17 OCT 2002

G00002

Access-protected data carrier

This invention relates to a data carrier having a semiconductor chip in which secret data are stored. The invention relates in particular to a smart card.

Data carriers containing chips are used in a great number of different applications, for example for performing monetary transactions, paying for goods or services or as identification means for access or admission controls. In all such applications the chip of the data carrier normally processes secret data which must be protected from access by unauthorized third parties. Such protection is ensured by, among other things, giving the inner structures of the chip very small dimensions so that it is very difficult to access said structures with the aim of spying out data processed in said structures. In order to impede access further, the chip can be embedded in a very firmly adhering mass whose forcible removal destroys the semiconductor plate or at least the secret data stored therein. It is likewise possible to provide the semiconductor plate during its production with a protective layer which cannot be removed without destroying the semiconductor plate.

With corresponding technical equipment, which is extremely expensive but nevertheless fundamentally available, an attacker could possibly succeed in exposing and examining the inner structure of the chip. Exposure could be effected for example by special etching methods or a suitable grinding process. The thus exposed structures of the chip, such as conductive paths, could be contacted with microprobes or examined by other methods to determine the signal patterns in said structures. Subsequently one could attempt to determine from the detected signals secret data of the data carrier, such as secret keys, in order to use them for purposes of manipulation. One could likewise attempt to selectively influence the signal patterns in the exposed structures via the microprobes.

The invention is based on the problem of protecting secret data present in the chip of a data carrier from unauthorized access.

This problem is solved by the feature combinations of the independent claims.

The inventive solution, unlike the prior art, involves no measures to prevent exposure of the internal structures of the chip and the mounting of microprobes. In-

stead measures are taken to make it difficult for a potential attacker to infer secret information from any intercepted signal patterns. The signal patterns depend on the operations which the chip is performing. Said operations are controlled with the aid of an operating program stored in a memory of the chip. The operating program is composed of a series of individual commands each triggering an exactly specified operation. So that the chip can perform the intended functions a corresponding command string is to be defined for each of said functions. Such a function can be for example the encryption of data with the aid of a secret key. To give an attacker intercepting the processes on the chip by microprobes he has mounted as little information as possible about the particular commands executed and the data used in executing the commands, a desired function is preferably realized using commands of such a kind, or using commands in such a way, that it is difficult if not impossible to spy out information. In other words, no commands or command strings are to be used which allow the processed data to be inferred in a simple way by interception.

It is always especially easy to infer data when a command processes very few data, for example one bit. For this reason one preferably uses commands, according to an embodiment of the invention, which simultaneously process a plurality of bits, e.g. one byte, at least for all security-relevant operations, such as encryption of data. Such simultaneous processing of a plurality of bits blurs the influence the individual bits have on the signal pattern caused by the command into a total signal from which it is very difficult to infer the individual bits. The signal pattern is much more complex than in the processing of individual bits and it is not readily evident which part of the signal belongs to which bit of the processed data.

Additionally or alternatively, one can impede an attack on the processed data according to the invention by using in security-relevant operations solely commands which trigger an identical or very similar signal pattern or commands by which the processed data have very little or no influence on the signal pattern.

According to another advantageous embodiment of the invention, one performs security-relevant operations not with authentic secret data but with falsified secret data from which the authentic secret data cannot be determined without the addition of further secret information. This means that even if an attacker succeeds in deter-

mining the secret data used in an operation, he cannot cause any damage since the spied-out data are not the authentic secret data but falsified secret data.

In order to guarantee the functioning of the data carrier one must ensure that the data carrier delivers the right results when rightfully used despite the falsified secret data. This is obtained by first specifying a function for falsifying the authentic secret data, for example EXORing the secret data with a random number. The authentic secret data are falsified with the thus specified function. The falsified secret data are used to perform all those operations in the data carrier in which falsification of the secret data can subsequently be compensated. In the case of EXOR-falsified secret data, these would be operations which are linear with respect to EXOR operations. Before execution of an operation not permitting such compensation, for example an operation which is nonlinear with respect to EXOR operations, the authentic secret data must be restored so that said operation is performed with the authentic secret data. The authentic secret data are restored after execution of a compensable function for example by EXORing the function value determined by means of the falsified secret data with a corresponding function value of the random number used for falsification. It is important in this context for random number and function value to be previously determined and stored in safe surroundings so that the calculation of the function value from the random number cannot be intercepted.

