TriviA: A Fast and Secure Authenticated Encryption Scheme

Avik Chakraborti, Anupam Chattopadhyay, Muhammad Hassan, Mridul Nandi

September 15, 2015

Outline of the talk

1. Introduction.
2. Underlying Mathematical Components.
3. Specification of TriviA
4. Security Theorems and Security Bounds
5. Properties of TriviA
6. Hardware Results for TriviA

Authenticated Encryption (AE)

Why AE?

- Privacy of Plaintext.
- Authenticity of the plaintext/ ciphertext and associated data.

More Formally....

- Tagged-encryption : AE.enc : $\mathcal{M} \times \mathcal{D} \times \mathcal{N} \times \mathcal{K} \rightarrow \mathcal{C}$
- Verified-decryption : AE.dec : $\mathcal{C} \times \mathcal{D} \times \mathcal{N} \times \mathcal{K} \rightarrow \mathcal{M} \cup \perp$

Stream Cipher

Formally

- Encrypts in *bit level*.

- Key stream $K = KeyGen(MK, N, |M|)$

- M, C and K are *bitstreams*.

- $C_i = Enc_{K_i}(M_i) = (K_i + M_i) \mod 2$
- $M_i = Dec_{K_i}(C_i) = (K_i + C_i) \mod 2$

Popular Ciphers : Trivium, Grain, Salsa etc.

ϵ-ΔU-(Universal) Hash

Formally

- $h : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$
- $\forall \delta \in \mathcal{R}, \forall x \neq x' \in \mathcal{D}$, $\Pr_{K \in_r \mathcal{K}}[h(K; x) - h(K; x') = \delta] \leq \epsilon$

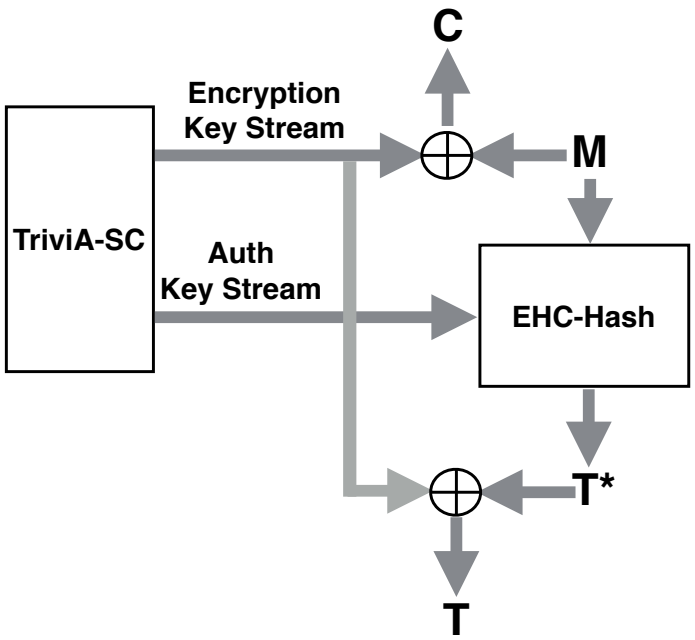
Examples

- Multilinear Hash (ML), Pseudo Dot Product Hash (PDP), Toeplitz Hash.

Universal Hash with Minimum Multiplications

- Encode-Hash-Combine (EHC).

