PRIMATEs: Round 2

Elena Andreeva Begül Bilgin Andrey Bogdanov
Atul Luykx Florian Mendel Bart Mennink
Nicky Mouha Qingju Wang Kan Yasuda

KU Leuven, iMinds, UTwente, DTU, TU Graz, Inria, SHJT, NTT

DIAC 2015 — September 29, 2015

PRIMATEs: Quick Recap

http://primates.ae/

PRIMATEs

- Lightweight, permutation-based AE
- Efficient threshold implementation

Three PRIMATEs

- HANUMAN: ideal permutation
- GIBBON: trade-off speed/security
- APE: misuse resistance

No Second-Round Tweaks

Outline

Introduction

- AE Security Notions

AE With Limited Storage

- APE solution
- GIBBON solution

Hardware Implementation

- FPGA
- ASIC

Typical AE Scheme

- To get IND-CCA: combine IND-CPA and INT-CTXT
  (Bellare and Namprempre, Asiacrypt '00)
- Invalid ciphertext? Return $\perp$!

Typical AE Scheme

- To get IND-CCA: combine IND-CPA and INT-CTXT
  (Bellare and Namprempre, Asiacrypt '00)
- Invalid ciphertext? Return $\perp$!

Can Fail in Practice?

- Short tags
- Unverified plaintext

Typical AE Scheme

- To get IND-CCA: combine IND-CPA and INT-CTXT
  (Bellare and Namprempre, Asiacrypt '00)
- Invalid ciphertext? Return $\perp$!

Can Fail in Practice?

- Short tags
- Unverified plaintext

Achieve IND-CCA Without INT-CTXT?

- Ciphertext difference must affect entire message
  $\Rightarrow$ no online encryption!

Crypto Component

- Contains secret key
- Needs to perform AE
- Cannot store entire message/ciphertext

Crypto Component

- Contains secret key
- Needs to perform AE
- Cannot store entire message/ciphertext

Solution by Fouque et al. (SAC '03)

- Initialize PRNG
- "Decrypt anyway", but mask with PRNG output
- If tag correct: reveal PRNG seed, otherwise $\perp$

Limited Storage Scenario

Crypto Component

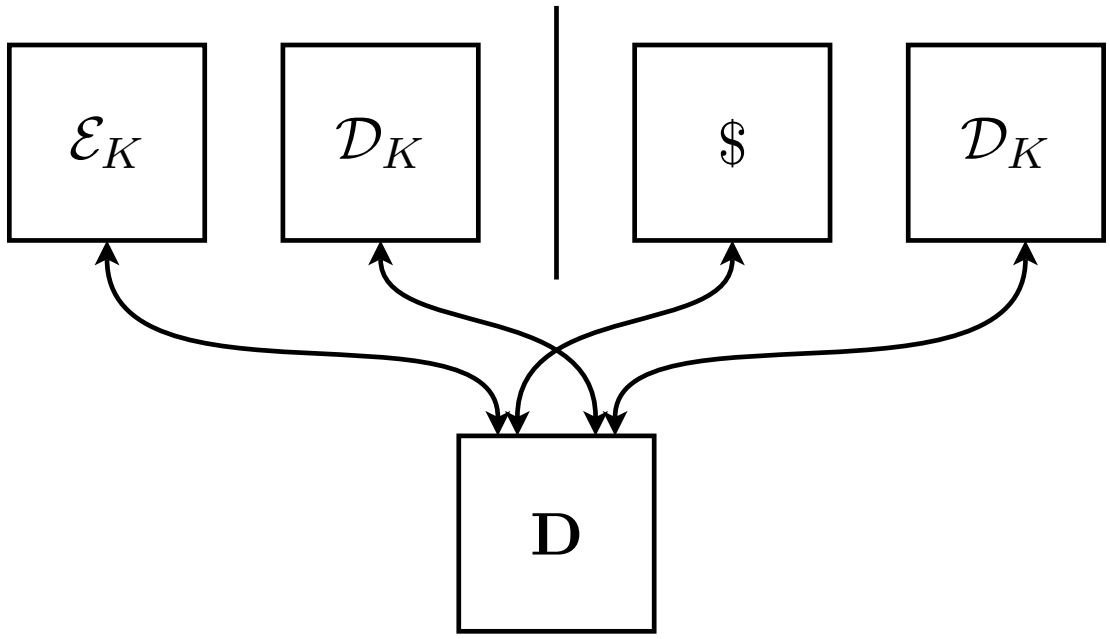
- Contains secret key
- Needs to perform AE
- Cannot store entire message/ciphertext

Solution by Fouque et al. (SAC '03)

- Initialize PRNG
- "Decrypt anyway", but mask with PRNG output
- If tag correct: reveal PRNG seed, otherwise $\perp$

More Efficient Solutions?