The above procedure means that the authentic secret data are used only for performing operations, such as nonlinear operations, for which this is absolutely necessary, i.e. which cannot be performed alternatively with falsified secret data. Since such operations are normally very complex and not easy to analyze, it is extremely difficult if not impossible for a potential attacker to find out the authentic secret data from analyzing the signal patterns caused by said operations. Since the simply structured functions permitting subsequent compensation of falsification are performed with falsified secret data, the described procedure makes it extremely difficult to determine the authentic secret data of the data carrier from illegally intercepted signal patterns.

The signal patterns depend on the operations which the chip is executing. If said operations are always executed according to the same rigid pattern, i.e. in par-

17 OCT 2002

ticular in the same order, and the attacker knows this order, an attacker need overcome much fewer difficulties to spy out data than if he does not even know which operation is being executed at which time. It is therefore provided according to a further embodiment of the invention to move as far away as possible from a rigid flow pattern when executing security-relevant operations within the smart card, thereby offering the attacker next to no hints for analyzing the secret data. This is obtained by executing as many operations as possible, ideally even all operations, which are independent of each other insofar as each of the operations requires no data determined by the other operations, in a variable order, for example one that is random or dependent on input data. This achieves the result that an attacker, who will normally be oriented by the order of the operations, cannot readily find out which operation is being executed. This holds especially when the operations resemble each other very strongly or are even the same with respect to the signal pattern they cause with the same input data. If the attacker does not even know the kind of operation which is being executed, it is extremely difficult to spy out data selectively. If there is the danger of an attacker making a great number of spying attempts in order to average out the random variation of the order, it is recommendable to make the variation dependent on the input data.

The invention will be explained below with reference to the embodiments shown in the figures, in which:

Fig. 1 shows a smart card from the front, and

Fig. 2 shows a greatly enlarged detail of the chip of the smart card shown in Fig. 1 from the front.

Fig. 3 shows a schematic representation of part of an operational sequence within the smart card, and

Fig. 4 shows a variant of the operational sequence shown in Fig. 3.

Fig. 5 shows a schematic representation of the sequence in the execution of some operations by the smart card.

Fig. 1 shows smart card 1 as an example of the data carrier. Smart card 1 is composed of card body 2 and chip module 3 set in a specially provided gap in card body 2. Essential components of chip module 3 are contact surfaces 4 via which an

17 OCT 2002

electric connection can be made with an external device, and chip 5 electrically connected with contact surfaces 4. Alternatively or in addition to contact surfaces 4, a coil not shown in Fig. 1 or other transfer means can be present for producing a communication link between chip 5 and an external device.

Fig. 2 shows a greatly enlarged detail of chip 5 from Fig. 1 from the front. The special feature of Fig. 2 is that it shows the active surface of chip 5, i.e. Fig. 2 omits all layers which generally protect the active layer of chip 5. In order to obtain information about the signal patterns inside the chip one can for example contact exposed structures 6 with microprobes. The microprobes are very thin needles which are brought in electric contact with exposed structures 6, for example conductive paths, by means of a precision positioning device. The signal patterns picked up by the microprobes are processed with suitable measuring and evaluation devices in order to infer secret data of the chip.

The invention achieves the result that an attacker cannot gain access, or only with great difficulty, to in particular secret data of the chip even if he succeeds in removing the protective layer of chip 5 without destroying the circuit and contacting exposed structures 6 of chip 5 with microprobes or otherwise intercepting them. The invention is of course also effective if an attacker gains access to the signal patterns of chip 5 in another way.

According to the invention, the commands or command strings of the operating program of the chip are selected at least in all security-relevant operations in such a way that the data processed with the commands can either not be inferred at all or at least only with great difficulty from the intercepted signal patterns.

This can be achieved for example by fundamentally using in security operations no commands which process individual bits, such as the shift of individual bits, intended to cause a permutation of the bits of a bit string. Instead of bit commands one can use for example byte commands such as copy or rotation commands which process not an individual bit but a whole byte comprising eight bits. The byte command triggers a much more complex signal pattern than the bit command, it being extremely difficult to associate individual bits with partial areas of the signal pattern.

17 OCT 2002

This blurs the information processed with the byte command, making it difficult to spy out said information.

