Towards Practical Whitebox Cryptography: Optimizing Efficiency and Space Hardness

Andrey Bogdanov, Takanori Isobe and Elmar Tischhauser
DTU and Sony

Hanoi, Vietnam
Asiacrypt'16
5 December 2016

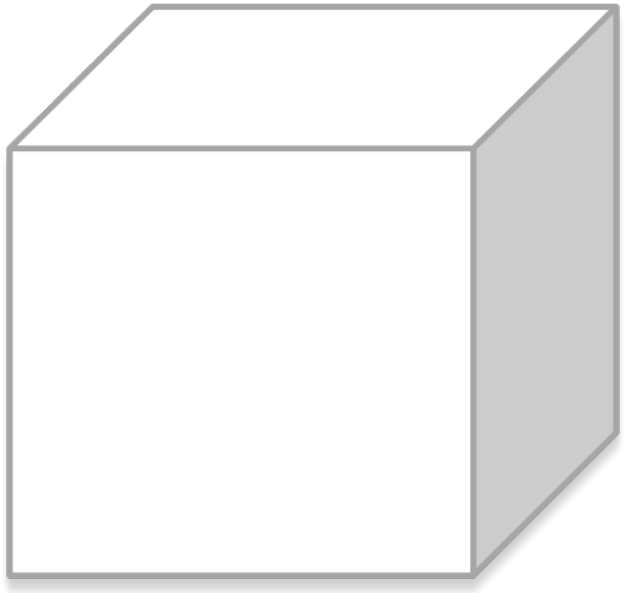
Motivation

- What can our techniques from symmetric-key domain say about whitebox primitives?

- Is it possible to attain any arguable level of residual security in the whitebox setting?

In this talk

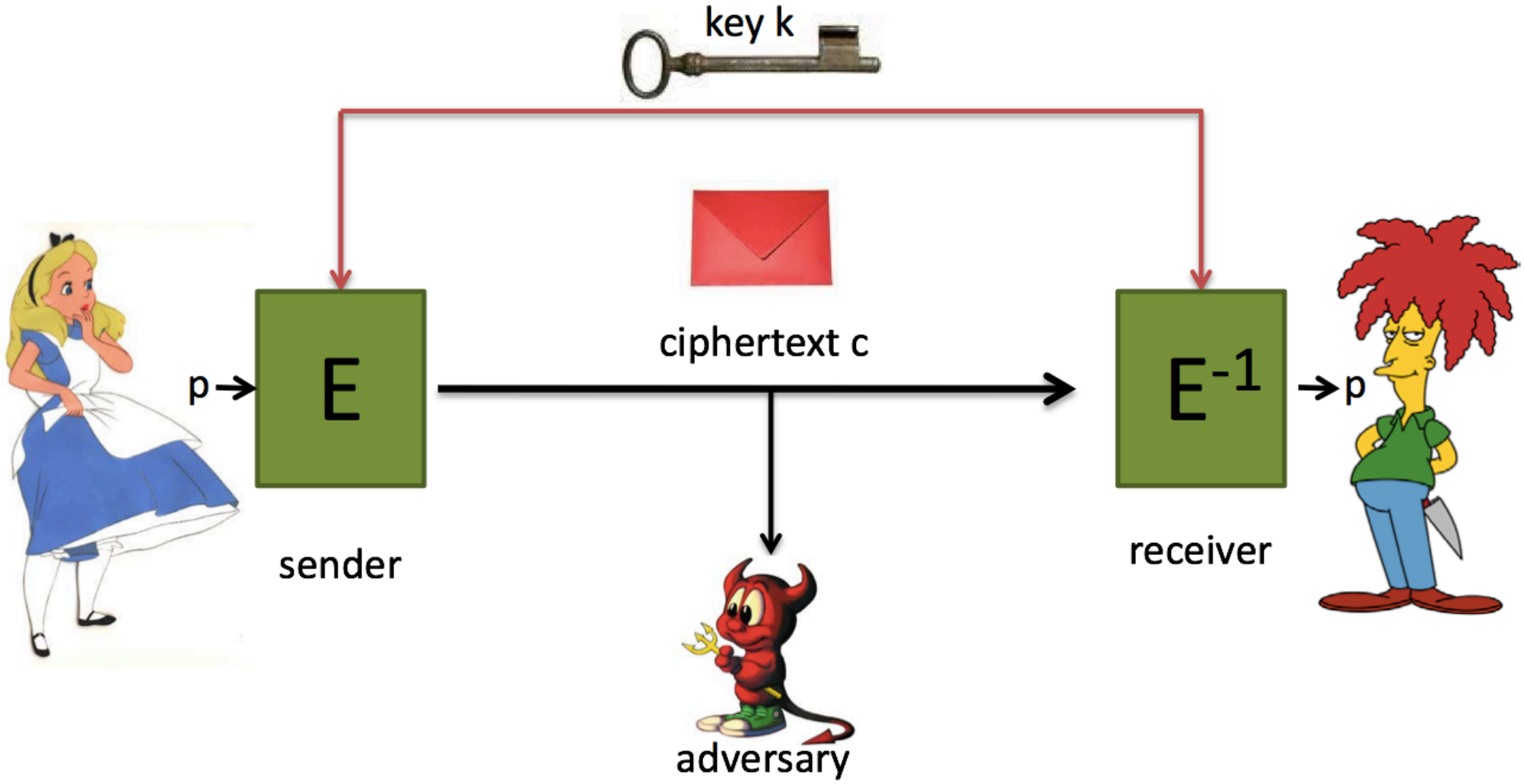
- Setting and Requirements
- Applications
- Existing Whitebox Solutions
- *SPACEcipher*: AES-based Whitebox Block Cipher
- *SPNbox*: Dedicated Whitebox Block Cipher
- Implementations in the Black and White Boxes