TriviA Encryption Mode

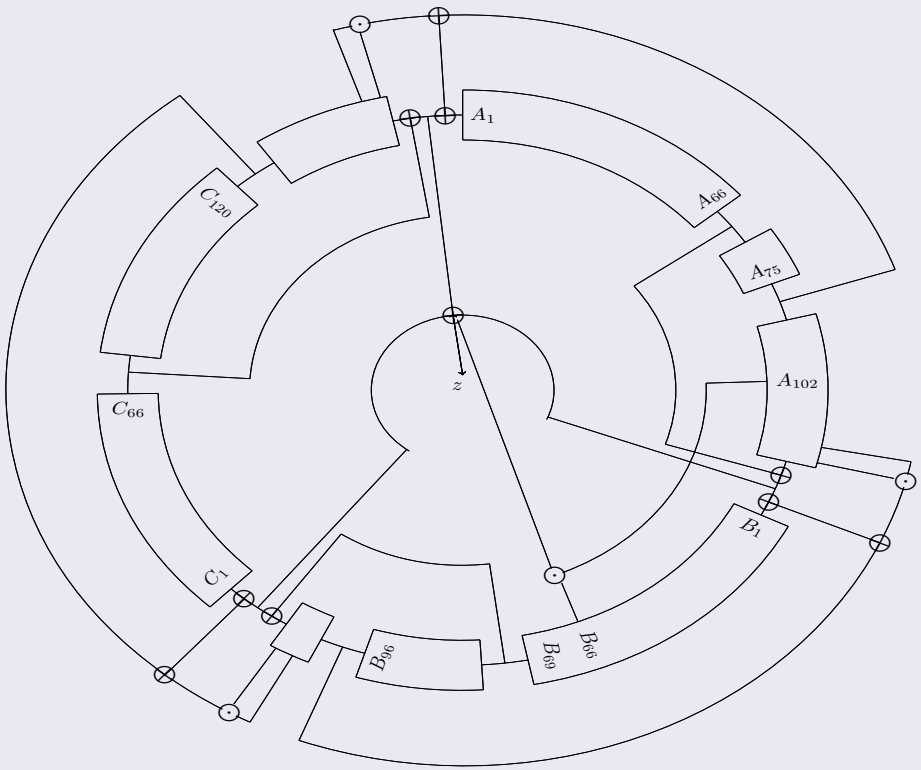


- TriviA-SC - Updated version of Trivium.
- EHC-Hash - Universal Hash follows EHC technique.
- TriviA-SC generates
  - *Encryption* key stream
  - *Authentication* key stream

  parallely

A Trivium Based Stream Cipher : TriviA-SC

TriviA-SC Circuit

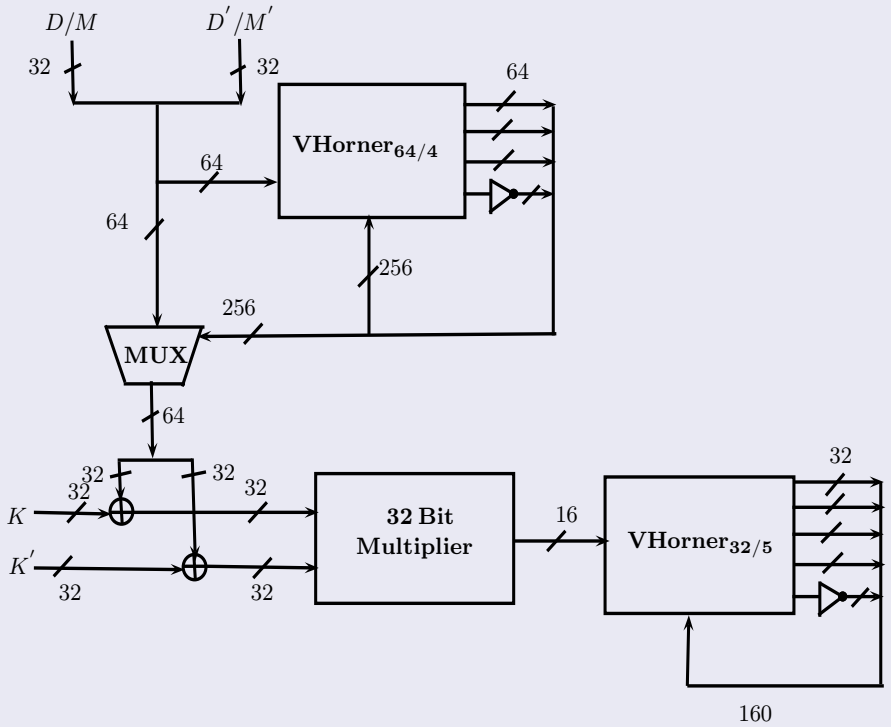


TriviA-SC Informations

- 384-bit state - A (132-bit), B (105-bit) and C (147-bit)
- Load 128-bit key and 128-bit nonce, 1152-round init
- 64-bit parallelism (KeyExt64 and Update64)
- *Nonlinearity* in the output
- KeyExt64 - From output, StExt64 - From state

