

CLAIMS

What is claimed is:

1. A node in a network, comprising:

a meter configured to measure consumption of a resource comprising electricity, gas or water, and configured to generate encrypted data representing consumption of the resource and to record the encrypted data in a memory device located externally to a
5 microcontroller unit;

a first cache, located within the microcontroller unit, and configured to store a block of the encrypted data obtained from the memory device;

a decryption engine, located within the microcontroller unit, and configured to obtain
10 the block of the encrypted data from the first cache and to decrypt the block of the encrypted data, thereby producing a block of decrypted data;

a second cache, located within the microcontroller unit, and configured to receive the block of decrypted data from the decryption engine;

an address alignment module configured to receive input from a program counter
15 within the microcontroller unit and to calculate an offset pointer to a particular word in the block of decrypted data within the second cache;

an address generator configured to receive input from the address alignment module and to indicate a block of memory in the memory device to be loaded into the first cache; and

20 an application having program statements configured for operation on the microcontroller unit, the program statements being stored on the external memory in encrypted form and being executed in decrypted form in an order indicated by the offset pointer.

25 2. The node of claim 1, wherein the address alignment module calculates the offset pointer to the particular word based on a mapping of an address in the memory device external to the microcontroller unit to a register in the second cache.

3. A node, comprising:

30 a microcontroller;

a memory device located external to the microcontroller and comprising encrypted data and encrypted software program statements; and

a meter configured to measure consumption of a resource comprising electricity, gas or water, and configured to generate the encrypted data, wherein the encrypted data
5 represents consumption of the resource;

wherein the microcontroller comprises:

a first cache configured to store N words of encrypted data obtained from a memory device external to the processing device;

10 a decryption engine configured to obtain the N words of encrypted data from the first cache and to decrypt the N words;

a second cache, configured to receive the N words of decrypted data from the decryption engine; and

an address alignment and generator module, configured to:

15 receive input from a program counter and to calculate an offset pointer to a particular word of the N words of decrypted data within the second cache; and

indicate N words of encrypted data in the memory device external to the processing device to be loaded into the first cache.

20 4. The node of claim 3, wherein the address alignment and generator module translates the input from the program counter, which indicates a word from a particular location in external memory, into an offset to a register in the second cache.

5. The node of claim 3, wherein the address alignment and generator module is additionally configured to:

25 increment the offset pointer in response to input from the program counter, wherein the offset pointer points to a decrypted word of data for transmission to the instruction register.

6. The node of claim 3, wherein:

30 calculating the offset pointer includes translating an address, the address originally indicating a location in the memory device external to the processing device, to indicate a register in the second cache; and

indicating N words of encrypted data in the memory device includes mapping an address obtained from the program counter to indicate a block of memory in the external memory device.

5 7. The node of claim 3, wherein:

the address alignment and generator module signals the decryption engine to overwrite the second cache with newly decrypted data; and

the offset pointer is reset when decrypted data from the decryption engine overwrites the second cache.

10

8. The node of claim 3, wherein address alignment and generator module is additionally configured to:

adapt address buses of different widths; and

notify the first cache when it is time to read the data bus.

15

9. The node of claim 3, additionally comprising:

a first clock configured to send a signal to the decryption engine; and

a second clock configured to send a signal to the instruction register and the program counter.

20

10. The node of claim 3, wherein:

the first cache contains one block of encrypted data;

the decryption engine contains one block of data in the process of decryption; and

the second cache contains one block of decrypted data.

25

11. A method in a processor of executing encrypted code from an external memory, comprising:

receiving, from a meter, encrypted meter data representing consumption of a resource comprising electricity, gas or water;

30 recording the encrypted meter data in a memory device located externally to a microcontroller unit;

fetching a first block of encrypted program statements from the external memory;

storing the fetched block of encrypted program statements in a first cache memory;

decrypting one or more encrypted program statements obtained from the first cache memory;

storing the decrypted program statements in a second cache memory;

receiving an address from a program counter, the address indicating a next statement
5 from among the stored decrypted program statements to be executed by a microcontroller;

accessing the encrypted meter data by execution of statements indicated by the program counter;

calculating an offset to a decrypted program statement in the second cache memory, the calculating comprising incrementing an offset pointer;

10 calculating an address of a second block of program statements in the external memory based at least in part on an address of the first block of program statements; and

fetching the second block of encrypted program statements from the external memory to provide executable statements for execution after a last decrypted program statement in the second cache memory.

