determines what data space will be required for the resulting response, and reserves the appropriate space in the CPU 3. Each parser also breaks the commands into strings of a desired size and format for the associated logical unit.

[0039] Each micro engine of the CPU 3 is independently programmable and routes commands from the queues generated by the parsers to the relevant logical units. Also, each micro engine may, instead of routing the commands to the destination logical unit, actually perform the requested operation itself. An example is a hashing operation. The micro engines also receive responses (via the bus 5) from the logical units and route them to the host server applications via the relevant sockets.

[0040] The block cipher function 6 comprises firmware for implementing bulk/symmetric cipher, for example those specified in the DES (Data Encryption Standard).

[0041] The modular exponentiation sub-system (logical unit) 7 performs exponentiation, described in detail below.

[0042] The random number generator 8 is programmed to generate a random bit stream and to perform a statistical analysis to ensure that the bits are indeed random. The random bit stream is routed to the CPU 3 via the bus 5, and the CPU 3 stores the bits in memory. The CPU 3 then uses the stored random bits to determine prime numbers. It stores the prime numbers in different, configurable pools for use in performing cryptography operations.

[0043] Referring to Fig. 2, the exponentiator sub-system 7 comprises an ASIC comprising ten groups 20 each comprising four 544 bit exponentiator blocks 30. The sub-system 7 also comprises an input buffer 21, an output buffer 22, an IX bus interface 23, a SRAM bus interface 24, a PLL 25, and a scheduler 26. The SRAM interface allows access to off-chip SRAM.

[0044] The blocks 30 may be operated alone or dynamically chained together up to the size of a group providing for 2174 bit exponentiations. This is illustrated in Fig. 3. The primary clock is generated by the on-chip PLL 25. Each 544 bit exponentiator 30 is a unit capable of completing each Montgomery multiply of a number up to 542 bits in 1089 clock cycles. As the units are configured in groups of four, each group provides for exponentiations of up to 2174 bits.

[0045] The 544 bit block with maximum 4 block chain size has been chosen to provide near to optimal utilisation of

the silicon resources for most common key sizes. Each group can be configured as a number of chains as shown in Table 1 below. However all chains within a group are configured to the same size, and for optimal performance all exponents within a group are of approximately the same number of bits as the sub-system 7 will asynchronously terminate once all exponentiations have completed.

Table 1.

Standard Modulus Sizes vs. Chain Length		
Modulus Size	Blocks Per Chain	Chains per Group
256	1	4
512	1	4
768	2	2
1024	2	2
1536	3	1
2048	4	1

Scheduling

[0046] The scheduler 26 controls the allocation of work to each of the groups 20. Data is transferred to the input buffer 21 complete with all of the information necessary to control the group 20 in performing the exponentiation. The scheduler 26 allocates the work from the input buffer to the first free group. As the software has no control over which group will carry out the exponentiations or how long it will take to process each block, data is transferred with a block identifier. The block identifier is returned in the output buffer with the exponentiation result. Additionally a group identifier is returned allowing the group 20 responsible for a particular result to be identified.

[0047] The sub-system 7 only transfers data to the output buffer 22 when a valid input buffer is available. Dummy data and keys are used in the absence of valid data to process. This mechanism is intended to keep the sub-system 7 busy at all times processing a range of data and therefore increases the difficulty of any attempts at power or tempest-type analysis.

IX Bus Interface 23

[0048] This interfaces with the IX bus 5, which is an open bus defined by Level-1 communications for direct interfacing of communication chips in bridges and routers. It is a FIFO based bus driven at the processor end by micro-code on the IXP-1200 and an entire family of networking chips including GigaBit Ethernet.

Input and Output Buffers

[0049] The input buffer 21 is arranged as follows.

Field	Exponentiator Unit	MSB	LSB
DATA	Exponentiator 0	543	0
	Exponentiator 1	1087	544
	Exponentiator 2	1631	1088
	Exponentiator 3	2175	1632
EXPONENT-1	Exponentiator 0	2719	2176
	Exponentiator 1	3263	2720
	Exponentiator 2	3807	3264
	Exponentiator 3	4351	3808
R2 MOD M	Exponentiator 0	4895	4352
	Exponentiator 1	5439	4896

(continued)

Field	Exponentiator Unit	MSB	LSB
	Exponentiator 2	5983	5440
	Exponentiator 3	6527	5984
(M+1)/2	Exponentiator 0	7071	6528
	Exponentiator 1	7615	7072
	Exponentiator 2	8159	7616
	Exponentiator 3	8703	8160
GBCS		8767	8704
BPC		8831	8768

[0050] The output buffer 22 is arranged as follows:

Field	Exponentiator Unit	MSB	LSB
RESULT	Exponentiator 0	543	0
	Exponentiator 1	1087	544
	Exponentiator 2	1631	1088
	Exponentiator 3	2175	1632
GBCS		2239	2176
BPC		2303	2240

[0051] The data returned from the exponentiator is normally the correct result but may for some specific input data be the result + m and require a single subtract to normalise it to the correct range.