Circuit of EHC Hash

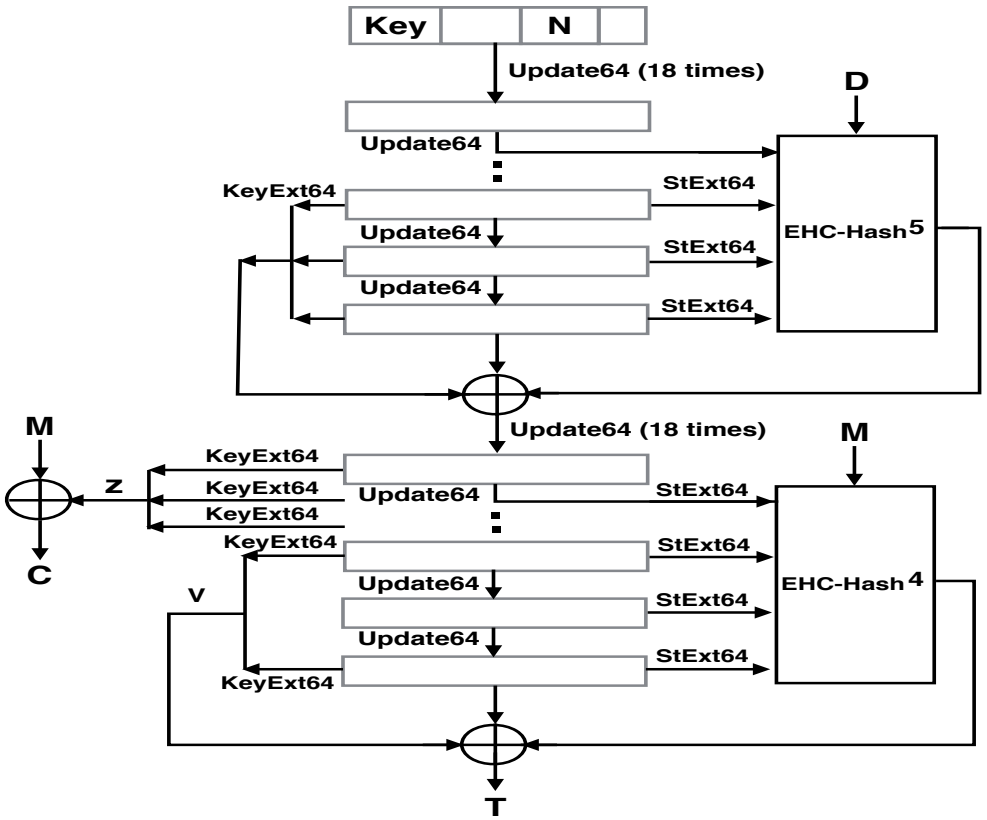
EHC Circuit



EHC-Hash Informations

- Underlying Fields - $\mathbb{F}_{2^{32}}$ (α) and $\mathbb{F}_{2^{64}}$ (β)
- *Expand*/*Encode*-*Hash*-*Combine*
  - *Encode*(*Expand*) by ECCode_d $(\mathrm{VHorner}_{64/d})$
  - *Blockwise Hash* by PDP-Hash (32-bit Multiplier)
  - *Combine* by $\mathrm{VMult}_{\alpha,d}$ $(\mathrm{VHorner}_{32/d+1})$
- One 32-bit Multiplication for 64-bit block.
- EHC is 2^{-128}-Δ-U hash

Lower Level Structure of TriviA



Informations on TriviA

- Updated to the *CAESAR* second round
- *Arbitrary* length M (padded with 10^*) divided into 64-bit *Blocks*
- Intermediate tag (if any) - Computed after each ck blocks.
  - $ck = 0$ for this Paper (no intermediate tag).
  - $ck \in \{0, 128\}$ for CAESAR submission.
- $|C| = |M|$, $|T| = 128$

Privacy Bound for TriviA

Theorem

Let A be a relaxed nonce-respecting adversary which makes at most q encryption queries. Moreover we assume that A can make at most 2^{32} queries with a same nonce. Then, $\mathbf{Adv}^{\mathrm{priv}}_{TriviA}(A) \leq \frac{q}{2^{128}}$.

