Cagri Önal, Hubert Kirrmann, ISPCS 2012 San Francisco, 27 September 2012

Security improvements for IEEE 1588 Annex K

Implementation and comparison of authentication codes

Content

- Introduction
- Overview of IEEE 1588 Annex K
  - Is it sufficient?
  - Is it necessary?
- An improved minimizing solution
- Authentication algorithms
  - Design structure
  - On-the-fly calculation
  - Comparison criteria and findings
- Conclusion

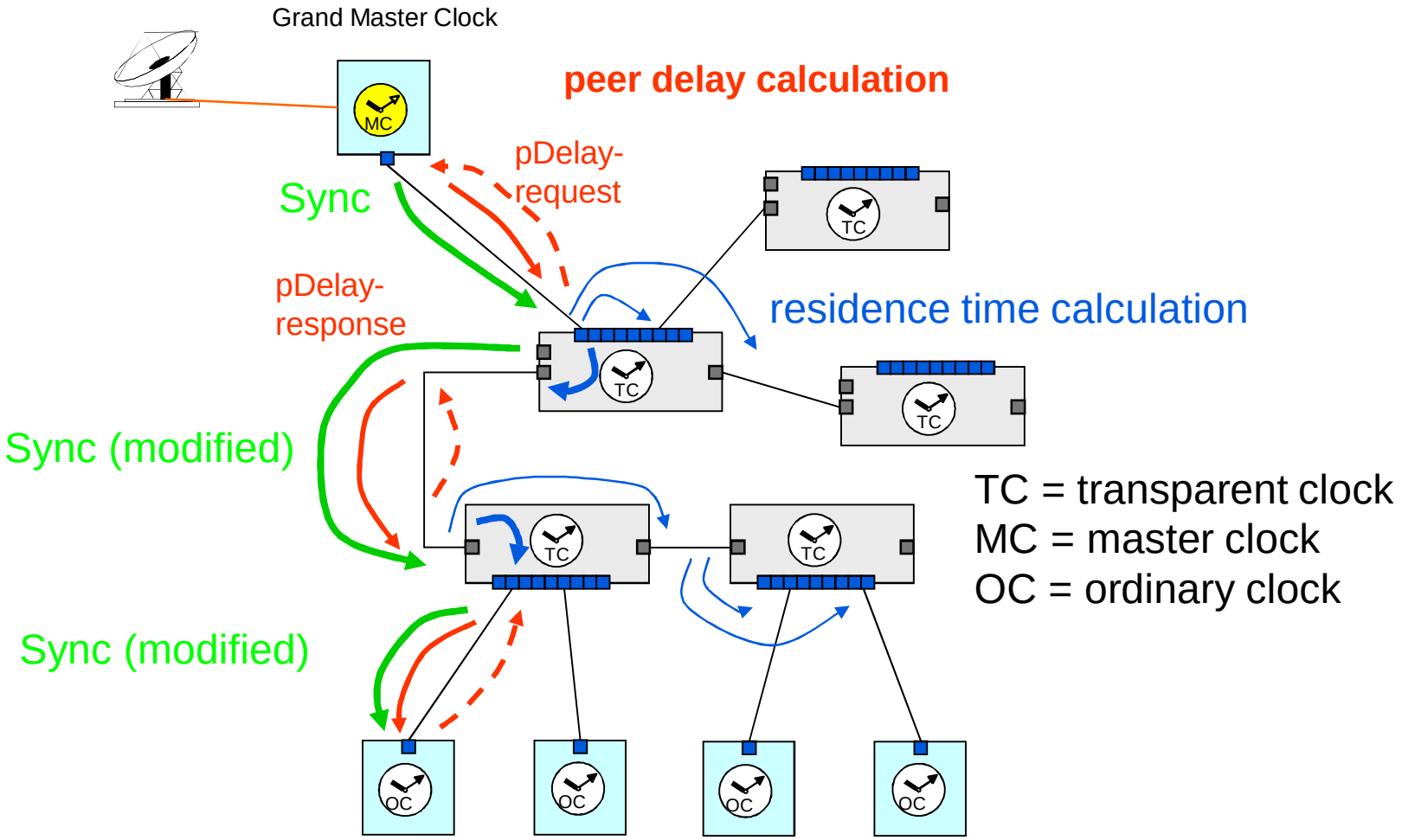
Introduction

IEEE 1588 Annex K

- An informative annex of IEEE1588v2 which presents a security mechanism for PTP,
  - which proposes HMAC (hash-based) as authentication algorithm.
  - which checks authenticity of the sources with a 3-way handshake.
  - which does not define key distribution scheme.
  - which does not mention “on the fly” calculation of MACs.