Group ID, Block identifier. Control and Status (GBCS)

[0052] This is a 64 bit field present in both the input and output buffers. These 64 bits are organised as follows:

Field	Bits
Status	7:0
Control	15:8
Block Identifier	31:16
Group Identifier	63:32

Status (7:0)
7:1 Reserved
0 BPC Downstream Status

[0053] In the input buffer 21 the BPC bit in the status field is ignored. In the output buffer 22 the BPC bit will be set if an error in the downstream transfer to the input buffer was detected and the result should therefore be discarded.

Control (15:8)
15:10 Reserved
9:8 Exponentiator Group Mode

[0054] The exponentiator group mode field in the input buffer 21 determines the group chaining configuration. In the output buffer 22 this field is reset.

Mode	Control(9:8)	Configuration
0	00	4x544

(continued)

Mode	Control(9:8)	Configuration
1	01	2x1088
2	10	Undefined
3	11	1X2176

Block Identifier (31:16)

[0055] A 16 bit sequence number set by software in the input buffer and set in the output buffer so that results in the output buffer may be associated with requests sent via the input buffer.

Group Identifier (63:32)

[0056] Group identifier, set in the output buffer 22 to indicate which exponentiator group 20 generated a particular result. A single bit of the 10 lower order bits will be set to uniquely identify the group. This field is ignored in the input buffer.

BPC (Block Parity Check)

[0057] This is a 64 bit block parity check used to check for data transfer errors. In the input buffer 21 this is set to the XOR of the input data and GBCS. Should an error be detected in the downstream transfer to the device a BPC Downstream Status error is indicated in the GBCS status field of the output buffer. In the output buffer the BPC is generated by the device. On the receive side the BPC may be calculated for the output buffer and compared with the generated BPC to detect transmission errors.

Arrangement of per Block Operands

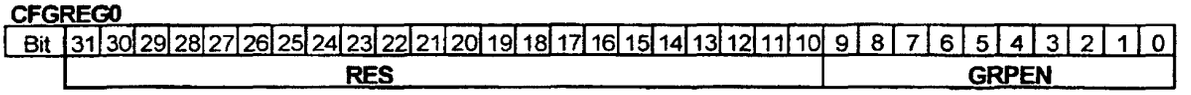
[0058] All per exponentiator data, keys, and modulus are arranged in the buffers starting at bit 0 of the exponentiator block in which the operand starts and running up contiguously. This means that for a group configured for 4 x 512 bit exponentiations bits 543:512 would be set to zero in the input buffer, whereas for 2x1024 bit exponentiations these bits are used in the middle of the data. For standard modulus sizes the arrangement is as follows:

256	512	768	1024	2048
255:0	511:0	767:0	1023:0	2047:0
799:544	1055:544	1855:1088	2111:1088	NONE
1343:1088	1599:1088	NONE	NONE	NONE
1887:1632	2143:1632	NONE	NONE	NONE

Configuration Registers

[0059] Four 32-bit configuration registers are accessible by the scheduler 26 via a configuration register interface. These are CFGREG0, CFGREG1, CFGREG2 and CFGREG3.

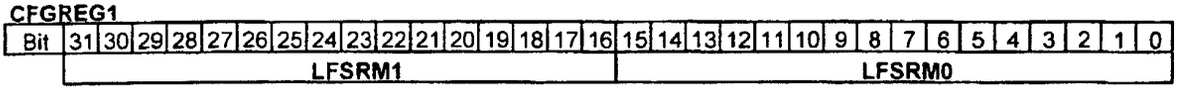
CFGREG0



Bits	Name	Description
31:10	RES	Reserved.
9:0	GRPEN	Exponentiator group enable/disable. 0: Disable. 1: Enable.

[0060] CFGREG0 is read/writable. All writes to CFGREG0 cause the scheduler 26 to go through its initialisation sequence at the end of which scheduling will commence with group 9 in decreasing order. The GRPEN field within CFGREG0 provides a mechanism to exclude particular groups from being scheduled. Each of the 10 bits in GRPEN allows its respective group to be enabled/disabled (1 = enabled, 0 = disabled). In the event that an exponentiation result is found to be in error, the exponentiator group responsible can be identified using the group identifier field within the GBCS quad-word in the output buffer. That group can then be excluded from scheduling by resetting the appropriate bit in the GRPEN field.

CFGREG1

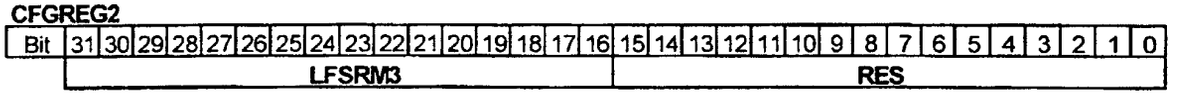


Bits	Name	Description
31:16	LFSRM1	Linear feedback shift register constant for Mode 1.
15:0	LFSRM0	Linear feedback shift register constant for Mode 0.