Further, the invention offers the possibility of fundamentally using in security-relevant operations only commands triggering a very similar signal pattern so that it is very difficult to differentiate the commands being executed by the signal patterns. It is likewise possible to design the commands so that the kind of processed data has very little or no influence on the signal pattern triggered by the command.

The described variants can be used either alternatively or in combination with respect to the individual commands. An inventive set of security-relevant commands can thus be composed of commands belonging to one or more of the abovementioned variants. One can likewise use an instruction set in which all commands belong to the same variant, it also being allowed that some or all commands belong to other variants as well. For example, one can allow solely byte commands, preferably using those commands which in addition trigger a very similar signal pattern.

Security-relevant operations include e.g. encryption operations which are frequently used in smart cards. Such encryptions involve execution of a series of single operations which lead to bit-by-bit changes in a data word. According to the invention all these commands are replaced with byte commands and/or the abovementioned inventive measures are taken. This makes it even more difficult for an attacker to infer the secret keys used in encryption from the intercepted signal patterns, thereby preventing abuse of said secret keys.

Fig. 3 shows a schematic representation of part of an operational sequence in the smart card. An encryption operation was selected for the representation by way of example. However, the principles explained by this example are also applicable to any other security-relevant operations. At the onset of the part of the encryption operation shown in Fig. 3 data *abc*, which can be present in plaintext or already encrypted, are supplied to logic point 7. At logic point 7 data *abc* are combined with key $K1$. In the present example this combination is an EXOR operation but other suitable forms of combination can also be used. Nonlinear function g is then applied to the result of combination in function block 8. In order to show that function block 8 represents a nonlinear function it has the form of a distorted rectangle in Fig. 3.

17 OCT 2002

The data produced with function block 8 are EXORed with random number Z at logic point 9 and subsequently processed in function block 10. Combination with random number Z causes falsification of the data which makes it difficult for an attacker to analyze the processes in function block 10 representing a linear mapping by means of function f. An undistorted rectangle is used as a symbol of a linear function in Fig. 3. The data produced in function block 10 are combined at logic point 11 with data $f(Z)$ previously generated e.g. during production of the card by application of function f to random number Z. This combination compensates the falsification of the data with random number Z at logic point 9. Said compensation is necessary since nonlinear function g is subsequently to be applied to the data in function block 12 and compensation of falsification is no longer possible after application of a nonlinear function to the data. Further, the data are EXORed at logic point 11 with key $K2$ which is necessary in connection with the encryption operation.

The combination at logic point 11 with the data $f(Z)$ and $K2$ can be effected either with single components $K2$ and $f(Z)$ or with the result of an EXOR operation of said components. The latter procedure opens up the possibility of key $K2$ not needing to be available in plaintext but only key $K2$ EXORed with $f(Z)$. If this combination value was calculated and stored in the memory of the card previously, e.g. during initialization or personalization of smart card 1, it is unnecessary to store key $K2$ in smart card 1 in plaintext. This further increases the security of smart card 1.

After application of function g to the data in function block 12 the thus determined result is in turn combined with random number Z at logic point 13 and thereby falsified. Linear function f is then applied to the result of combination in function block 14. Finally, the data are EXORed with the result of an application of function f to random number Z and with key $K3$ at logic point 15. This operation can be followed by further processing steps not shown in Fig. 3.

All in all, the procedure shown in Fig. 3 can be summarized by saying that the data processed in the encryption operation are falsified whenever possible by EXORing with random number Z in order to prevent secret data from being spied out. Falsification is fundamentally possible with all functions f showing linear behavior with respect to EXOR operations. With nonlinear functions g the unfalsified data

17 OCT 2002

must be used. It is therefore necessary that the falsification be compensated by EX-ORing the data with function value $f(Z)$ before application of nonlinear function g to the data. It is less critical from a security point of view that nonlinear functions g can only be applied to the unfalsified data since said nonlinear functions g are much more difficult to spy out than linear functions f. The diagram shown in Fig. 3 is applicable both for identical functions g or functions f and for different respective functions.

The diagram shown in Fig. 3 achieves the result that it is almost impossible to spy out secret data during the processing of data *abc*. However, since upon provision of secret keys $K1$, $K2$ and $K3$ operations are also to be executed with said keys which could in turn be the target of a spying attempt by an attacker, it is recommendable to take corresponding safety precautions in the processing of the keys. An embodiment of the invention involving such safety precautions is shown in Fig. 4.