Why is the last point important?

Introduction - Challenge in PTP security



- Transparent clocks modify one-step Sync messages “on-the-fly”.
- Changes apply to the correction field, to the FCS and to the MAC.

Content

- Introduction
- Overview of IEEE 1588 Annex K
  - Is it sufficient?
  - Is it necessary?
- An improved minimizing solution
- Authentication algorithms
  - Design structure
  - On-the-fly calculation
  - Comparison Criteria and Findings
- Conclusion

Is IEEE 1588 Annex K sufficient to secure PTP?

- Sufficiency of a security mechanism is determined by asking whether it satisfies the rules of cryptographic integrity.

  - **Integrity:** Prevention of Message Tampering and Replaying by the "Man-in-the-middle"

    -> satisfied by the authentication algorithm (HMAC) and replay check mechanism.

  - **Membership Control:** Identification of the secure sources and the attackers in order to alarm the network management to take actions when needed

    -> Challenge-response mechanism guarantees that no messages are further processed before a secure 3-way handshake is realized.

Answer: YES, provided that a secure key distribution is defined.

Is Annex K necessary to secure PTP?

- The algorithm HMAC brings no advantage over the other MACs, then why not another MAC?
- Challenge-response mechanism brings only overhead, but no additional security.
  One-way registration and authentication of the sources can do the same required function.

Answer: NO, it is in fact overkill due to the additional traffic introduced by the 3-way handshake.

Number of nodes	Plain [KBytes]	Secured [KBytes]	load increase (by factor)	Number of messages
10	0,43	8	19	65
20	0,86	31	36	230
40	1,72	122	71	860
60	2,58	271	105	1890
100	4,30	749	174	5150
500	21,48	18590	865	125750
1024	44,00	77896	1770	525824

Reference: A. Treytl and B.Hirschler, Practical Application of 1588 Security, ISPCS 2008

An improved minimizing solution

- **Idea:**
  - Remove and/or update what makes it overkill
  - Simplify the requirements and definitions of security mechanisms (replay check, security association, etc.)

- **Actions:**
  - Remove 3-way challenge-response mechanism and related parameters
  - Simplify the security association and TLV structures accordingly
  - Replace the authentication algorithm with a more efficient one that allows on-the-fly computation

Content

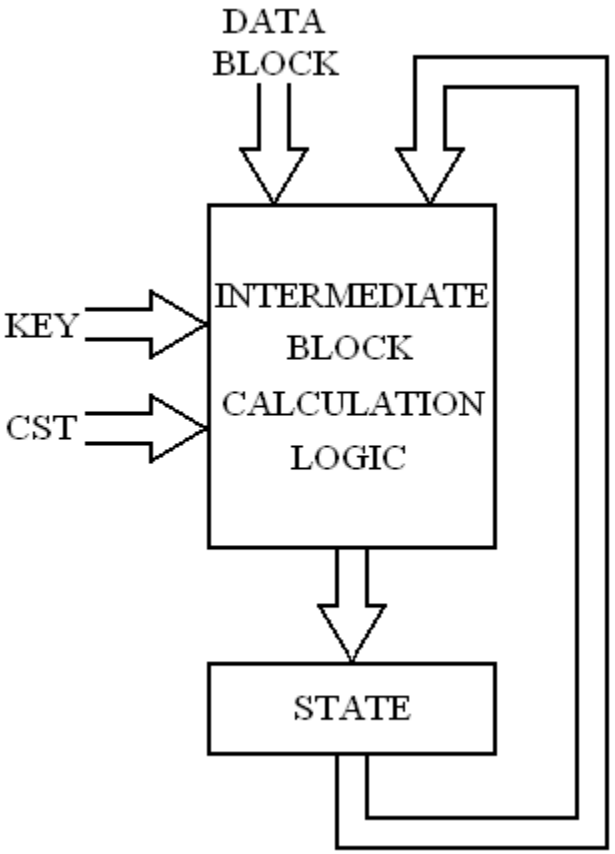
- Introduction
- Overview of IEEE 1588 Annex K
  - Is it sufficient?
  - Is it necessary?
- An improved minimizing solution
- Authentication algorithms
  - Design structure
  - On-the-fly calculation
  - Comparison Criteria and Findings
- Conclusion