[0061] CFGREG1 holds the linear feedback shift register constants for mode 0 and mode 1. These must be initialised to appropriate values prior to starting the scheduler with a write to CGFREG0:

LFSRM0 0x0F6A
LFSRM 1 0x08BA

CFGREG2

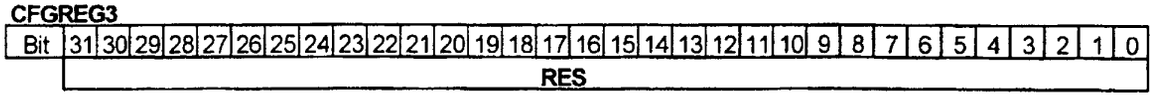


Bits	Name	Description
31:16	LFSRM3	Linear feedback shift register constant for Mode 3.

[0062] CFGREG2 holds the linear feedback shift register constant for mode 3. This must be initialised to an appropriate value prior to starting the scheduler with a write to CGFREG0:

LFSRM3 0x0148

CFGREG3



Bits	Name	Description

[0063] CFGREG3 is reserved for future use.

Initialisation

[0064] Initialisation of the sub-system 7 requires the following steps:

1. Linear Feed Shift Register Initialisation.
Write of 0x08BA0F6A to CFGREG1.
Write of 0x01480000 to CFGREG2.

2. Scheduler Initialisation.
In normal operation all ten exponentiator groups 20 are enabled by a write of 0x000003FF to CFGREG0. Should a group be known to be faulty it may be prevented from being scheduled by resetting the appropriate bit in CGFREG0.

3. Exponentiator group initialisation.
Each exponentiator group 20 must be initialised by executing an initialisation operation. The initialisation operation consists of a specific input data block and operation mode:

DATA = 0
EXPONENT-1 = 0
R2MODM = 0
M+1/2=0
MODE = 0

Assuming all ten exponentiator groups have been enabled the first ten operations must be initialisation operations. The first ten results contain post-reset data which is undefined and as such should be discarded. The block and group identifier fields of the GBCS may be used to verify that each exponentiator group has executed an initialisation operation. Known test data is cycled through each group to verify its operation prior to putting the device 1 into use at each power up.

[0065] It will be appreciated that the accelerator 1 provides for very fast operation in a simple and effective manner. The CPU 3 implements unusual cryptographic algorithms, and the block cipher 6 performs efficient symmetric encryption of large blocks of data. The subsystem 7 is extremely important to performance of the accelerator 1 as a whole. It provides a very high throughput per gate count because of use of small exponentiators. The interconnects in the block/chain/group structure allow selection of the size of multipliers with only serial data streams. The buffers operate effectively to group operations of the same modulus and similar exponent size into a group totalling up to 2048 modulus bits. The buffers also calculate the Montgomery residue of data, submit grouped data to the groups 20, convert the final Montgomery residue to a result, and submit dummy data to the groups 20 in the absence of sufficient real data.

[0066] A security feature is that of product verification and ownership. Prior to leaving the factory each cryptographic accelerator is sealed and digitally signed to verify its integrity. The micro-controller which monitors the tamper detection circuitry operates in transit to the customer.

[0067] When the product is delivered to the customer, as far as the firmware is concerned the box is the property of the manufacturer. The customer should connect his own computer to the serial line interface, touch his own Cryptoibutton to the blue dot receptor and obtain the box's verification message. This includes the serial number of the box, the mode of initial configuration, and the signature. This message should be verified using known public keys of the manufacturer. The customer should then send a digitally signed transfer of ownership request to the manufacturer who will sign the request and return it to the customer. This signed message is input to the unit and the transfer of ownership is complete.

[0068] After the transfer of ownership the accelerator has become the exclusive "property" of the holder of the cryptoibutton used in the transfer. The owner of this button may authorise other users at various access levels as required.

[0069] This verification and transfer of ownership protocol may be repeated between departments, crypto-officers etc. or just to replace old keys with new ones as needed. Once ownership has been transferred the unit will cease to recognise any configuration or logon requests signed by its previous owner and the transfer of ownership is recorded in a permanent audit trail.

[0070] As the transfer of ownership is permanent and irrevocable the loss of private keys with which to re-configure the unit is a substantial problem which will necessitate return to factory and complete re-initialisation with complete loss of audit trail. The manufacturer therefore signs all units prior to shipment with two messages generated using

separate private keys in separate secure locations. A message generated using either of these keys may be used to transfer ownership of the unit. It is suggested that customers adopt a similar approach.

[0071] The invention is not limited to the embodiments described, but may be varied in construction and detail. For example, the host interface may comprise several cascaded SCSI devices instead of a PCI interface.

Claims

1. A cryptographic accelerator system comprising a host interface (2) comprising means for interfacing with a host system having applications requesting cryptographic operations, means for performing exponentiation, means in the host interface for routing request responses to the host system, a plurality of logical units (6-9), including an exponentiation sub-system (7), and a CPU (5) connected between the host interface (2) and the logical units (6-9) and comprising means for managing operation of the logical units.
   characterised in that,
   the exponentiation subsystem comprises individual modular exponentiators (30) of a predetermined bit size, which can be operated alone, and means for dynamically forming a group (20) of modular exponentiators chained together for providing exponentiations up to a multiple of said predetermined bit size.

2. A cryptographic accelerator system as claimed in claim 1, wherein the exponentiation subsystem (7) comprises an ASIC.

3. A cryptographic accelerator system as claimed in claim 1, wherein the exponentiation subsystem (7) comprises means for chaining modular exponentiators (30) within a group (20), wherein all chains within a group are of the same length.