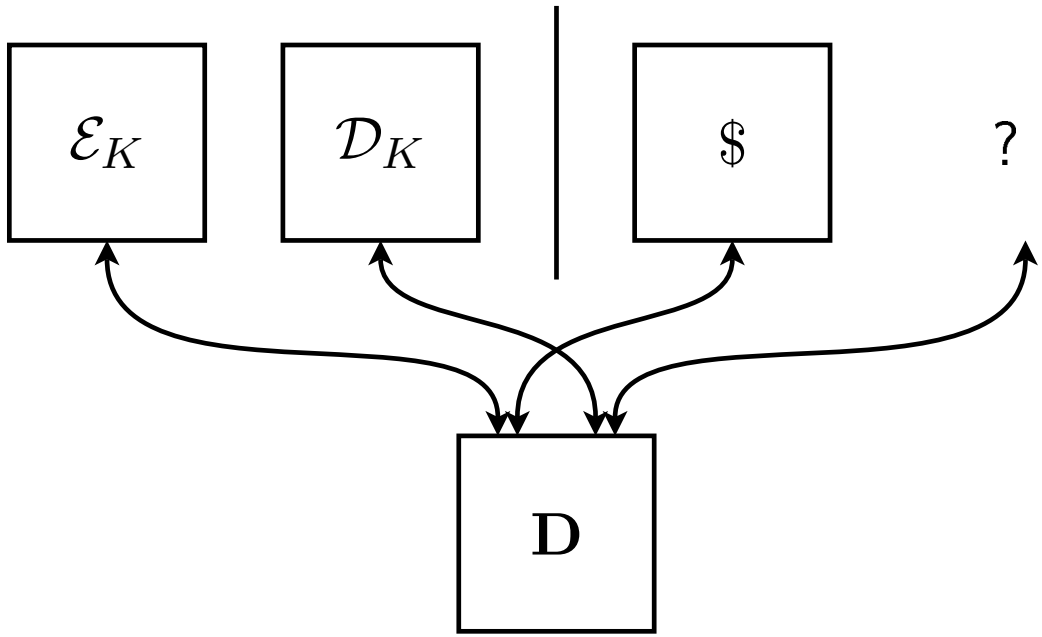
- APE
- GIBBON



Security Notion

- IND-CCA: decryption with K in both worlds

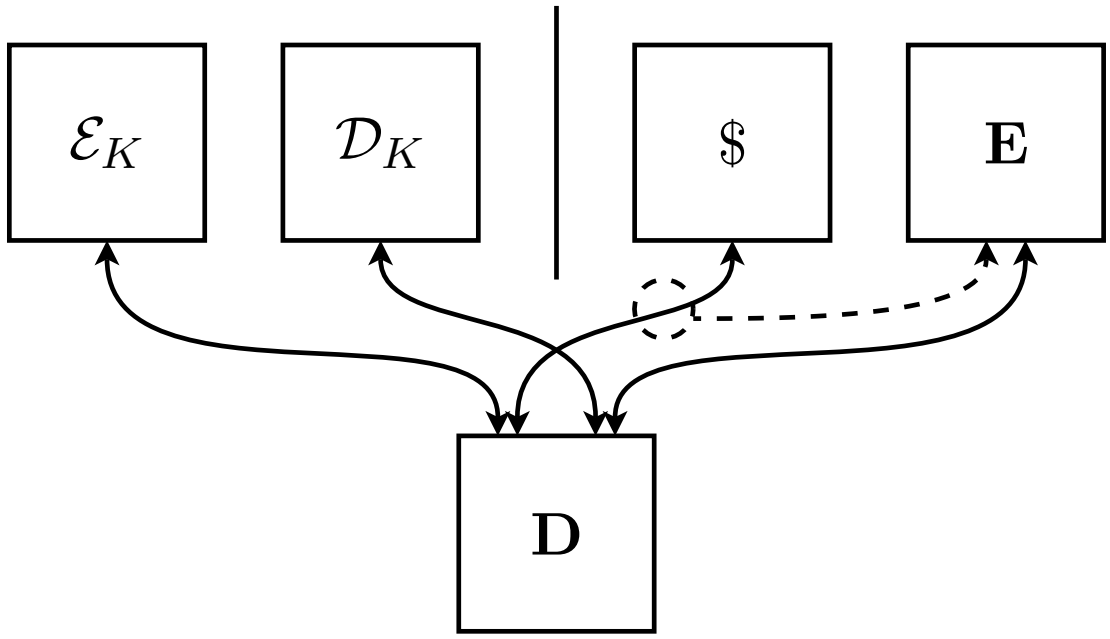
IND-CCA: Too Strong?