Authentication Algorithms

- 4 well-known candidate MAC algorithms are compared :
  - HMAC-SHA256 (hash-based) 512-bit blocks
  - GMAC-128 (Galois-field multiplier-based) 128-bit blocks
  - XCBCMAC-AES128 (cipher-based) 128-bit blocks
  - CMAC-AES128 (cipher-based) 128-bit blocks
- All MACs are working with block based processing

Architectural point of view

- What does block-based processing mean?



- Input data is divided into fixed size blocks
- Each block is processed sequentially
- The results are accumulated to the final authentication code
- The design efficiency is determined by:
  - **number of resources** used
  - **time** consumed (**latency**)

    in the INTERMEDIATE BLOCK CALCULATION LOGIC

State of the art

- Input data whose authentication code to be calculated are fully received or prepared, before the authentication code calculation may start, which causes the accumulation of block delays.
  - for long frames, time delay might be critical, since each step takes 0,16 µs

    for a 134-byte frame => 1.5 µs *

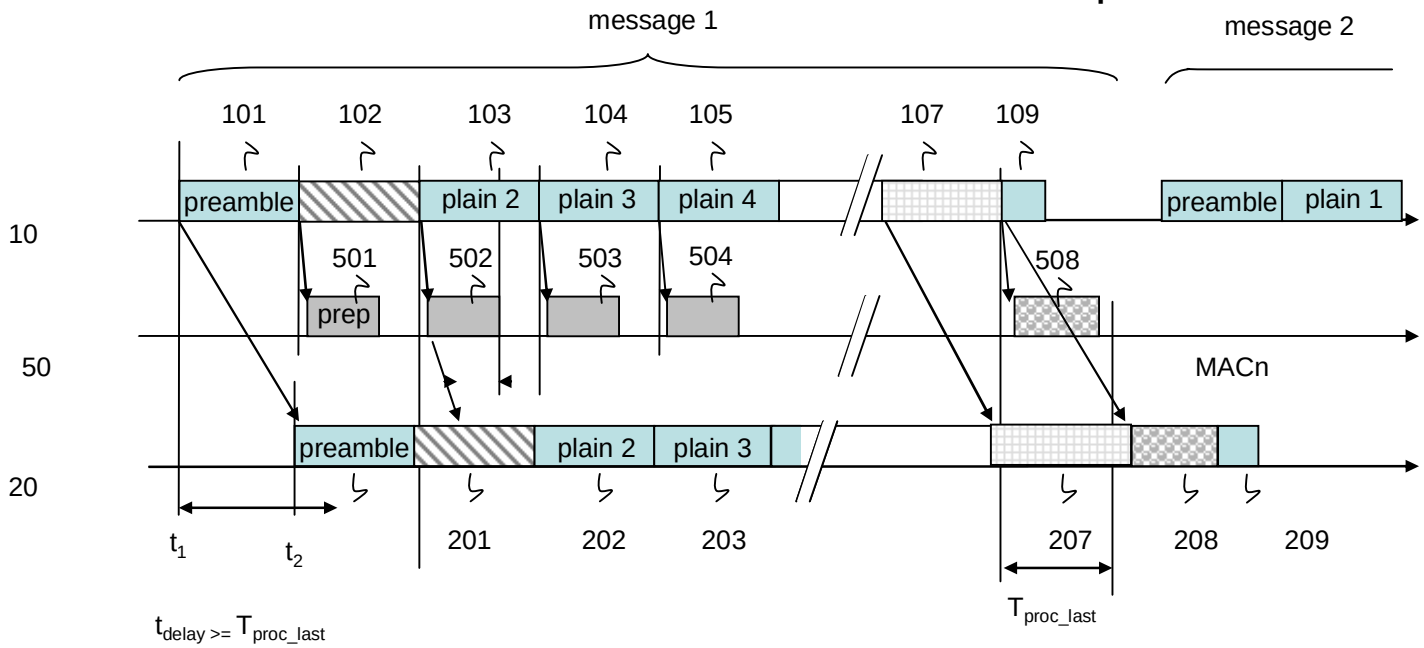
    for a 1000-byte frame => 10 µs * with XCBCMAC
- Time delay is crucial for intermediate nodes such as IEDs behaving as bridges (HSR) or transparent clocks applying IEEE 1588.