4. A cryptographic accelerator system as claimed in any preceding claim, wherein the exponentiation subsystem (7) comprises means for executing exponentiation based on the Montgomery algorithm.

5. A cryptographic accelerator system as claimed in claims 3 or 4, wherein each modular exponentiator has a size of 544 bits.

6. A cryptographic accelerator system as claimed in any preceding claim, wherein the exponentiation sub-system comprises a scheduler (26), an exponentiator input buffer (21), and an exponentiator output buffer (22), and the scheduler comprises means for routing scheduling instructions to the exponentiators via the input buffer (21).

7. A cryptographic accelerator system as claimed in claim 6, wherein the instructions include a status field for insertion of an error in the output buffer (22) if a result should be discarded.

8. A cryptographic accelerator system as claimed in claim 6 or 7, wherein the instructions include a control field with a group mode instruction for a chaining configuration.

9. A cryptographic accelerator system as claimed in claim 8, wherein the control field instruction is associated with a particular group.

10. A cryptographic accelerator system as claimed in any of claims 6 to 9, wherein the instructions include a block identifier field for insertion in the output buffer of an identifier of the block which generated the result.

11. A cryptographic accelerator system as claimed in any of claims 6 to 10, wherein the instructions include a group identifier field for insertion in the output buffer of an identifier of the group which generated the result.

12. A cryptographic accelerator system as claimed in any preceding claim, wherein the exponentiation sub-system (7) comprises means for accessing control registers, including a register for an instruction causing the scheduler (26) to commence initialisation of groups with exclusion of certain error-prone groups.

13. A cryptographic accelerator system as claimed in claim 12, wherein a control register stores linear feedback shift register contents.

14. A cryptographic accelerator system as claimed in any of claims 6 to 13, wherein the scheduler (26) and the input

buffer (21) comprises means for transferring dummy data to exponentiators (30) in the absence of real data.

15. A cryptographic accelerator system as claimed in any preceding claim, wherein the host interface comprises a daemon and a plurality of APIs (15) for a host system, and said daemon comprises means for managing request queues on a per-logical unit (6-9) basis.

16. A cryptographic accelerator system as claimed in claim 15, wherein the CPU (3) comprises a parser comprising means for breaking each request into commands, for automatically determining a required response data space, and for allocating said space.

17. A cryptographic accelerator system as claimed in claim 16, wherein each parser is associated with a particular logical unit (6-9) and comprises means for breaking the commands into strings of a desired format and size for the associated logical unit.

18. A cryptographic accelerator system as claimed in any preceding claim, wherein the CPU (3) comprises a plurality of micro sequencers, each comprising means for either routing parsed command strings to the destination logical unit or for performing the requested operation itself.

19. A cryptographic accelerator system as claimed in any preceding claim, wherein the logical units comprise a block cipher unit (6) comprising means for implementing bulk and/or symmetric cipher operations.

20. A cryptographic accelerator system as claimed in any preceding claim, wherein the logical units comprise a random number generator (8) comprising means for generating a random number bit stream, and for performing a statistical analysis to ensure that the bits are random.

21. A cryptographic accelerator system as claimed in claim 20, wherein the CPU (3) comprises means for using the random number bit stream to generate prime numbers and for storing the prime numbers in configurable pools.

22. A cryptographic accelerator system as claimed in any preceding claim, wherein the accelerator further comprises a bus (5) for communication of the CPU (3) with the logical units (6-9).

Patentansprüche

1. Verschlüsselungsbeschleunigersystem, mit einer Hauptschnittstelle (2) mit einer Einrichtung zur Verbindung mit einem Hauptrechner mit Verschlüsselungsvorgänge anfordernden Anwendungen, einer Einrichtung zum Potenzieren, einer Einrichtung in der Hauptschnittstelle zum Leiten von Antworten auf Anforderungen an den Hauptrechner, mehreren Logikeinheiten (6-9) einschließlich einem Potenzierungs-Subsystem (7), und einer zwischen die Hauptschnittstelle (2) und die Logikeinheiten (6-9) geschalteten CPU (5) sowie einer Einrichtung zum Verwalten von Vorgängen der Logikeinheiten,
dadurch gekennzeichnet,
dass das Potenzierungs-Subsystem einzelne modulare Potenzierer (30) einer vorgegebenen Bitgröße, die alleine betätigt werden können, und eine Einrichtung zum dynamischen Bilden einer Gruppe (20) von verketteten modularen Potenzierern zum Vorsehen von Potenzierungen bis zu einem Vielfachen der vorgegebenen Bitgröße aufweist.

2. Verschlüsselungsbeschleunigersystem nach Anspruch 1, bei welchem das Potenzierungs-Subsystem (7) eine ASIC aufweist.

3. Verschlüsselungsbeschleunigersystem nach Anspruch 1, bei welchem das Potenzierungs-Subsystem (7) eine Einrichtung zum Verketten modularer Potenzierer (30) in einer Gruppe (20) aufweist, wobei alle Ketten in einer Gruppe von der gleichen Länge sind.

4. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem das Potenzierungs-Subsystem (7) eine Einrichtung zum Ausführen einer Potenzierung basierend auf dem Montgomery-Algorithmus aufweist.

5. Verschlüsselungsbeschleunigersystem nach Anspruch 3 oder 4, bei welchem jeder modulare Potenzierer eine

Größe von 544 Bits hat.

6. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem das Potenzierungs-Subsystem einen Scheduler (26), einen Potenzierer-Eingabepuffer (21) und einen Potenzierer-Ausgabepuffer (22) aufweist und der Scheduler eine Einrichtung zum Leiten von Ablaufbefehlen zu den Potenzierern über den Eingabepuffer (21) aufweist.

7. Verschlüsselungsbeschleunigersystem nach Anspruch 6, bei welchem die Befehle ein Statusfeld zum Einfügen eines Fehlers in dem Ausgabepuffer (22) enthalten, falls das Ergebnis verworfen werden soll.

8. Verschlüsselungsbeschleunigersystem nach Anspruch 6 oder 7, bei welchem die Befehle ein Kontrollfeld mit einem Gruppenmodusbefehl für eine Kettenkonfiguration enthalten.

9. Verschlüsselungsbeschleunigersystem nach Anspruch 8, bei welchem der Kontrollfeldbefehl mit einer speziellen Gruppe zusammen hängt.

10. Verschlüsselungsbeschleunigersystem nach einem der Ansprüche 6 bis 9, bei welchem die Befehle ein Blockkennungsfeld zum Einfügen einer Kennung des Blocks, der das Ergebnis erzeugte, in den Ausgabepuffer enthalten.

11. Verschlüsselungsbeschleunigersystem nach einem der Ansprüche 6 bis 10, bei welchem die Befehle ein Gruppenkennungsfeld zum Einfügen einer Kennung der Gruppe, die das Ergebnis erzeugte, in den Ausgabepuffer enthalten.

12. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem das Potenzierungs-Subsystem (7) eine Einrichtung zum Zugriff auf Steuerregister aufweist, einschließlich eines Registers für einen Befehl, der den Scheduler (26) eine Initialisierung von Gruppen unter Ausschluss von bestimmten fehleranfälligen Gruppen beginnen lässt.

13. Verschlüsselungsbeschleunigersystem nach Anspruch 12, bei welchem ein Steuerregister Inhalte eines linearen Rückkopplungsschieberegisters speichert.

14. Verschlüsselungsbeschleunigersystem nach einem der Ansprüche 6 bis 13, bei welchem der Scheduler (26) und der Eingabepuffer (21) eine Einrichtung zum Übertragen von Leerdaten zu den Potenzierern (30) bei Abwesenheit von echten Daten aufweisen.

15. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem die Hauptschnittstelle eine Hintergrundroutine und mehrere APIs (15) für einen Hauptrechner aufweist und die Hintergrundroutine eine Einrichtung zum Verwalten von Anforderungsketten auf einer Basis je Logikeinheit (6-9) aufweist.

16. Verschlüsselungsbeschleunigersystem nach Anspruch 15, bei welchem die CPU (3) einen Parser mit einer Einrichtung zum Knacken jeder Anforderung in Befehle, zum automatischen Bestimmen eines geforderten Antwortdatenraums und zum Zuordnen des Raums aufweist.

17. Verschlüsselungsbeschleunigersystem nach Anspruch 16, bei welchem jeder Parser zu einer speziellen Logikeinheit (6-9) gehört und eine Einrichtung zum Knacken der Befehle in Zeichenketten eines gewünschten Formats und einer gewünschten Größe für die zugehörige Logikeinheit aufweist.

18. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem die CPU (3) mehrere Mikrosequenzer aufweist, die jeweils eine Einrichtung entweder zum Leiten syntaktisch analysierter Befehlszeichenketten zu der Ziellogikeinheit oder zum Durchführen des geforderten Vorgangs selbst aufweisen.

19. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem die Logikeinheiten eine Blockverschlüsselungseinheit (6) mit einer Einrichtung zum Implementieren von Massen- und/oder symmetrischen Verschlüsselungsvorgängen aufweisen.

20. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem die Logikeinheiten einen Zufallszahlengenerator (8) mit einer Einrichtung zum Erzeugen eines Zufallszahlenbitstroms und zum Durchführen einer statistischen Analyse, um sicherzustellen, dass die Bits zufällig sind, aufweisen.

21. Verschlüsselungsbeschleunigersystem nach Anspruch 20, bei welchem die CPU (3) eine Einrichtung zum Verwenden des Zufallszahlenbitstroms zum Erzeugen von Primzahlen und zum Speichern der Primzahlen in konfigurierbaren Pools aufweist.

22. Verschlüsselungsbeschleunigersystem nach einem der vorhergehenden Ansprüche, bei welchem der Beschleuniger ferner einen Bus (5) zur Kommunikation der CPU (3) mit den Logikeinheiten (6-9) aufweist.