Security Notion

- No decryption with K in ideal world?
- Syntax error!

IND-CPA + PA1



Security Notion

- Introduce simulator: Plaintext Awareness (PA)
- IND-CPA + PA1: level between IND-CPA and IND-CCA
- Andreeva et al. (Asiacrypt '14)

On-line Schemes Achieving IND-CPA + PA1

- Random IV: CTR, (Delayed-)CBC,...
- Nonce/Arbitrary IV: APE

Understanding IND-CPA + PA1

- Not "randomize unverified plaintext"
  (unverified plaintext can have structure)
- Not "skip verification"
  (but attacker can see unverified plaintext)
- To achieve IND-CCA: IND-CPA + PA2

On-line Schemes Achieving IND-CPA + PA1

- Random IV: CTR, (Delayed-)CBC,...
- Nonce/Arbitrary IV: APE

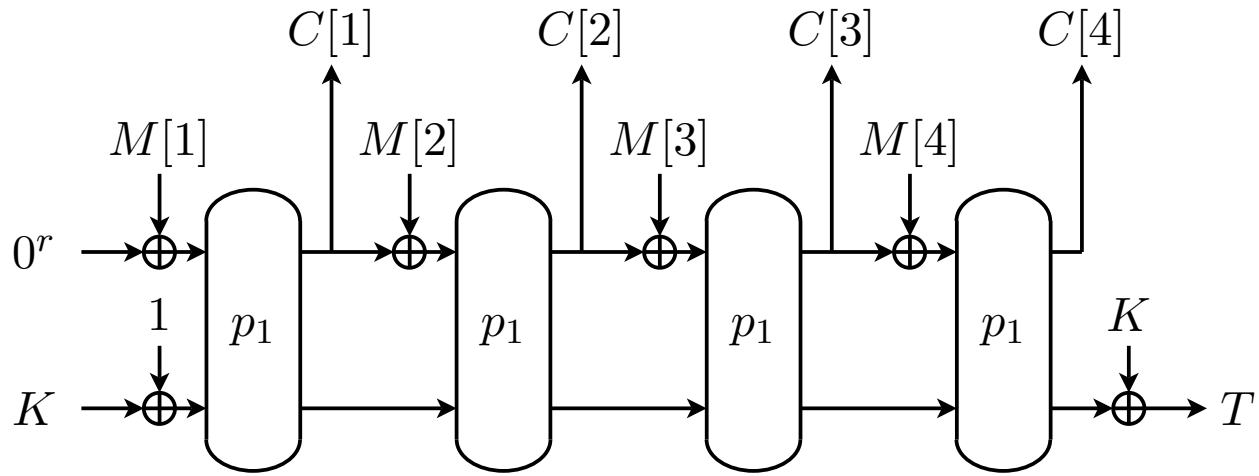
Understanding IND-CPA + PA1

- Not "randomize unverified plaintext"
  (unverified plaintext can have structure)
- Not "skip verification"
  (but attacker can see unverified plaintext)
- To achieve IND-CCA: IND-CPA + PA2