- So another method is needed to **decrease (and bound)** the latency

* assuming a reference implementation detailed later

On-the fly calculation of authentication codes

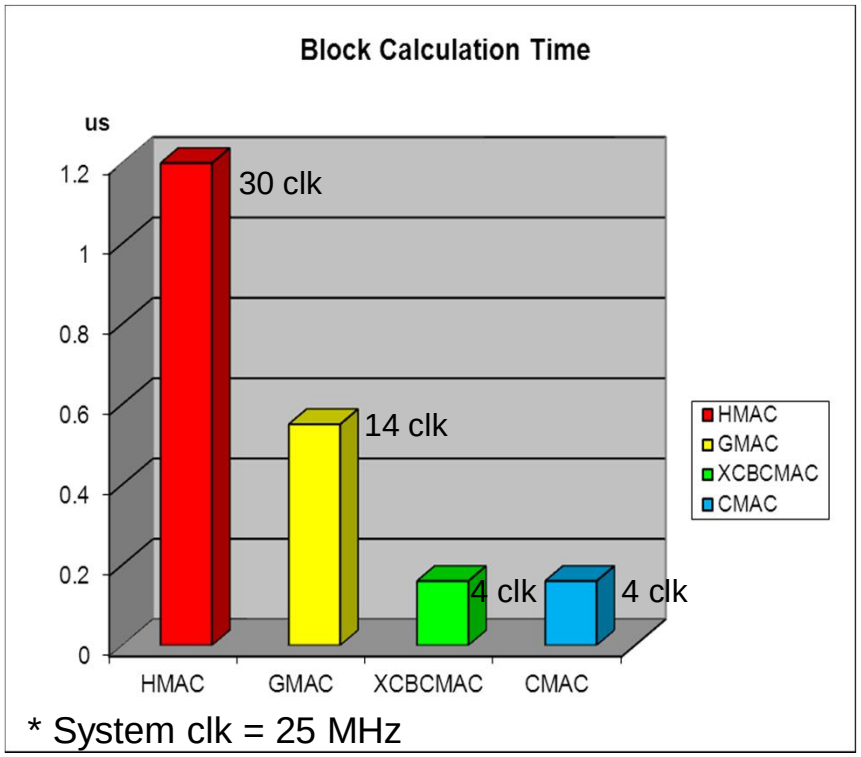
- On the fly authentication calculation taking advantage of block-based method => no accumulation of intermediate block calculation latency.
- The overall latency is just composed of two block calculation latencies (last block latency + padded block latency)
  - Bound and independent of input size
- The requirement: **1 block acquisition time > 1 block calculation time**
- The trick: to hide block calculation time under block acquisition time



Block calculation time comparison:

- Cipher based MACs outperform the others in block calculation time.
- BCT shows the amount of time consumed in the intermediate block calculation logic of each MAC implementation.