Authenticity Bound for TriviA

Theorem

Let A be a relaxed nonce-respecting adversary which makes at most q queries such that nonce can repeat up to 2^{32} times. In addition, A is making at most q_f forging attempt. If the stream cipher Trivia-SC is perfectly secure then

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathit{TriviA}}(A) \leq \frac{q}{2^{128}} + \frac{q_f}{2^{124}}.$$

Security Level for TriviA

Security Bounds

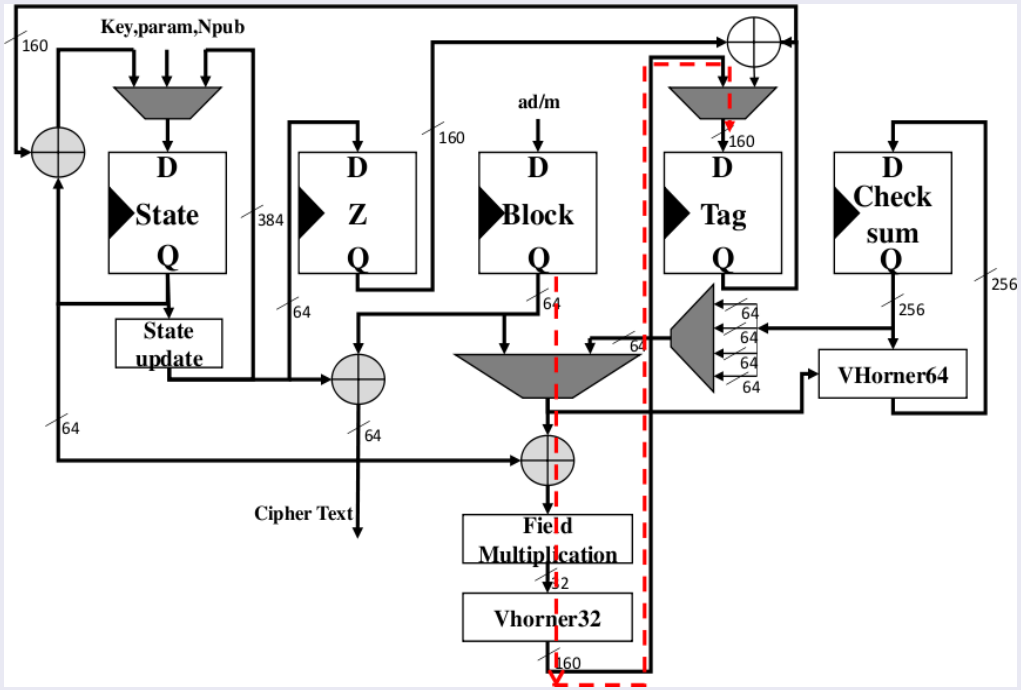
Version	Confdentiality	Authenticity
TriviA-0	128	124
TriviA-128	128	124

Important Properties of TriviA

- Options for Intermediate Tag.
- TriviA-SC - Updated design of a well studied and efficient (both in hardware and software) stream cipher Trivium.
- High security level- 128-bits for confidentiality and 124-bits for Authenticity of plaintext.
- High speed hardware.

