nano-tera.ch
swiss scientific initiative in health / security / environment systems

QCrypt RTD 2010

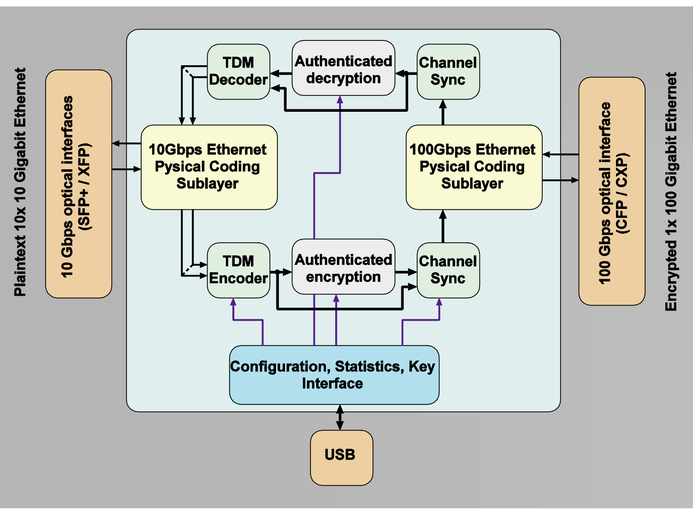
FNSNF

Comparing High-Speed Authenticated Cipher Cores for 100 Gbit/s Encryption Systems

Ch. Keller[1], M. Mühlberghuber [1], F. Gürkaynak[1], Ch. Pendl[2], N. Felber[1]

[1]IIS ETHZ, [2]TU Graz

UNIVERSITÉ DE GENÈVE · ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE · ETH Zürich · Hes·so Haute Ecole Spécialisée de Suisse occidentale · IDQ FROM VISION TO TECHNOLOGY



Evaluating alternative cryptographic algorithms

In the initial design of the Fast Encryptor, well-known cryptographic algorithms have been implemented for the authenticated encryption. The block cipher, responsible for the confidentiality of the message, was realized using AES, Advanced Encryption Standard. Authenticity was ensured using GCM, Galois Counter Mode of operation. GCM is widely used as it is officially recommended by the NIST. It is utilized in SSH, SSL/TLS and IPSec.

One of the project's goals was to evaluate alternatives for these algorithms. If for one of them, a vulnerability should be found, it can be replaced by the alternative.

For the block cipher, Serpent was chosen. Serpent is the runner-up algorithm in the AES competition. It proved to be very fast while offering a high level of security.

The alternative for GCM was chosen to be OCB, Offset Codebook Mode of Operation. OCB is a so called combined algorithm. This means that no dedicated hashing function is implemented. Instead, it uses the block cipher to calculate the checksum of the message. This property leads to considerably smaller designs, when implemented in hardware. One further advantage of OCB was the fact that is can be parallelized, which is required to reach our target throughput.

Serpent Block Cipher

Serpent was the runner-up of the AES competition. Designed for the same parameters, such as key size and block size as in AES, it rendered perfectly suitable for this project as no large changes in the data path and control unit had to be implemented.

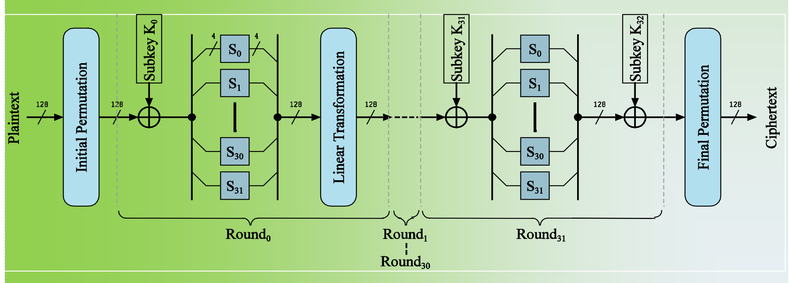
Serpent employs 32 rounds of transformation. Prior to those rounds, an initial permutation is conducted and after the rounds, a final permutation takes place.

The first 31 rounds contain a key-mixing stage, a substitution stage and a linear transformation stage. The S-Boxes, used in the substitution stage has eight variations. In each round, only one variation is used. Every eight rounds, the same S-Boxes are instantiated. In the key-mixing stage, the corresponding round key is XORed with the current state.

In the last round, the linear transformation is replaced by an additional key-mixing stage.

The main differences between AES and Serpent are the higher number of rounds, the smaller S-Boxes and the linear transformation.

The S-Boxes have been implemented using lookup tables. Since they have only 4 bits of input and 4 bits of output, they could easily be implemented using FPGA logic instead of ROM-based lookup tables as in AES.



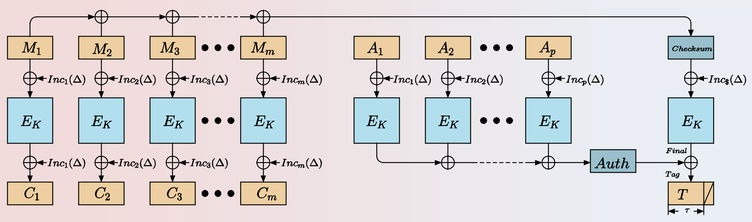
Offset Codebook Mode

The Offset Codebook Mode (OCB) is a so called combined authenticated encryption scheme. It has been first published in 2001 by Rogaway et al. and been updated twice since.
The message is being cut into blocks. The size of these blocks equals the block size of the underlying block cipher.
For the encryption, the message block is XORed with a Δ-value. The output of the block cipher is then again XORed with the same Δ-value. The Δ-value is incremented with each message block.