Figures for Altera Cyclone III



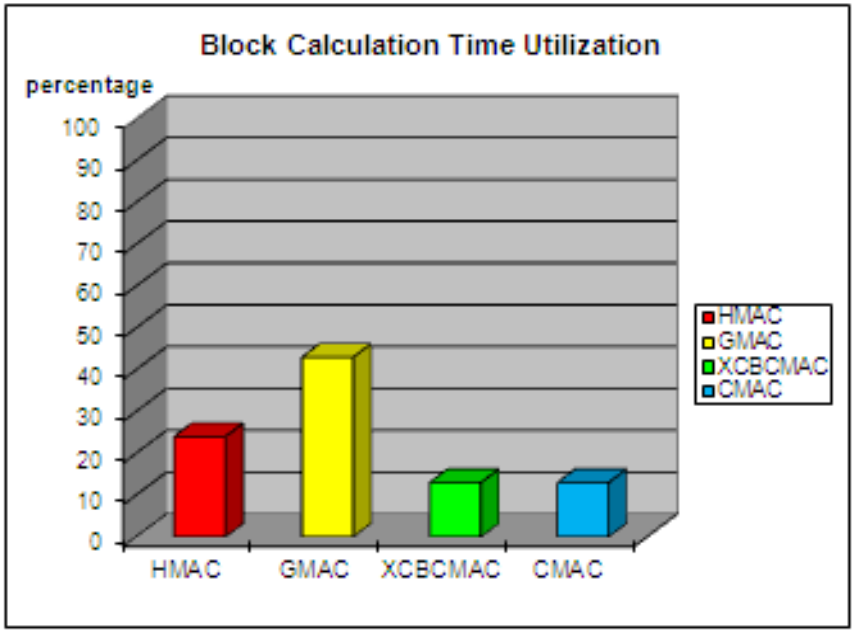
* System clk = 25 MHz

Block calculation time hiding:

- For network adaptability, what matters is the ratio of the time spent for block calculation and the time to acquire one block of data. Due to different processing block sizes of the algorithms, their acquisition takes also different amount of time.

- All implementations are suitable for 100 Mbit/s networks (utilization < 100%). The cipher-based CMAC and XCBCMAC (12,5%) are nearly compatible with 1 Gbit/s. But since the block size of GMAC is smaller than that of HMAC, it is executed more often, leading to a worse utilization.

Figures for Altera Cyclone III



In the figure:

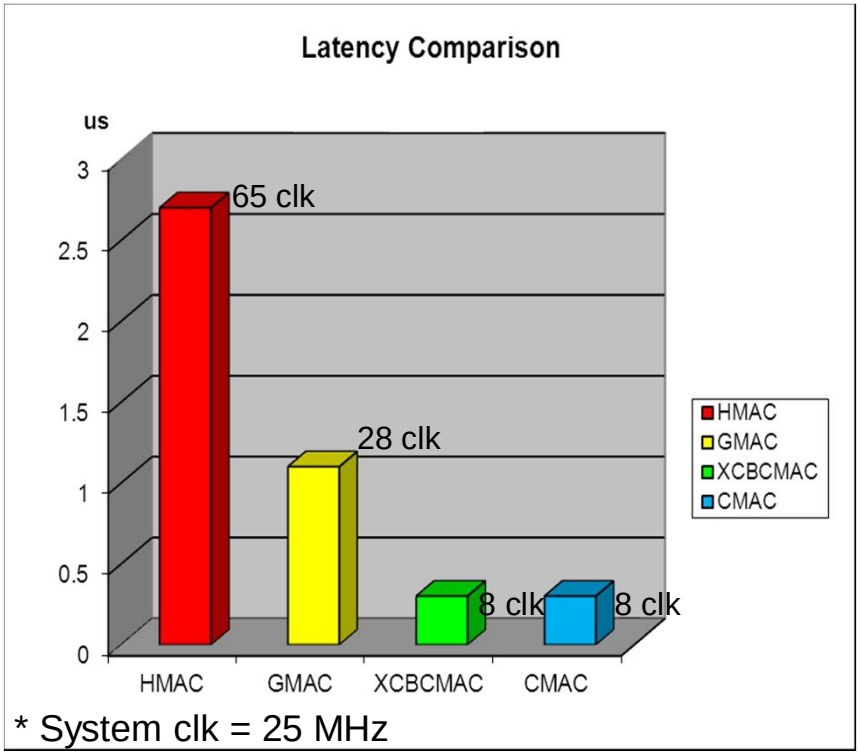
- 100% line defines upper limit for utilization of the algorithms to be compatible with 100 Mbit/s.
- 10% line similarly defines the upper limit for 1 Gbit/s networks.

Latency comparison:

- On average, this latency is equal to 2-3 block calculation time spent for the last block (and possibly the padded block)

- Then, this result can be obtained from the Block calculation time comparison. XCBCMAC and CMAC offer the shortest input-output latency.