Fig. 4 shows a part corresponding to Fig. 3 of an operational sequence of a smart card for a further variant of the invention. Processing of data *abc* is identical to Fig. 3 and will therefore not be explained again in the following. In contrast to Fig. 3, however, keys $K1$, $K2$ and $K3$ are not supplied to logic points 7, 11 and 15 in Fig. 4. Instead, falsified keys $K1'$, $K2'$ and $K3'$ are supplied together with random numbers $Z1$, $Z2$ and $Z3$ required for compensating falsification, the falsified keys preferably being supplied first and then the random numbers. This ensures that proper keys $K1$, $K2$ and $K3$ do not appear at all. This procedure is especially advantageous in encryption methods by which keys $K1$, $K2$ and $K3$ are derived from common key K. In this case key K falsified with random number Z is stored in smart card 1, and random numbers $Z1$, $Z2$ and $Z3$ determined by application of the key derivation method to random number Z are stored in smart card 1. Storage must be done in safe surroundings, for example in the personalization phase of smart card 1.

For carrying out the functional diagram shown in Fig. 4 one requires not only the stored data but also falsified derived keys $K1'$, $K2'$ and $K3'$. Said keys can be derived from falsified key K when they are required. With this procedure no operations are performed with authentic key K or authentic derived keys $K1$, $K2$ and $K3$ so that it is virtually impossible to spy out said keys. Since derived random numbers $Z1$, $Z2$

17 OCT 2002

and $Z3$ were also determined and stored in smart card 1 in advance, no more operations are performed therewith which could be spied out by an attacker. Thus, no access is possible to authentic derived keys $K1$, $K2$ and $K3$ by spying out falsified derived keys $K1'$, $K2'$ and $K3'$ since this requires derived random numbers $Z1$, $Z2$ and $Z3$.

In order to increase security further it is also possible to use a different random number Z for each EXOR operation, making sure that an $f(Z)$ is then also present for compensating the falsification in each case. In one embodiment, all random numbers Z and function values $f(Z)$ are stored in the memory of the smart card. However, it is likewise possible to store only a small number of random numbers R and function values $f(Z)$ and determine new random numbers Z and function values $f(Z)$ by EXORing or another suitable combination of several stored random numbers Z and function values $F(Z)$ whenever said values are required. Random numbers Z can be selected for EXORing from the set of stored random numbers Z at random.

In a further embodiment, there is no storage of random numbers Z and function values $f(Z)$ since they are generated by means of suitable generators whenever required. It is important that the generator or generators do not generate function values $f(Z)$ by applying linear function f to random number Z but that pairs of random numbers Z and function values $f(Z)$ be generated in another way since random number Z might otherwise be spied out by interception of the application of function f to random number Z and further secret data determined with the aid of this information.

According to the invention, basically all security-relevant data, for example keys, can be falsified with the aid of further data, such as random numbers, and then be supplied to processing. This achieves the result that an attacker spying out said processing can only determine worthless data since they are falsified. At the end of processing the falsification is undone.

Fig. 5 shows a schematic representation of the sequence during execution of some operations by the smart card. Fig. 5 shows in particular which operations must necessarily be executed sequentially by smart card 1 since they depend on each other, and which operations can basically be executed in parallel and thus in any order. In this connection Fig. 5 shows part of a program run of smart card 1 in which

17 OCT 2002

data *abc* are processed. All operations that have to be executed sequentially are shown sequentially in Fig. 5. All operations not requiring a special order of execution are disposed in parallel.