TriviA-Base Architecture

TriviA-Base

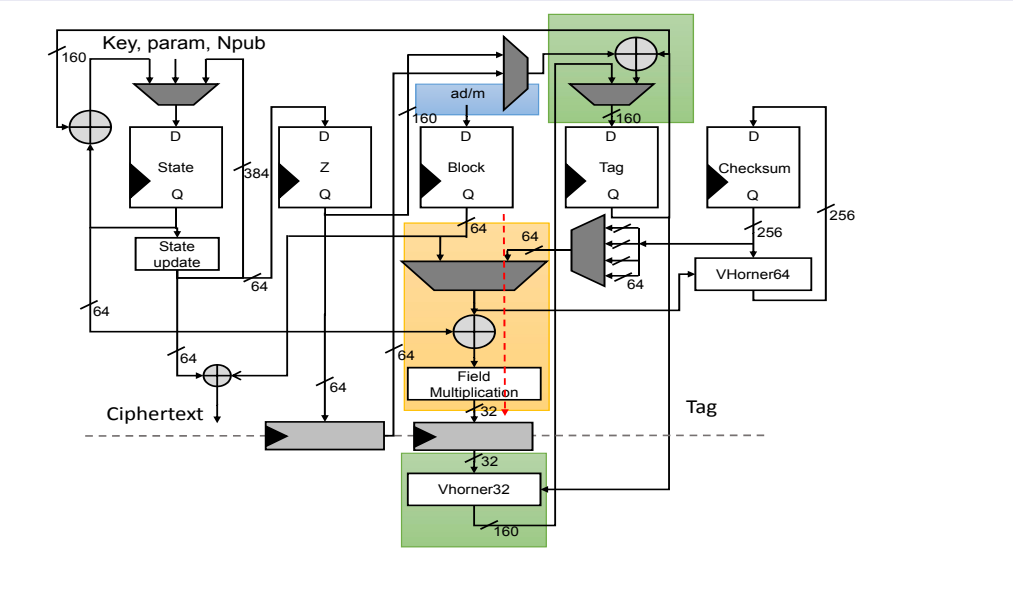


TriviA-Base Architecture Properties

- No *pipelined* register
- *Parallel* processing of data
- Processes 64-bits/ cycle
- *Long Critical path* : (2×1) 64-bit *MUX* $\rightarrow$ 64-bit *XOR* $\rightarrow$ 32-bit *Mult* $\rightarrow$ *Tag* Updation $\rightarrow$ (3×1) 160-bit *MUX*
- Reduced Speed, Throughput

TriviA-Pipelined Architecture

TriviA-Pipelined



TriviA-Pipelined Architecture Properties

- 2 operations in series
  - 32-bit multiplication
  - *Tag* updation
- *Shorter Critical path* : (2×1) 64-bit *MUX* $\rightarrow$ 64-bit *XOR* $\rightarrow$ 32-bit *Mult*
- 3 stage pipeline : Increased *throughput*, *frequency*
- 3 extra clock cycles (*Blue*, *Orange* and *Green* blocks)

TriviA ASIC Implementation

- Verilog HDL, Synopsys Design Compiler J-2014.09
- Technology node: UMC 65nm logic SP/RVT Low-K process

- Base Implementation
  - Area : 23.6 KGE
  - Frequency : 1150 MHZ, Throughput : 73.9 Gbps

- Pipelined Implementation
  - Area : 24.4 KGE
  - Frequency : 1425 MHZ, Throughput : 91.2 Gbps

Comparison with Other Results

AE Schemes		ASIC Implementation			Cycles/ Byte (cpb)
		Area (KGE)	Throughput (Gbps)	Efficiency (Mbps/ GE)	
TriviA Base		23.6	73.9	3.13	0.12
TriviA Pipelined		24.4	91.2	**3.73**	0.12
Scream, iScream		17.29	5.19	0.30	-
NORX		62	28.2	0.45	-
Ascon		7.95	7.77	0.98	0.75
AEGIS	*AO1*	20.55	1.35	0.07	6.67
	AO2	60.88	37.44	0.61	0.33
	TO1	88.91	53.55	0.60	0.20
	TO2	172.72	121.07	0.70	0.07

TriviA FPGA Results

- Xilinx ISE 14.7
- Default settings, no optimizations
- Pre-layout synthesis
- 5.4x better (in terms of area efficiency) of than *AES-CCM*

TriviA FPGA Results Comparison

Xilinx FPGA Platform	AES-CCM			TriviA-*Base*			TriviA-Pipelined
	# Slices	*Gbps*	*Area–Efficiency (Mbps/Slice)*	*# Slices*	*Gbps*	*Area–Efficiency (Mbps/Slice)*	*Area–Efficiency (Mbps/Slice)*
Spartan-6 -3	272	>0.57	2.09	815	7.6	9.3	11.29
Virtex-5 -3	343	>0.78	2.27	637	11.7	18.3	20.3
Virtex-6 -3	295	>0.87	2.95	725	16	22	25
Kintex-7 -3	296	>1	3.38	714	16.89	23.65	24.31
Virtex-7 -3	296	>1	3.38	714	16.89	23.65	24.31

Conclusion

- *SC* and *PI* hash based *AE*
- Achieves *high* provable security bound
- Well Studied *SC* and *PI* hash needs *minimum* multiplication
- *High speed* AE and *high area-efficiency*

Thank you