Figures for Altera Cyclone III



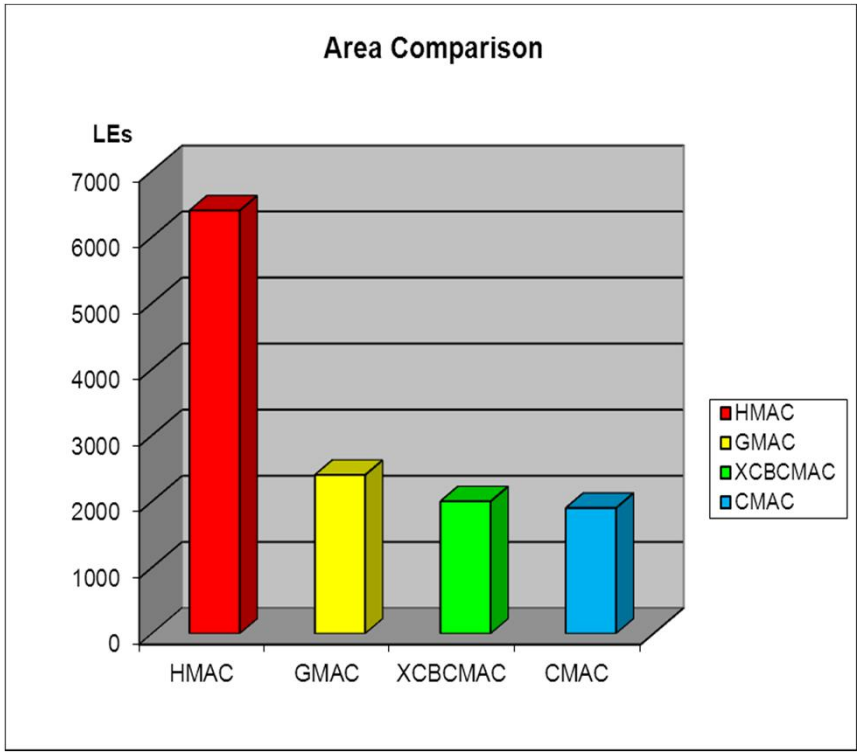
* System clk = 25 MHz

FPGA area comparison:

- CMAC outperforms the other algorithms in area cost. This result is due to the simplicity and the small block size.

- The smaller the block size is, the lesser the number of combinational functions in the design. HMAC uses 512-bit blocks which makes it the most inefficient.

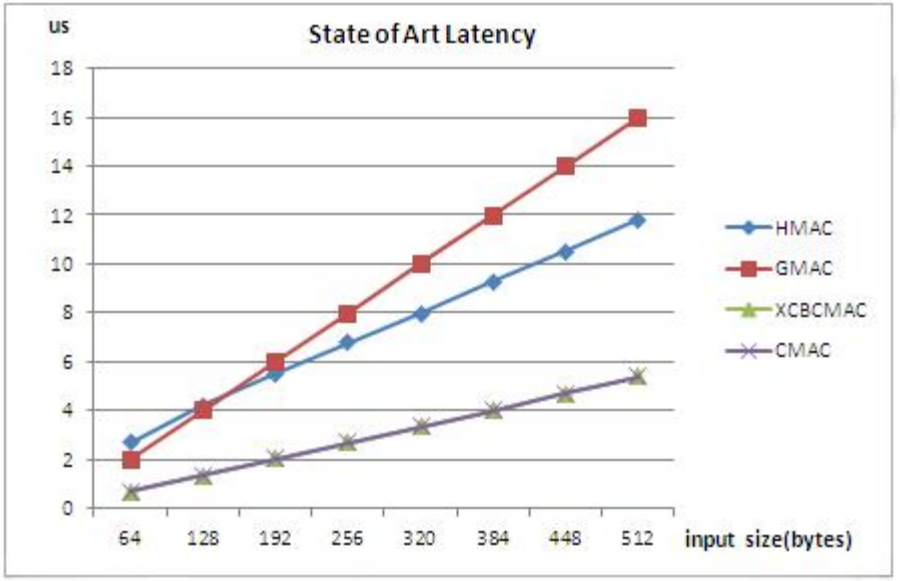
Figures for Altera Cyclone III



State-of-the-art (fully buffered) latency

- Even if the implementations are done with the traditional method, cipher-based MACs outperforms the others.
- The state-of-the-art solution can not bind latency if the frame size is large. It depends on the number of blocks.