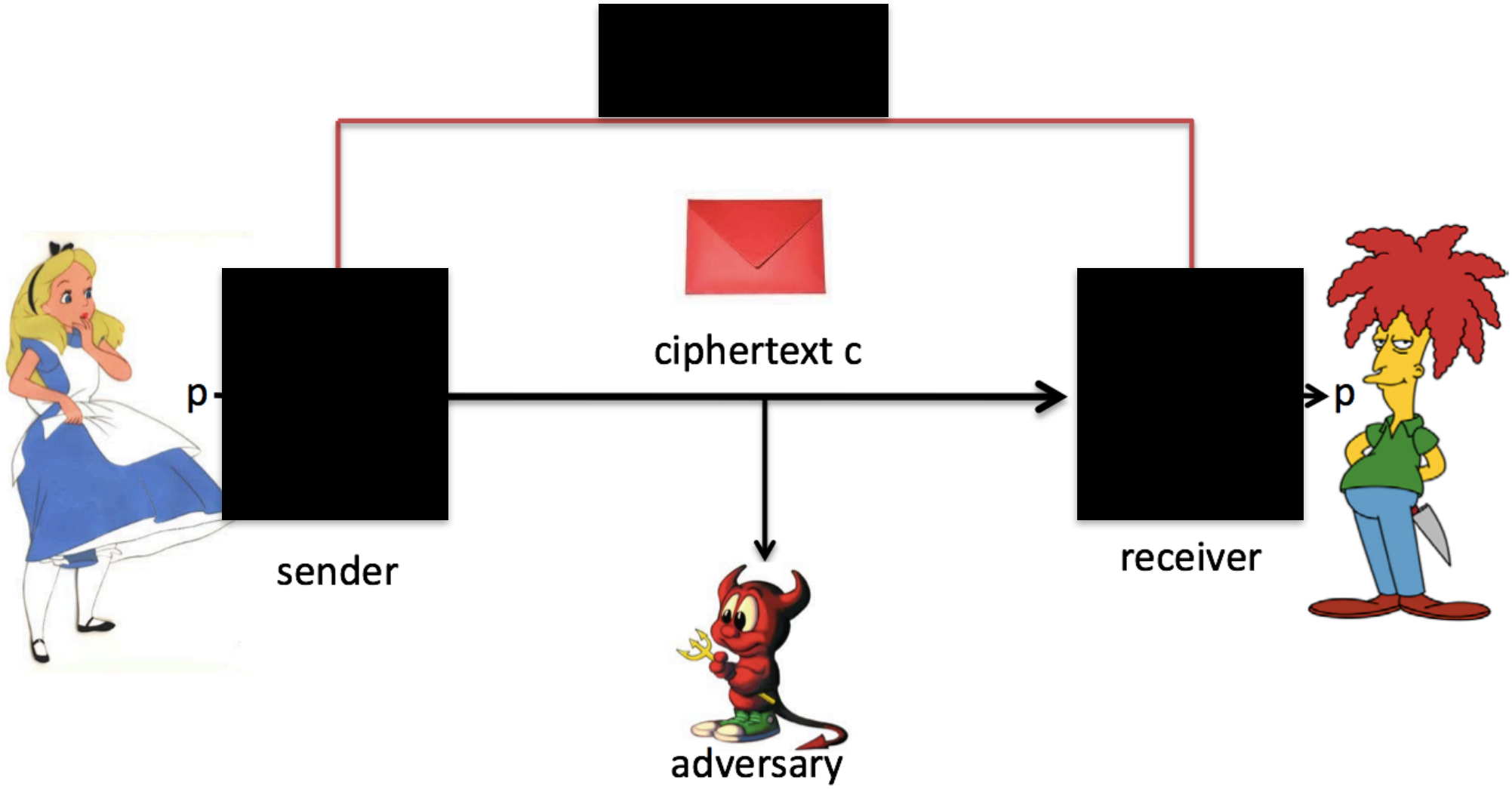
Part 1

IN THE WHITE BOX

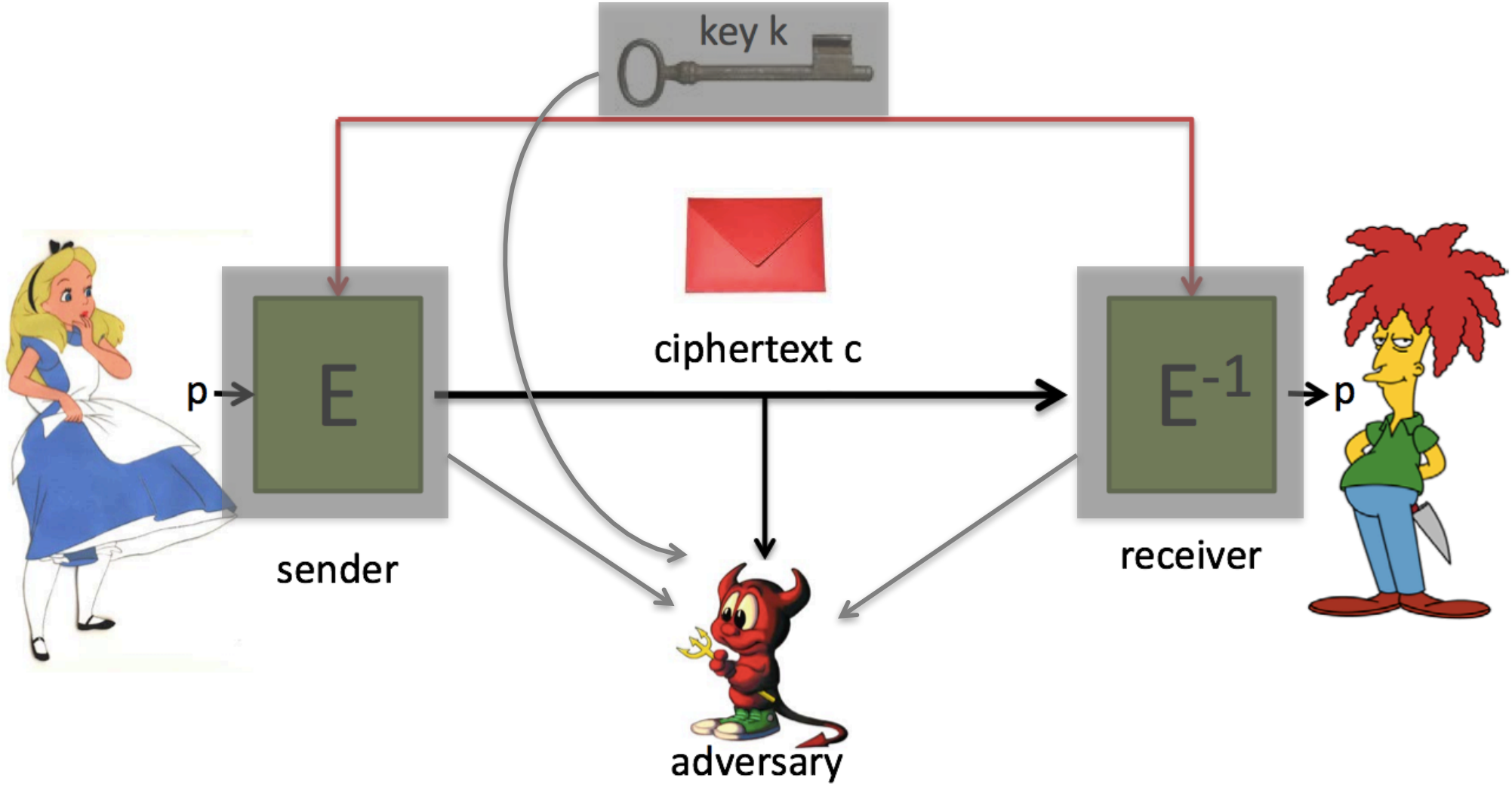
Theory



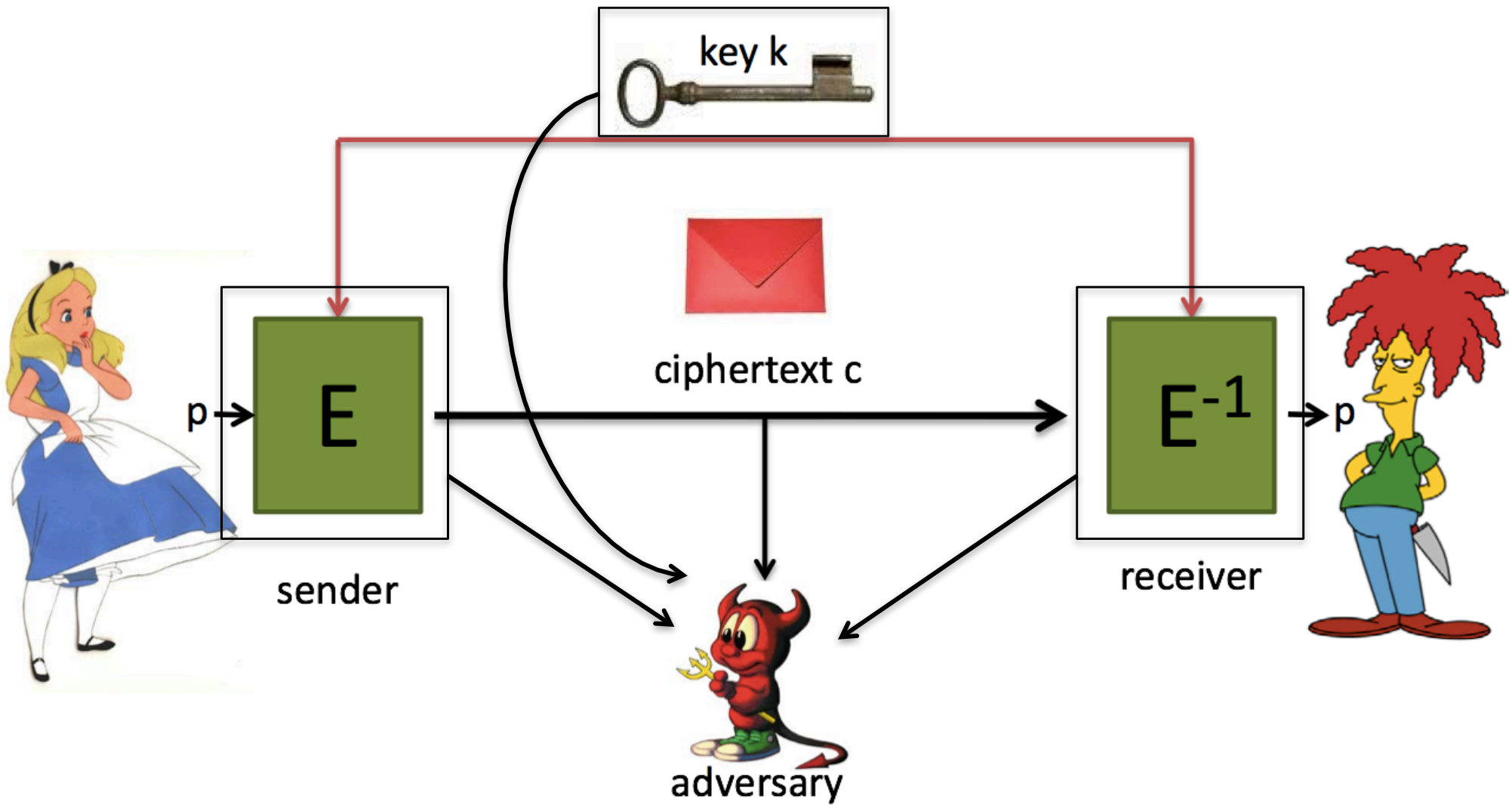
Theory: Black Box



More Realistic: Grey Box



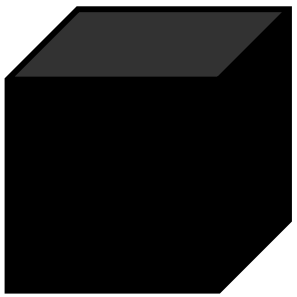
Practice: White Box



