IDEA – Verschlüsselung

Aus der Internetseite zur IDEA – Verschlüsselung von Ascom Systec Ltd. :

IDEA™ Algorithm

Short description

IDEA™ - (**International Data Encryption Algorithm**) is the name of the new, universally applicable **block encryption** algorithm, which permits the effective protection of transmitted and stored data against unauthorized access by third parties. The fundamental criteria for the development of **IDEA™** were the highest of security requirements and easy hardware and software implementation. They predestine the algorithm, which is immediately available, for use in a great number of commercial applications.

History

The **IDEA™** algorithm was developed in a joint project involving the Swiss Federal Institute of Technology in Zürich (Dr. X. Lai / Prof. J. Massey) and Ascom. The aim of the project was to develop an encryption algorithm which would replace the **DES** procedure developed in America in the seventies.

Benefits

The **IDEA™** data encryption algorithm ...

- provides high level security not based on keeping the algroithm a secret, but rather upon ignorance of the secret key
- is fully specified and easily understood
- is available to everybody is suitable for use in a range of applications can be economically implemented in electronic components (VLSI Chip)
- can be used efficiently
- may be exported world wide.

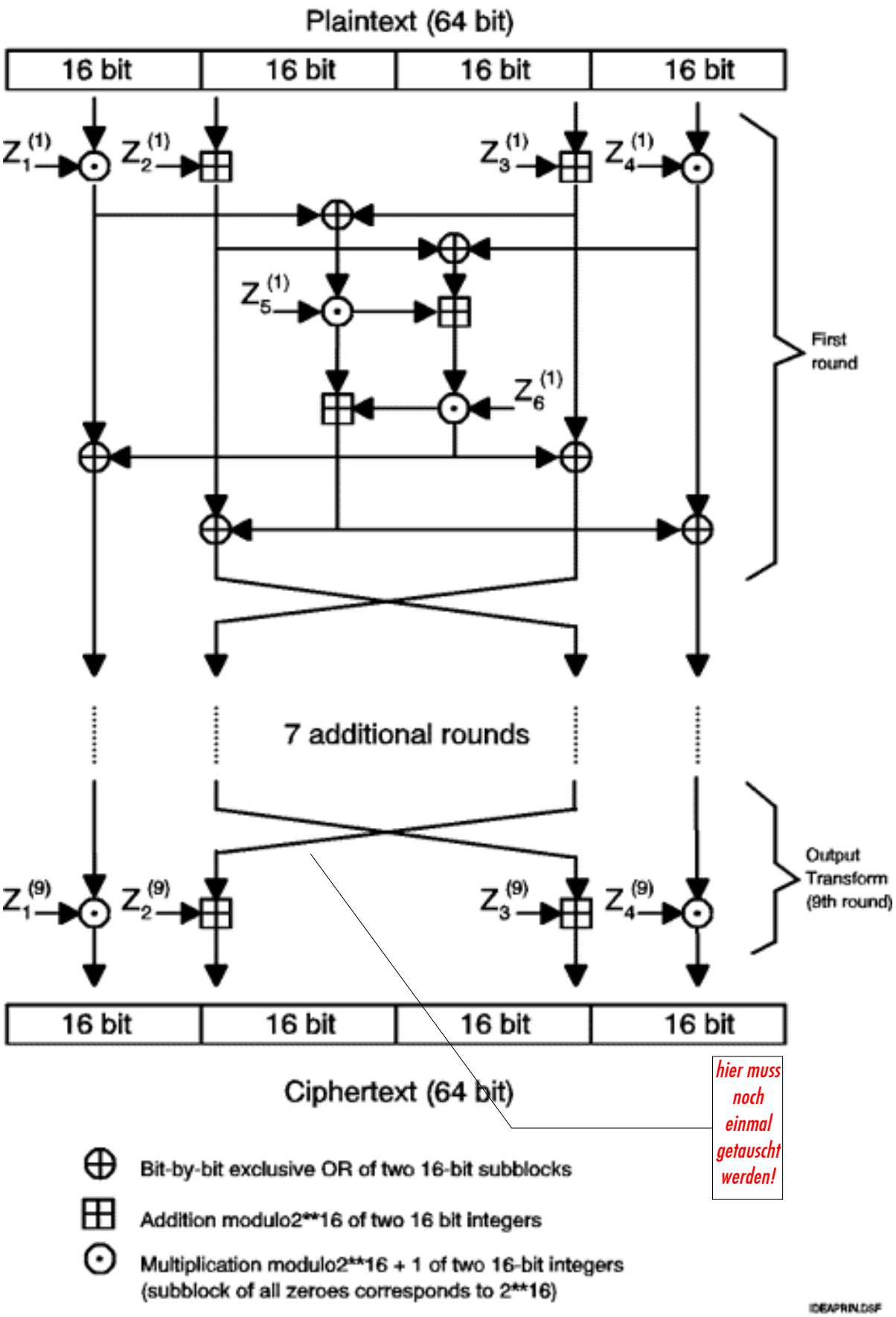
The solution

The result is the International Data Encryption Algorithm **IDEA™**, a state-of-the-art symmetrical block cipher algorithm with a 64-bit block length and a 128-bit key (twice as long as for **DES**). Implemented in a very powerful **VLSI** chip that has a ciphering capacity which clearly exeeds even the best **DES** chips, it will for the first time permit implementation of applications using the future broadband technology.