Revendications

1. Système accélérateur cryptographique comprenant une interface hôte (2) comportant des moyens pour l'interfaçage avec un système hôte possédant des applications nécessitant des opérations cryptographiques, des moyens pour l'exécution de l'élévation à une puissance, des moyens dans l'interface hôte pour l'acheminement vers le système hôte des réponses aux demandes, une pluralité d'unités logiques (6-9), y compris un sous-système d'élévation à une puissance (7) et une unité centrale de traitement (5) connectée entre l'interface hôte (2) et les unités logiques (6-9) et comprenant des moyens pour la gestion des opérations des unités logiques

caractérisé en ce que,

le sous-système d'élévation à une puissance comprend des systèmes modulaires individuels d'élévation à une puissance (30), d'une taille en bits prédéterminée, lequel peut fonctionner seul, et des moyens pour former de manière dynamique un groupe (20) de systèmes modulaires d'élévation à une puissance chaînés ensemble afin de fournir des élévations à une puissance jusqu'à un multiple de ladite taille de bits prédéterminée.

2. Système accélérateur cryptographique selon la revendication 1, dans lequel le sous-système d'élévation à une puissance (7) comprend un circuit intégré à application spécifique.

3. Système accélérateur cryptographique selon la revendication 1, dans lequel le sous-système d'élévation à une puissance (7) comprend des moyens pour le chaînage des systèmes modulaires d'élévation à une puissance (30) à l'intérieur d'un groupe (20) dans lequel toutes les chaînes à l'intérieur d'un groupe ont la même longueur.

4. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel le sous-système d'élévation à une puissance (7) comprend des moyens pour exécuter l'élévation à une puissance basée sur l'algorithme de Montgomery.

5. Système accélérateur cryptographique selon la revendication 3 ou 4, dans lequel chaque système modulaire d'élévation à une puissance a une taille de 544 bits.

6. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel le sous-système d'élévation à une puissance comprend un programmateur (26), une zone d'entrée du système d'élévation à une puissance (21), et une zone de sortie du système d'élévation à une puissance (22), et le programmateur comprend des moyens pour acheminer des instructions de programmation vers les systèmes d'élévation à une puissance via la zone d'entrée (21).

7. Système accélérateur cryptographique selon la revendication 6, dans lequel les instructions comprennent un champ d'état pour l'insertion d'une erreur dans la zone de sortie (22) si un résultat doit être écarté.

8. Système accélérateur cryptographique selon la revendication 6 ou 7, dans lequel les instructions comprennent un champ de commande avec une instruction de mode de groupe pour une configuration de chaînage.

9. Système accélérateur cryptographique selon la revendication 8, dans lequel l'instruction du champ de commande est associée avec un groupe particulier.

10. Système accélérateur cryptographique selon l'une quelconque des revendications 6 à 9, dans lequel les instructions comprennent un champ identificateur de bloc pour l'insertion dans la zone de sortie d'un identificateur du bloc qui a généré le résultat.

11. Système accélérateur cryptographique selon l'une quelconque des revendications 6 à 10, dans lequel les instructions comprennent un champ identificateur de bloc pour l'insertion dans la zone de sortie d'un identificateur du groupe qui a généré le résultat.

12. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel le sous-système d'élévation à une puissance (7) comprend des moyens pour accéder aux registres de commande, y compris un registre pour une instruction faisant que le programmateur (26) commence l'initialisation de groupes en excluant certains groupes prédisposés aux erreurs.

13. Système accélérateur cryptographique selon la revendication 12, dans lequel un registre de commande stocke les contenus de registre à décalage à boucle fermée.

14. Système accélérateur cryptographique selon l'une quelconque des revendications 6 à 13, dans lequel le programmateur (26) et la zone d'entrée (21) comprennent des moyens pour le transfert des données fictives vers les systèmes d'élévation à une puissance (30), en cas d'absence de données réelles.

15. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel l'interface hôte comprend un démon et une pluralité d'API (15) pour un système hôte, et que ledit démon comprend des moyens pour la gestion des files d'attente des demandes sur une base d'unité logique (6-9).

16. Système accélérateur cryptographique selon la revendication 15, dans lequel l'unité centrale de traitement (3) comprend un analyseur syntaxique comportant des moyens pour diviser chaque demande en commandes, pour déterminer automatiquement un espace de données de réponse requise, et pour allouer ledit espace.

17. Système accélérateur cryptographique selon la revendication 16, dans lequel chaque analyseur syntaxique est associé avec une unité logique particulière (6-9) et comprend des moyens pour diviser les commandes en chaînes d'une taille et d'un format désirés pour l'unité logique associée.

18. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel l'unité centrale de traitement (3) comprend une pluralité de micro-séquenceurs, chacun comprenant des moyens pour, soit acheminer des chaînes de commande faite par l'analyse syntaxique vers l'unité logique de destination, soit exécuter lui-même l'opération demandée.

19. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel les unités logiques comprennent une unité de chiffrement par bloc (6) qui comprend des moyens pour une implémentation en masse et / ou des opérations de chiffres symétriques.

20. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel les unités logiques comprennent un générateur de nombres aléatoires (8) comprenant des moyens pour générer un train binaire de nombres aléatoires, et pour exécuter une analyse statistique afin de s'assurer que ces bits sont aléatoires.

21. Système accélérateur cryptographique selon la revendication 20, dans lequel l'unité centrale de traitement (3) comprend des moyens pour utiliser le train binaire de nombres aléatoires afin de générer des nombres premiers et stocker les nombres premiers dans des ensembles configurables.