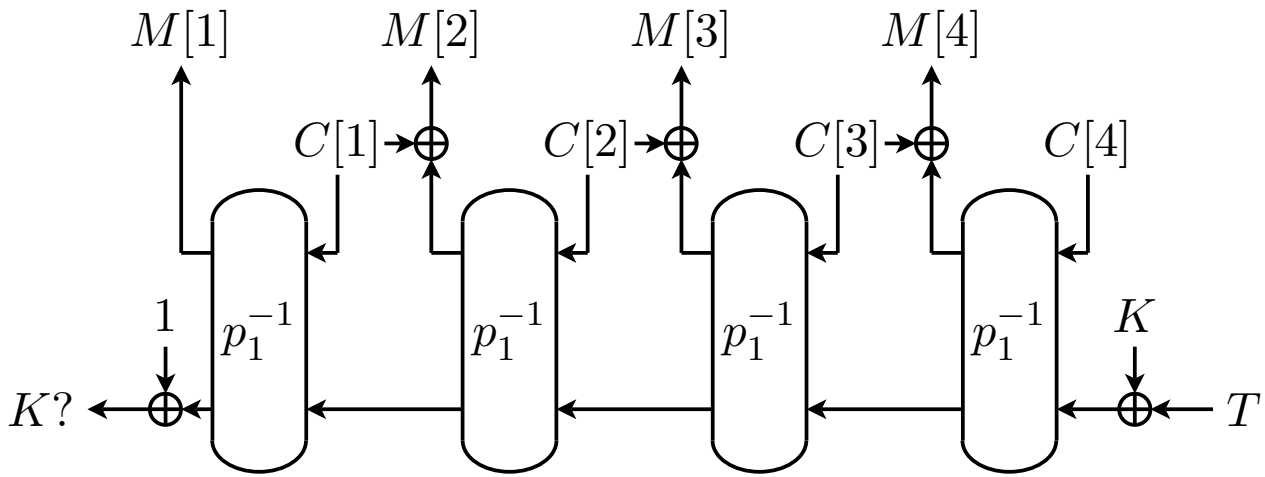
Integrity under release of unverified plaintext?

- INT-RUP (next session)

Encrypt



Decrypt and Verify



Hoang et al. (CRYPTO '15)

- Meaning of online misuse resistance
- Guess plaintext block-by-block
- Or even byte-by-byte (variant of BEAST attack)

Better Understanding of APE

Hoang et al. (CRYPTO '15)

- Meaning of online misuse resistance
- Guess plaintext block-by-block
- Or even byte-by-byte (variant of BEAST attack)

Saha et al.

- Fault attack using unverified plaintext
- EscApe (Indocrypt '14): requires repeating IV
- Scope (SAC '15): no IV requirement (yesterday morning)

Agrawal et al. (ACISP '15) (this afternoon)

- Crypto component does verification
- Tag correct? Reveals internal state
- Receiver can decrypt only one message

  (no other messages, no forgery, no key-recovery)

Agrawal et al. (ACISP '15) (this afternoon)

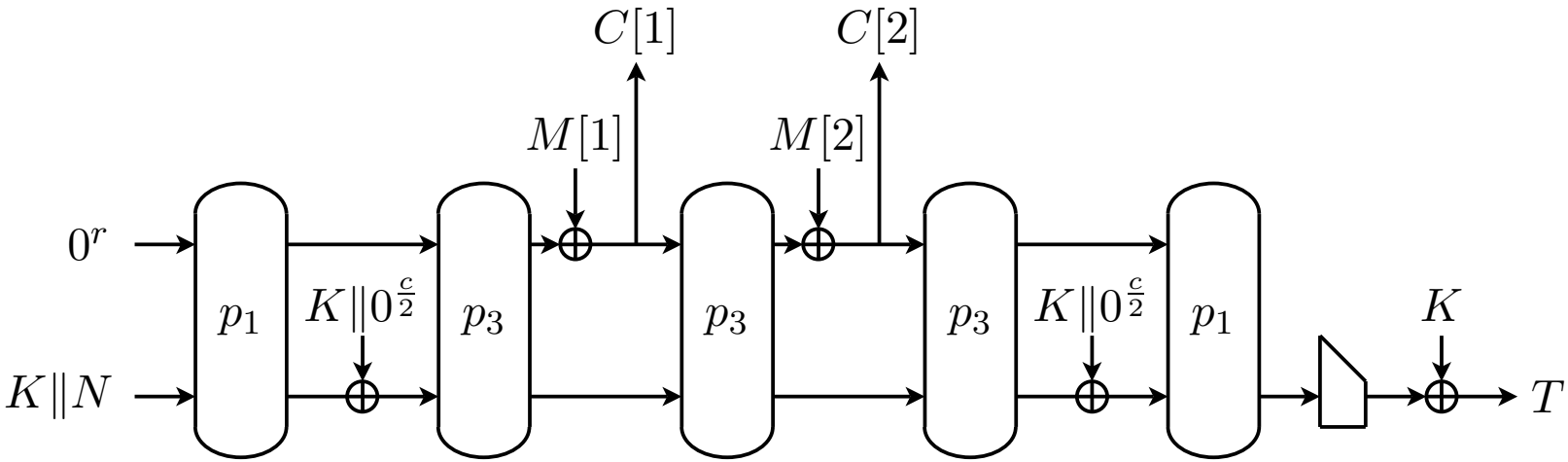
- Crypto component does verification
- Tag correct? Reveals internal state
- Receiver can decrypt only one message