15

12. The method of claim 11, wherein fetching the block of encrypted program statements from the external memory is in response to completing the decrypting of a previous block of encrypted program statements.

20

13. The method of claim 11, wherein decrypting the program statements comprises decrypting the program statements using a decryption engine configured in hardware for AES decryption.

25

14. The method of claim 13, wherein decrypting the program statements comprises receiving a clock signal at the decryption engine that is faster than a clock signal provided to the program counter.

30

15. The method of claim 11, additionally comprising calculating an offset to a word in the second cache memory, the offset based on the address received from the program counter.

16. The method of claim 11, additionally comprising calculating an address in the external memory associated with the address received from the program counter, wherein

the calculating comprises determining a block of data in the external memory that follows a block of data in the external memory that was most recently fetched.

17. The method of claim 11, wherein fetching the block of encrypted program
5 statements and decrypting the program statements obtained from the first cache memory are performed simultaneously.

18. The method of claim 11, additionally comprising:

10 fetching the second block of encrypted program statements from the external memory after the offset pointer has pointed to each register in the second cache memory; and
resetting the offset pointer to point at a first register in response to an increment of the offset pointer when the offset pointer is pointing to the last register.

19. A method of executing encrypted code from an external memory, comprising:

15 fetching a block of encrypted data from the external memory;
storing the fetched block of encrypted data in a first cache memory;
decrypting data obtained from the first cache memory, wherein decrypting the data comprises decrypting the data using a decryption engine configured in hardware for AES decryption, and wherein decrypting the data comprises receiving a clock signal at the
20 decryption engine that is faster than a clock signal provided to a program counter;
storing the decrypted data in a second cache memory;
receiving an address from the program counter, the address indicating a next statement to be executed by a micro-controller;
calculating an offset to a word in the second cache memory, the offset based on the
25 address received from the program counter;
providing the word indicated by the offset to an instruction register;
calculating an address in the external memory associated with the address received from the program counter; and
fetching a second block of encrypted data from the external memory, the second block
30 of encrypted data being associated with the calculated address.

CLAIMS

What is claimed is:

1. A node in a network, comprising:
 - 5 a meter configured to generate encrypted data representing consumption of a resource and to record the encrypted data in a memory device located externally to a microcontroller unit;
 - a first cache, located within the microcontroller unit, and configured to store a block of encrypted data obtained from the memory device;
 - 10 a decryption engine, located within the microcontroller unit, and configured to obtain the block of encrypted data from the first cache and to decrypt the block of encrypted data, thereby producing a block of decrypted data;
 - a second cache, located within the microcontroller unit, and configured to receive the block of decrypted data from the decryption engine;
 - 15 an address alignment module configured to receive input from a program counter within the microcontroller unit and to calculate an offset pointer to a particular word in the block of decrypted data within the second cache;
 - 20 an address generator configured to receive input from the address alignment module and to indicate a block of memory in the memory device to be loaded into the first cache; and
 - an application having program statements configured for operation on the microcontroller unit, the program statements being stored on the external memory in encrypted form and being executed in decrypted form in an order indicated by the offset pointer.

2. The node of claim 1, wherein the address alignment module calculates the offset pointer to the particular word based on a mapping of an address in the memory device external to the microcontroller unit to a register in the second cache.