22. Système accélérateur cryptographique selon l'une quelconque des revendications précédentes, dans lequel l'accélérateur comprend un bus (5) pour faire communiquer l'unité centrale de traitement (3) avec les unités logiques (6-9).

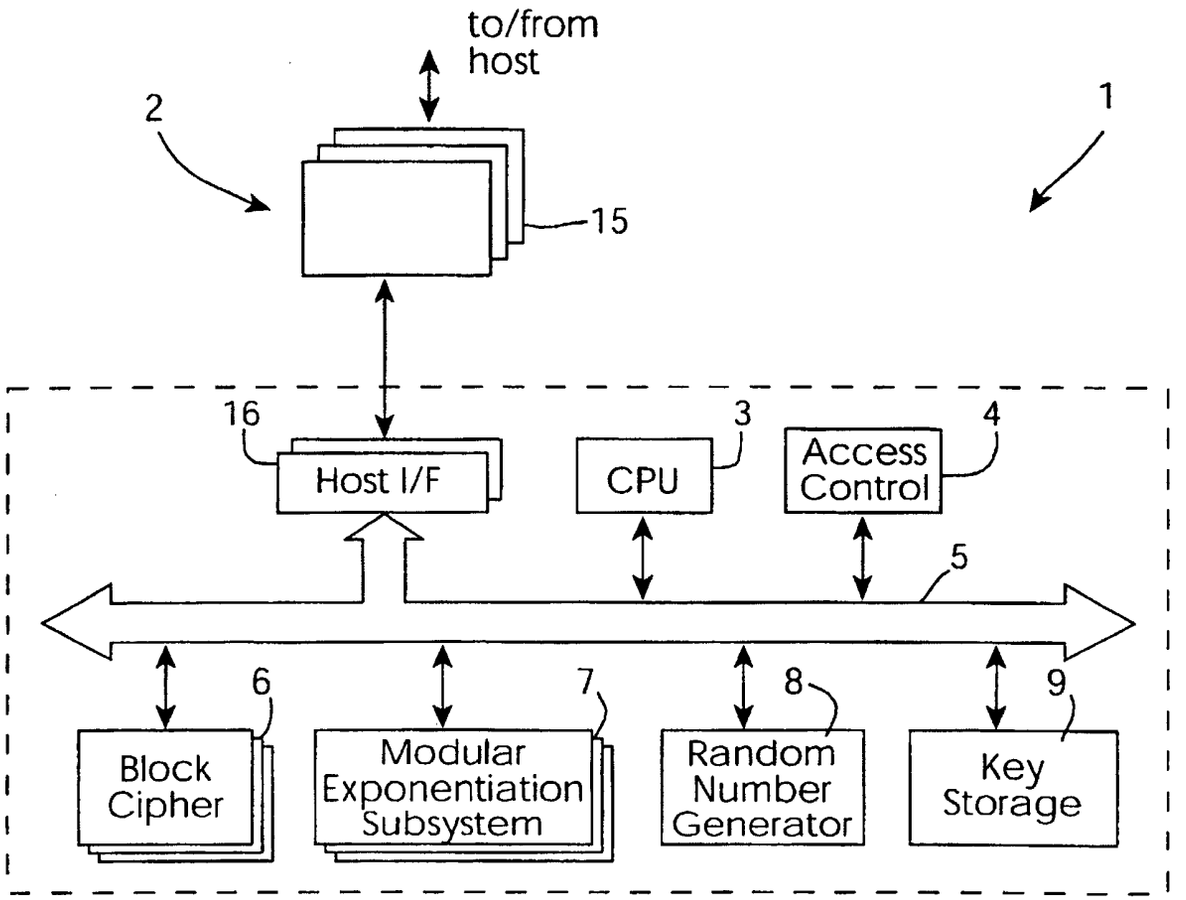

to/from
host
2
1
15
16
Host I/F
3
CPU
4
Access
Control
5
6
Block
Cipher
7
Modular
Exponentiation
Subsystem
8
Random
Number
Generator
9
Key
Storage