Figures for Altera Cyclone III



	HMAC	GMAC	XCBCMAC	CMAC
State of Art Time Complx.	**0.3 n***	**0.55 n**	**0.16 n**	**0.16 n**

* n = # of 128-bit (16-byte) input blocks

Summary of comparison

	HMAC	GMAC	XCBCMAC	CMAC
Security bits*	**256**	128	128	128
Key Length	512	128	128	128
Block Size	512	128	128	128
Block Calculation Logic	SHA-256 (hash)	Galois Field Multiplier	AES-128	AES-128
BCT**	1.2 µs (30 clks***)	0.55 µs (14 clks)	**0.16 µs (4 clks)**	**0.16 µs (4 clks)**
BCT Utilization**	24%	43%	**12.5%**	**12.5%**
State of Art Time Complx.	0.3 n*****	0.55 n	**0.16 n**	**0.16 n**
On the Fly Calc. Latency	2.6 µs	1.1 µs	**0.3 µs**	**0.3 µs**
Area Cost	6400 LEs****	2400 LEs	2000 LEs	**1900 LEs**

** BCT: Block Calculation Time *** clock frequency = 25 MHz

**** LEs = Logical Elements ***** n = # of 128-bit input blocks

Is only 1588 affected ?

It makes no sense to change Annex K alone, without considering other layer-2 protocols.

Indeed, we can expect that in the future, the network interface will implement in hardware the redundancy (HSR), the clock synchronization (1588), the time-critical sampled value (SV) transmission (IEC 61859-9-2) and the GOOSE transmission of event-triggered values (IEC 61850-8-1).

It is out of question to give each protocol its own authentication scheme, this would result in a waste of the FPGA.

Therefore, IEEE 1588 would be well advised to synchronize with the work done in IEC 62351 and IEC TC57 WG10.

Conclusion

- The three-way handshake foreseen in IEEE 1588 brings only additional generated payload and traffic on the network, but no additional security.
- A key distribution system should be specified.
- The MAC algorithm proposed in Annex K is sub-optimal.
  - CMAC offers a better solution that allows on-the-fly calculation of the MAC at practically no loss in performance.
- A unified layer 2 authentication should be standardized for all protocols.

Thanks for your attention!

Any questions?

(or do you want to see more about unification of security mechanisms?)

Unification of security mechanisms

- **Different** algorithms proposed in **different** standards to secure **different** protocols, no unique security mechanism exists yet

	Authentication	Encryption
IPSec	**HMAC-SHA1** **AES-XCBC-MAC-96**	**AES-CBC-128**
TLS	**HMAC-SHA1** **HMAC-SHA256**	**RC4-128** **AES-CBC-128**
IEC 61850-90-5	**HMAC-SHA256** **AES-GMAC**	**AES-128-GCM**
IEC 62351-6	**HMAC-SHA256**	**NA**
IEEE 1588 Annex K	**HMAC-SHA1** **HMAC-SHA256**	**NA**

- Multiple security mechanisms in one node => excess of resource usage
  => increasing latency
  => increasing complexity

Unification of security mechanisms

- Idea is to choose the best suite and implement this security engine in a

  place that is common to every protocol => level 2

- Every frame processed in link layer can have a security check, no need to have another one in upper layers

- The protocols identified when the frame is on the fly, necessary authentication realized accordingly

- The mechanism is transparent => Upper layers can only receive secure and decrypted frames => decreased utilization of resources

Proposal

- Multilevel security (on layer 2 , above layer 3 and at application level is a waste of computing power.
- If the communication between communication interface and the application is trusted, only a layer 2 security is needed.
- If the messages cross routers (at layer 3) they should have a layer 3 authentication between the routers (routers are trusted entities)
- A layer 2 authentication can use the same algorithm as IPsec, therefore only one security engine is needed.
- CMAC seems to offer today the best performance / complexity / resilence and allows to conciliate layer 2 and layer 3 security.

Power and productivity
for a better world™