5

3. A processing device, comprising:

a first cache configured to store N words of encrypted data obtained from a memory device external to the processing device;

a decryption engine configured to obtain the N words of encrypted data from the first cache and to decrypt the N words;

a second cache, configured to receive the N words of decrypted data from the decryption engine;

an address alignment and generator module, configured to:

receive input from a program counter and to calculate an offset pointer to a particular word of the N words of decrypted data within the second cache; and

indicate N words of encrypted data in the memory device external to the processing device to be loaded into the first cache.

4. The processing device of claim 3, wherein the address alignment and generator module translates the input from the program counter, which indicates a word from a particular location in external memory, into an offset to a register in the second cache.

25

5. The processing device of claim 3, wherein the address alignment and generator module is additionally configured to:

increment the offset pointer in response to input from the program counter, wherein the offset pointer points to a decrypted word of data for transmission to the instruction register.

6. The processing device of claim 3, wherein:

calculating the offset pointer includes translating an address, the address originally indicating a location in the memory device external to the processing device, to indicate a register in the second cache; and

indicating N words of encrypted data in the memory device includes mapping an address obtained from the program counter to indicate a block of memory in the external memory device.

7. The processing device of claim 3, wherein:

the address alignment and generator module signals the decryption engine to overwrite the second cache with newly decrypted data; and

the offset pointer is reset when decrypted data from the decryption engine overwrites the second cache.

8. The processing device of claim 3, wherein address alignment and generator module is additionally configured to:

adapt address buses of different widths; and

notify the first cache when it is time to read the data bus.

9. The processing device of claim 3, additionally comprising:
a clock configured to send a signal to the decryption engine; and
a second clock to send a signal to the instruction register and the
program counter.

5

10. The processing device of claim 3, wherein:
the first cache contains one block of encrypted data;
the decryption engine contains one block of data in the process of
decryption; and

10 the second cache contains one block of decrypted data.

11. A method of executing encrypted code from an external memory,
comprising:

15 fetching a block of encrypted data from the external memory;
storing the fetched block of encrypted data in a first cache memory;
decrypting data obtained from the first cache memory;
storing the decrypted data in a second cache memory;
receiving an address from a program counter, the address indicating a
next statement to be executed by a microcontroller;
20 calculating an offset to a word in the second cache memory, the offset
based on the address received from the program counter;
providing the word indicated by the offset to an instruction register;
calculating an address in the external memory associated with the
address received from the program counter; and
25 fetching a second block of encrypted data, the second block being
associated with the calculated address.

12. The method of claim 11, wherein fetching the block of encrypted data from the external memory is in response to:

completing the decrypting of the data; and

incrementing the offset to the last word in the second cache.

5

13. The method of claim 11, wherein decrypting the data comprises:

decrypting the data using a decryption engine configured in hardware for AES decryption.

10

14. The method of claim 13, wherein decrypting the data comprises:

receiving a clock signal at the decryption engine that is faster than a clock signal provided to the program counter.

15

15. The method of claim 11, wherein calculating the offset to the word in the second cache memory comprises:

incrementing an offset pointer to indicate the word contained within the second cache, wherein the incrementing is based on action by the program counter.

20

16. The method of claim 11, wherein calculating the address in the external memory associated with the address received from the program counter comprises:

determining a block of data in the external memory that follows a block of data in the external memory that was most recently fetched.

25

17. The method of claim 11, wherein:

fetching the block of encrypted data, decrypting the data obtained from the first cache memory and providing the word indicated by the offset are performed simultaneously.

5

18. The method of claim 11, wherein:

calculating the offset to the word in the second cache memory comprises incrementing an offset pointer by one register; and

calculating the address in the external memory comprises calculating an address of a block of data in the external memory that includes the address received from the program counter.

10

19. The method of claim 11, wherein fetching the second block of encrypted data comprises:

fetching the second block after an offset pointer has pointed to a last word in the second cache.

15

20. The method of claim 11, wherein fetching the second block of encrypted data comprises:

fetching the second block after an offset pointer has pointed to each register in the second cache; and

20

resetting the offset pointer to point at a first register in response to an increment of the offset pointer when the offset point is pointing to the last register.

25