Processing of data *abc* begins with operation *P*1 shown in the form of block 70. The block is followed sequentially by block 80 representing operation *P*2. Fig. 5 thus indicates that the processing order of operations *P*1 and *P*2 cannot be interchanged, i.e. is obligatory. After block 80 the diagram shown in Fig. 5 branches into five blocks 90, 100, 110, 120, 130 representing operations *P*3, *P*4, *P*5, *P*6 and *P*7. It results that blocks *P*3, *P*4, *P*5, *P*6 and *P*7 can be executed simultaneously and thus also executed in any order. According to the invention the execution order of operations *P*3, *P*4, *P*5, *P*6, *P*7 is varied in each run, i.e. it is not foreseeable for an attacker which of said operations follows operation *P*2, which operations are performed after that, etc. Variation of the order can be effected either according to a fixed pattern or, better still, randomly or in accordance with input data by fixing by means of a random number or by the input data which of operations *P*3, *P*4, *P*5, *P*6 and *P*7 is executed next. This possibly random variation of the execution of the individual operations makes it difficult to spy out the data processed with the operations. When all operations *P*3, *P*4, *P*5, *P*6 and *P*7 are executed, operation *P*8 necessarily follows whose processing order is not variable. Operation *P*8 is shown by block 140. Operation *P*8 can be followed by further operations whose order is either variable or fixed, which are not shown in Fig. 5.

The invention can be used for example for the execution of encryption algorithms which frequently contain similar operations whose processing order is variable. The processing order can either be fixed before the first variable operation jointly for all operations interchangeable with said first operation, or the operation to be processed next can be determined before each variable operation from the set of remaining variable operations. In both cases one can use random numbers for fixing the processing order.

17 OCT 2002

Patent claims

1. A data carrier with a semiconductor chip (5) having at least one memory in which an operating program containing a plurality of commands is stored, each command causing signals detectable from outside the semiconductor chip (5), characterized in that the data carrier is designed to perform security-relevant operations solely executing operating program commands of such a kind, or executing said commands in such a way, that the data processed with the corresponding commands cannot be inferred from the detected signals.
2. A data carrier according to claim 1, characterized in that the commands used are designed for at least byte-by-byte processing of data.
3. A data carrier according to either of the above claims, characterized in that the commands used do not differ, or differ very little, from each other with respect to the signal patterns caused thereby.
4. A data carrier according to any of the above claims, characterized in that the commands used each lead to a signal pattern which does not depend, or depends only to a very small extent, on the data processed with the command.
5. A data carrier according to any of the above claims, characterized in that the operating program is able to execute a series of operations (f), input data being required for executing the operations (f) and output data being generated by execution of the operations (f), whereby
   - the input data are falsified by combination with auxiliary data (Z) before execution of one or more operations (f),
   - the output data determined by execution of the one or more operations (f) are combined with an auxiliary function value ($f(Z)$) in order to compensate the falsification of the input data,
   - whereby the auxiliary function value was previously determined by execution of the one or more operations (f) with the auxiliary data (Z) as input data in safe surroundings and stored on the data carrier along with the auxiliary data (Z).

17 OCT 2002

6. A data carrier according to claim 5, characterized in that the combination with the auxiliary function values ($f(Z)$) for compensating the falsification is performed at the latest directly before execution of an operation (g) which is nonlinear with respect to the combination generating the falsification.
7. A data carrier according to either of claims 5 and 6, characterized in that the auxiliary data (Z) are varied, the corresponding auxiliary function values ($f(Z)$) being stored in the memory of the data carrier.
8. A data carrier according to claim 7, characterized in that new auxiliary values (Z) and new auxiliary function values ($f(Z)$) are generated by combining two or more existing auxiliary data (Z) and auxiliary function values ($f(Z)$).
9. A data carrier according to claim 8, characterized in that the existing auxiliary data (Z) and auxiliary function values ($f(Z)$) intended for the combination are each selected randomly.
10. A data carrier according to any of claims 5 to 7, characterized in that pairs of auxiliary data (Z) and auxiliary function values ($f(Z)$) are generated by a generator without the operation ($f(Z)$) being applied to the auxiliary data (Z).
11. A data carrier according to any of claims 5 to 10, characterized in that the auxiliary data (Z) are a random number.
12. A data carrier according to any of claims 5 to 11, characterized in that the combination is an EXOR operation.
13. A data carrier according to any of the above claims, characterized in that a plurality of operations can be executed with the operating program, it holding for at least a subset of said operations that the total result achieved by execution of several operations of the subset does not depend on the order of execution of the operations, and the order of execution of the stated subset of operations is varied at least when the subset contains one or more security-relevant operations.
14. A data carrier according to claim 13, characterized in that the order of execution is varied at each run through the stated subset of operations.
15. A data carrier according to claim 13 or 14, characterized in that the order of execution is varied according to a fixed principle.