Fig. 1

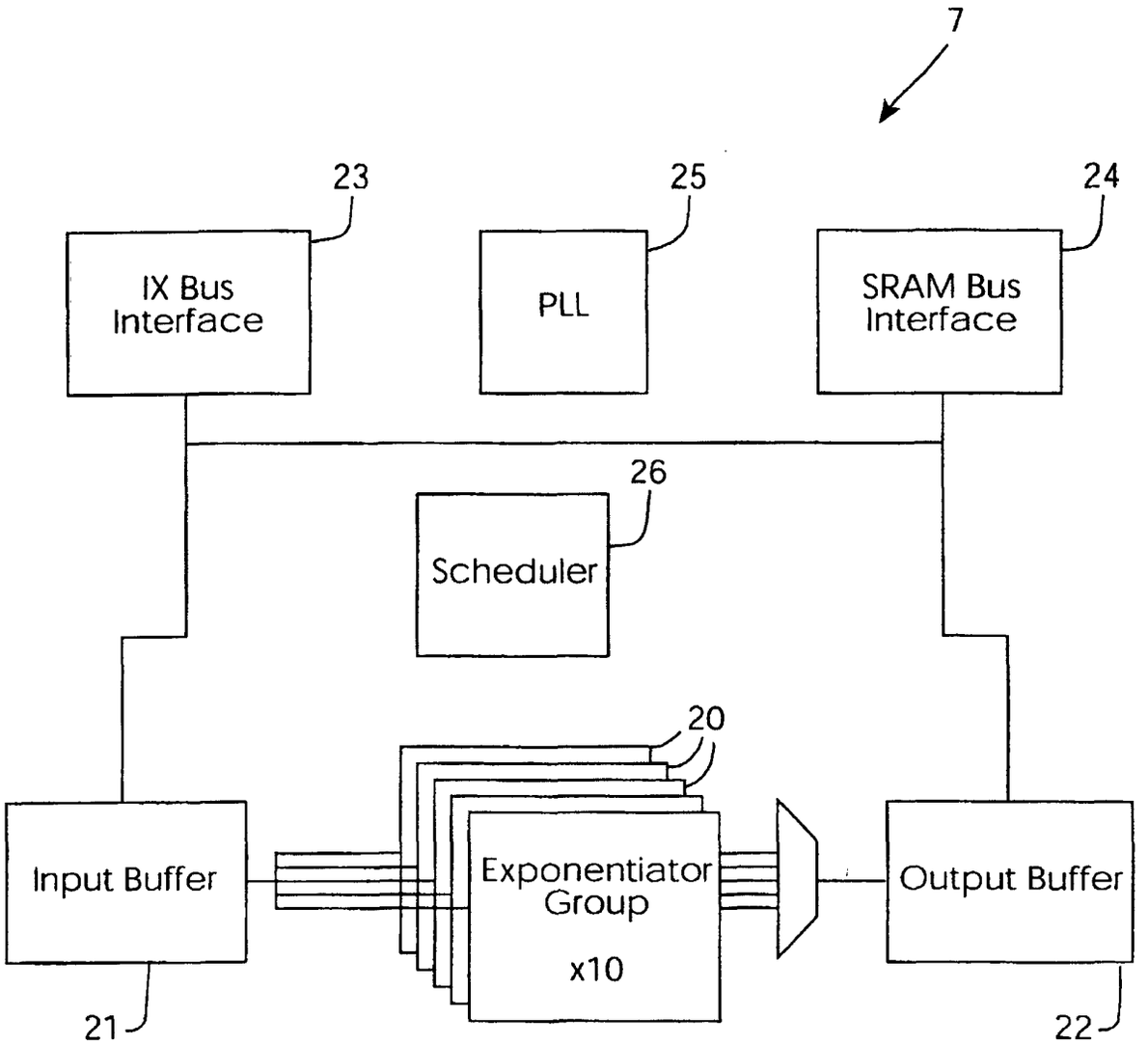

7
23
25
24
IX Bus Interface
PLL
SRAM Bus Interface
26
Scheduler
20
Input Buffer
Exponentiator Group
x10
Output Buffer
21
22

Fig. 2

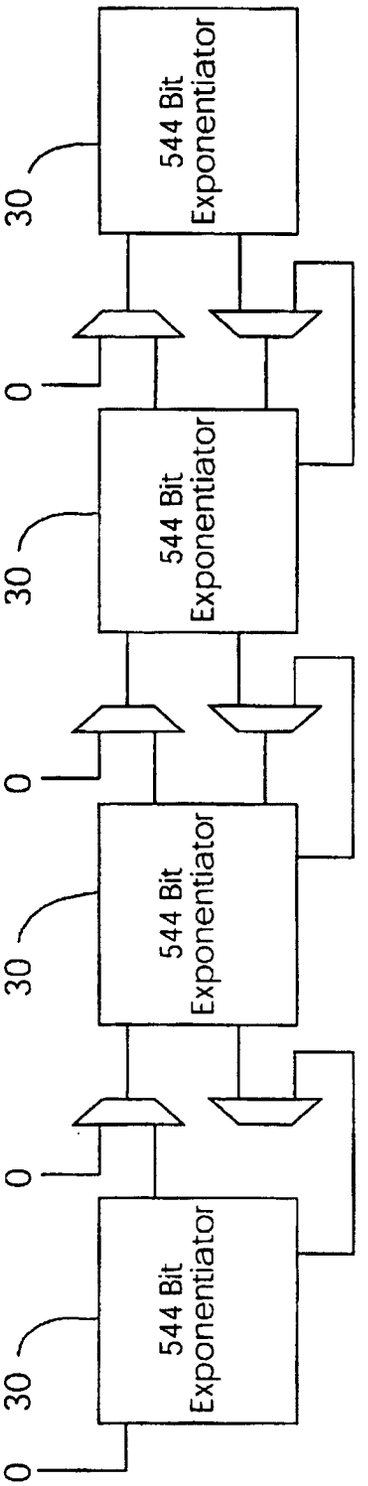

544 Bit Exponentiator
544 Bit Exponentiator
544 Bit Exponentiator
544 Bit Exponentiator
30
30
30
30
0
0
0
0

Fig. 3