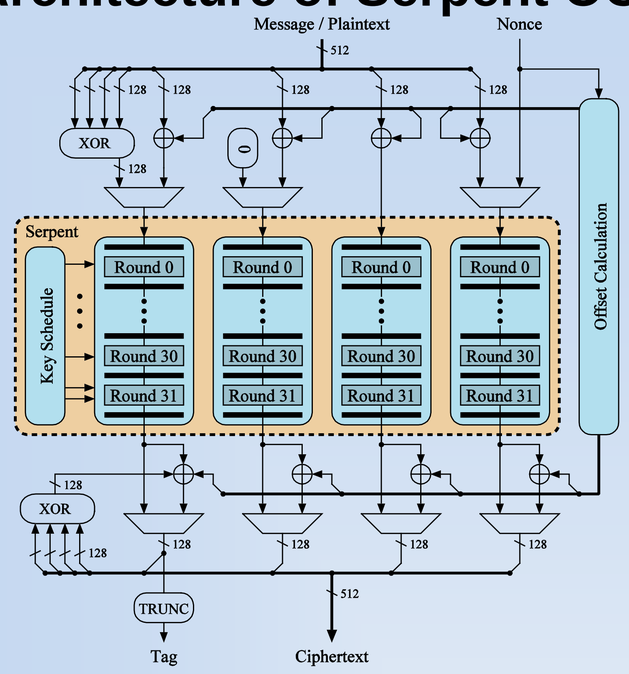
To calculate the authentication tag, we have to distinguish between encrypted data and data that is only authenticated, i.e. the header of an Ethernet frame. For the encrypted data, the not yet encrypted message blocks (M_1 – M_m) are XORed together to build a *checksum* (see fig). This checksum is then XORed with the final $\Delta_\$$ and encrypted using the block cipher. The message blocks which are only authenticated (A_1 – A_p), are XORed with a Δ, then encrypted and XORed into the *Auth* checksum.
To calculate the final tag, both the *Auth* checksum and the encrypted checksum are XORed into the authentication tag.

This scheme is very suitable for our purposes as it can be easily parallelized. Only the for the Δ-values, there exists a dependency on the previous value.

For the decryption, we require block cipher cores in both encryption and decryption mode, since e.g. for the authenticated data, encryption is still required to calculate *Auth*.



Architecture of Serpent OCB



To reach our goal of 100 Gbit/s of throughput, we instantiated four Serpent cores. These cores are fully unrolled. Further, after each round, pipeline registers have been inserted.
The Offset calculation (Δ-value) had to be adapted to generate four values per cycle.
Since the messages we are using in our project cannot be larger than a standard Ethernet frame (1500 bytes of payload) we could reduce the overall complexity. We could further reduce complexity as we do not require any padding. All our messages have a length of a multiple of the block size.
Compared with GCM, OCB has a higher latency. In our GCM architecture, the latency is only one cycle, while in Serpent OCB it is 34 cycles.

Results

The four resulting architectures have been compiled, placed and routed for our target device. With the exception of M9K memory blocks, no Altera-specific logic blocks have been used.
For a better comparison, two cipher-only designs have been compiled as well.

We used as fully unrolled and pipelined architecture of both block ciphers, meaning each round is present in hardware. To reach the target throughput of 100 Gbit/s, each Cipher core is present four times.

Due to the fact that in Serpent, the S-Boxes were realized in logic, while in the AES architecture, they were implemented in SRAM memory, AES is considerably smaller. Further more, the higher number of rounds in Serpent contributes to a four times larger area consumption.

In terms of speed, the two block ciphers perform almost equally well. They reach a throughput of 144 Gbit/s and 137 Gbit/s respectively.

When comparing OCB and GCM, we can see that the GCM architecture requires more than twice the area of the OCB architecture. This is mainly due to the fact that OCB does not require a separate hashing core.

In terms of speed, OCB is performing considerably better than GCM. The more complex logic of the GHash cannot be realized with a delay as low as the one of OCB. Serpent OCB, reaching a throughput of **141 Gbit/s**, is 33% faster than our reference implementation AES GCM.

Our Serpent OCB implementation is the fastest design that has been published so far.

Block Cipher	Mode of Operation	Area		f_{max}	Throughput	
		ALMs	M9K Bl	MHz	Gbit/s	%
Cipher-Only Architectures						
Serpent	cipher-only	28,399	0	281	144	136
AES	cipher-only	7,661	314	267	137	130
Authenticated Encryption Architectures						
Serpent	OCB	38,312	0	275	141	133
AES	OCB	11,948	314	250	128	121
Serpent	GCM	56,474	0	203	104	99
AES	GCM	24,313	314	206	105	100

Note: there are patents on OCB. The inventor has recently eased licensing for OCB which might lead to a higher acceptance.

Principal Investigator: Nicolas Gisin (Nicolas.Gisin@unige.ch)

Contact: Norbert Felber (felber@iis.ee.ethz.ch)