17 OCT 2002

16. A data carrier according to claim 13 or 14, characterized in that the order of execution is varied randomly.
17. A data carrier according to either of claims 13 and 14, characterized in that the order of execution is varied in accordance with the data processed with the operations.
18. A data carrier according to any of claims 13 to 17, characterized in that the order of execution is fixed before execution of the first operation of the subset for all operations of the subset whose execution is intended to be directly successive.
19. A data carrier according to any of claims 13 to 18, characterized in that it is fixed before the onset of execution of an operation of the subset which operation of the subset whose execution is intended to be successive is executed next.
20. A data carrier according to any of the above claims, characterized in that the security-relevant operations are key permutations or permutations of other secret data.
21. A data carrier according to any of the above claims, characterized in that the data carrier is a smart card.
22. A method for executing security-relevant operations in a data carrier with a semiconductor chip (5) having at least one memory in which an operating program containing a plurality of commands is stored, each command causing signals detectable from outside the semiconductor chip (5), characterized in that the data carrier performs security-relevant operations solely using operating program commands of such a kind, or using said commands in such a way, that the data processed with the corresponding commands cannot be inferred from the detected signals.
23. A method according to claim 22, characterized in that the commands used employ data present at least byte by byte.
24. A method according to either of claims 22 and 23, characterized in that the commands used do not differ, or differ very little, from each other with respect to the signal patterns caused thereby.

17 OCT 2002

25. A method according to any of claims 22 to 24, characterized in that the commands used each lead to a signal pattern which does not depend, or depends only to a very small extent, on the data processed with the command.

26. A method for protecting secret data serving as input data for one or more operations, characterized in that
    - the input data are falsified by combination with auxiliary data (Z) before execution of the one or more operations (f),
    - the output data determined by execution of the one or more operations (f) are combined with an auxiliary function value ($f(Z)$) in order to compensate the falsification of the input data,
    - whereby the auxiliary function value was previously determined by execution of the one or more operations (f) with the auxiliary data (Z) as input data in safe surroundings and stored along with the auxiliary data (Z).

27. A method according to claim 26, characterized in that the combination with the auxiliary function values ($f(Z)$) for compensating the falsification is performed at the latest directly before execution of an operation (g) which is nonlinear with respect to the compensation generating the falsification.

28. A method according to either of claims 26 and 27, characterized in that the auxiliary data (Z) are varied, the corresponding auxiliary function values ($f(Z)$) being stored in the memory of the data carrier.

29. A method according to claim 28, characterized in that new auxiliary values (Z) and new auxiliary function values ($f(Z)$) are generated by combination of two or more existing auxiliary data (Z) and auxiliary function values ($f(Z)$).

30. A method according to claim 29, characterized in that the existing auxiliary data (Z) and auxiliary function values ($f(Z)$) intended for the combination are each selected randomly.

31. A method according to any of claims 26 to 30, characterized in that pairs of auxiliary data (Z) and auxiliary function values ($f(Z)$) are generated by a generator without the operation ($f(Z)$) being applied to the auxiliary data (Z).

32. A method according to any of claims 26 to 31, characterized in that the auxiliary data (Z) are a random number.

3 OCT 2002

33. A method according to any of claims 26 to 32, characterized in that the combination is an EXOR operation.

34. A method for executing a plurality of operations within the operating system of a data carrier, it holding for at least a subset of said operations that the total result achieved by execution of several operations of the subset does not depend on the order of execution of the operations, and the order of execution of the stated subset of operations is varied at least when the subset contains one or more security-relevant operations.

35. A method according to claim 34, characterized in that the order of execution is varied at each run through the stated subset of operations.

36. A method according to claim 34 or 35, characterized in that the order of execution is varied according to a fixed principle.

37. A method according to claim 34 or 35, characterized in that the order of execution is varied randomly.

38. A method according to either of claims 34 and 35, characterized in that the order of execution is varied in accordance with the data processed with the operations.

39. A method according to any of claims 34 to 38, characterized in that the order of execution is fixed before execution of the first operation of the subset for all operations of the subset.

40. A method according to any of claims 35 to 39, characterized in that it is fixed before the onset of execution of an operation of the subset which operation of the subset whose execution is intended to be successive is executed next.

41. A method according to any of claims 22 to 40, characterized in that the security-relevant operations are key permutations or permutations of other secret data.