  (no other messages, no forgery, no key-recovery)

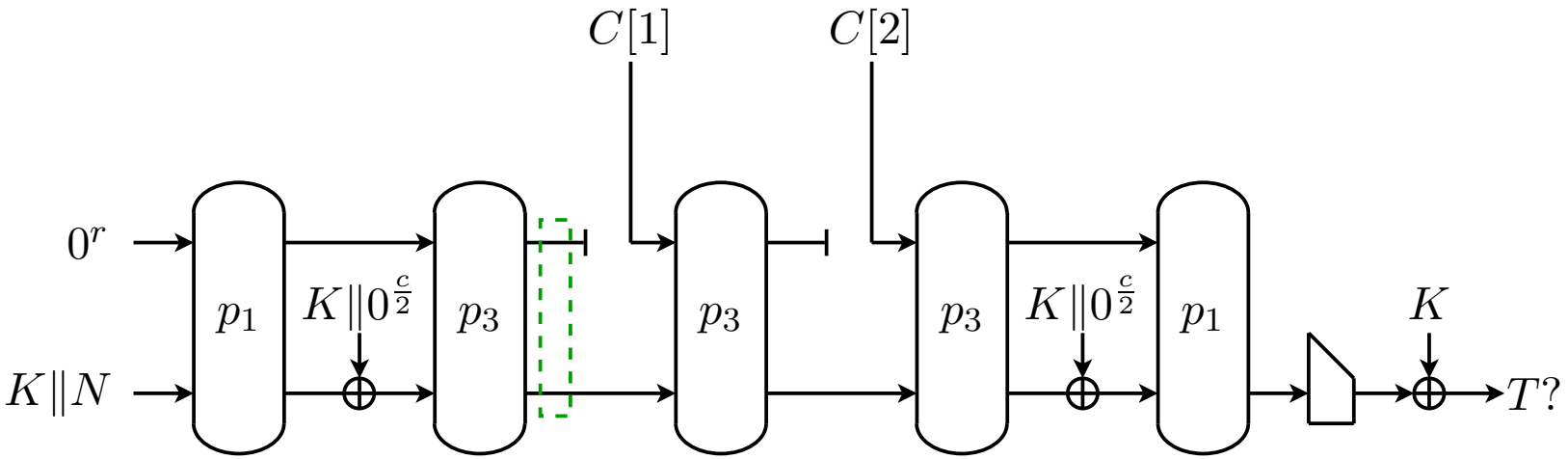
Supported Schemes

- GIBBON
- ASCON
- sp-AELM and variants

Encrypt



Verify (and Decrypt)



PRIMATEs-80 Permutation

- ASIC: NanGate 45nm, Faraday 130/180 nm
- HW/SW co-design with 16-bit MSP430
- FPGA: Xilinx Spartan-6

Implementation

- Thanks to Danilo Šijačić
- Threshold implementation: to follow!

PRIMATEs-P on ASIC

http://primates.ae/

PRIMATEs Permutation (200-bit)				
Library	Data (bits)	Latency (cycles)	C. Path (ns)	Area (GE)
NanGate 45 nm	200	1	1.85	3240.00
NanGate 45 nm	25	9	1.19	1947.33
NanGate 45 nm	25	16	1.28	**1680.66**
NanGate 45 nm	25	41	0.30	2115.66
Faraday 130 nm	25	9	2.95	1885.25
Faraday 180 nm	25	9	2.62	1551.25
Faraday 130 nm	25	16	3.16	1579.25
Faraday 180 nm	25	16	2.33	**1297.25**

PRIMATEs-P on FPGA

http://primates.ae/

PRIMATEs-P (200-bit), 25-bit data, 16 cycles	
Critical Path [ns]	2.36
Maximum Frequency [MHz]	423.55
Maximum Throughput [Mbps]	5293.75
Throughput @100 kHz [kbps]	1250.00

Device Utilization Summary (XC6SLX45-3CSG324)		
Slice Logic Utilization	Used	Available
Number of Slice Registers	69	54576
Number used as FF	69	
Number of Slice LUTs	95	27288
Number used as logic	68	
Number used as shift reg.	26	
Number of occupied slices	**26**	6822

Conclusion

http://primates.ae/

PRIMATEs: Round 2

- No tweaks

AE with Limited Storage

- Motivation: small crypto component
- Solutions: APE, GIBBON

Primates-P (200-bit) in Hardware

- Faraday 180 nm: 1297.25 GE
- Xilinx Spartan-6: 26 slices

Questions?