Der Verschlüsselungsvorgang ist also vollständig beschrieben und damit auch voll überprüfbar. Seine Sicherheit liegt nicht darin, dass nicht bekannt ist **wie** verschlüsselt wird, sondern in der Tatsache, **womit** verschlüsselt wird.
Das setzt voraus, dass der Schlüssel

- statistischen Analysen widersteht
- lang genug ist, um brute – force –Methoden zu widerstehen.

Die bildliche Beschreibung des Algorthmusses:


Plaintext (64 bit)
16 bit
16 bit
16 bit
16 bit
$Z_1^{(1)}$
$Z_2^{(1)}$
$Z_3^{(1)}$
$Z_4^{(1)}$
$Z_5^{(1)}$
$Z_6^{(1)}$
First round
7 additional rounds
$Z_1^{(9)}$
$Z_2^{(9)}$
$Z_3^{(9)}$
$Z_4^{(9)}$
Output Transform (9th round)
16 bit
16 bit
16 bit
16 bit
Ciphertext (64 bit)
hier muss noch einmal getauscht werden!
Bit-by-bit exclusive OR of two 16-bit subblocks
Addition modulo2**16 of two 16 bit integers
Multiplication modulo2**16 + 1 of two 16-bit integers
(subblock of all zeroes corresponds to 2**16)
IDEAPRIN.DSF

Erläuterung des Algorithmusses

Der Algorithmus wurde in ct 21'99 sehr ausführlich beschrieben. Das Verfahren ist für private Nutzung kostenlos erhältlich.
Adresse: http://www.ascom.ch/infosec/idea.html

Der Schlüssel ist ein 128 – bit Schlüssel, der blockweise mit je 64 – bit Botschaften verknüpft wird. Je 16 bit davon werden einem der vier Eingänge zugeordnet, vom Schlüssel werden beim erstenmal nur die ersten 6 als Eingaben verwendet. Die letzten beiden 16 – bit – Blöcke gehen in die zweite Runde der Verschlüsselung ein. weiterhin verwendet man weitere 128 – bit – Blöcke, die dadurch entstehen, dass die 128 bits des ursprünglichen Schlüssels um 25 bit nach links rotieren. Rotieren bedeutet, dass die vorderen 25 herausgeschobenen bits hinten wieder angehängt werden.

So entsteht aus

1 – 4	***1111 1010***	***1111 1010***	***1111 1010***	***1***111 1010
5 – 8	1111 1010	1111 1010	1111 1010	1111 1010
9 – 12	1111 1010	1111 1010	1111 1010	1111 1010
13 – 16	1111 1010	1111 1010	1111 1010	1111 1010

im ersten Schritt:

1 – 4	1111 0101	1111 0101	1111 0101	1111 0101
5 – 8	1111 0101	1111 0101	1111 0101	1111 0101
9 – 12	1111 0101	1111 0101	1111 0101	1111 0101
13 – 16	1111 010***1***	***1111 0101***	***1111 0101***	***1111 0101***

usw.

Bemerkenswert ist dabei, dass die Anzahl 25 von bits weder in einen 16 – er noch in einen 64 – er oder 128 – er Block genau hineinpasst, so dass die bit – Folgen erstaunlicherweise an unterschiedlichen Positionen der Schlüssel wieder auftauchen. Es werden nämlich der 26 – te zum ersten, dann der 51 – te, dann der 86 – te und nun käme der 141 – te. Da der Schlüssel aber nur die Länge 128 hat, müssen diese Positionen modulo 128 gerechnet werden, es ist nun also der 13 –te, dann der 38 –te usw.
Genau in diesem „Durcheinander" (siehe auch die grafische Darstellung des Datenflusses) liegt die Sicherheit des Algorithmus begründet:
Die Dateninhalte der verschlüsselten Blöcke werden in so unvorhersagbarer Weise mit einander verrechnet und dabei „durcheinander gewürfelt", dass ihre Position und Wert nicht mehr vorhersagbar wird.
So kann eine Entschlüsselung nur bei Kenntnis des Umkehrschlüssels erfolgen, den man nur dann herstellen kann, wenn man den Originalschlüssel kennt.

Umkehrschlüssel

Bei der Zuweisung der Umkehrschlüssel – also des Schlüssels, der bei der Entschlüsselung gebraucht wird – ist der Gedanke, dass jede Rechenoperation durch die zu ihr mathematisch inverse (umgekehrte) rückgängig gemacht wird.
Daher ist folgende Belegung notwendig:

Z1‘		1 / Z49
Z2‘		— Z50
Z3‘		— Z51
Z4‘		1 / Z52
Z5‘		Z47
Z6‘		Z 48
Z7‘		1 / Z43
Z8‘		— Z45
Z9‘		— Z44
Z10‘		1 / Z46
Z11‘		Z 41
Z12‘		Z 42
usw.		usw.
...		...
Z49‘		1 / Z1
Z50‘		— Z2
Z51‘		— Z3
Z52‘		1 / Z4

Kryptologie : IDEA

Beispielausgabe zur IDEA – Verschluesselung nach ct 21/99