Black Box vs White Box

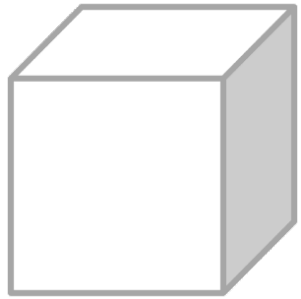
Black box

- Security mechanisms invisible
- Trustworthy hardware and software
- Computer security is based upon confidentiality of secret key



White box

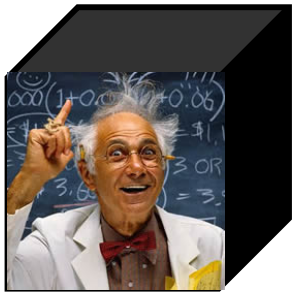
- Malware, Trojans
- Memory leakage, side channels
- Critical weaknesses in OS and applications



Black Box vs White Box

Black box

- Security mechanisms invisible
- Trustworthy hardware and software
- Computer security is based upon confidentiality of secret key



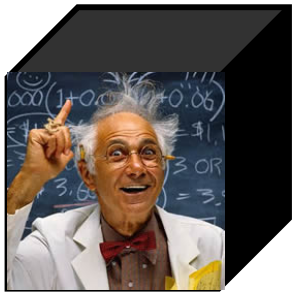
White box

- Malware, Trojans
- Memory leakage, side channels
- Critical weaknesses in OS and applications

Black Box vs White Box

Black box

- Security mechanisms invisible
- Trustworthy hardware and software
- Computer security is based upon confidentiality of secret key

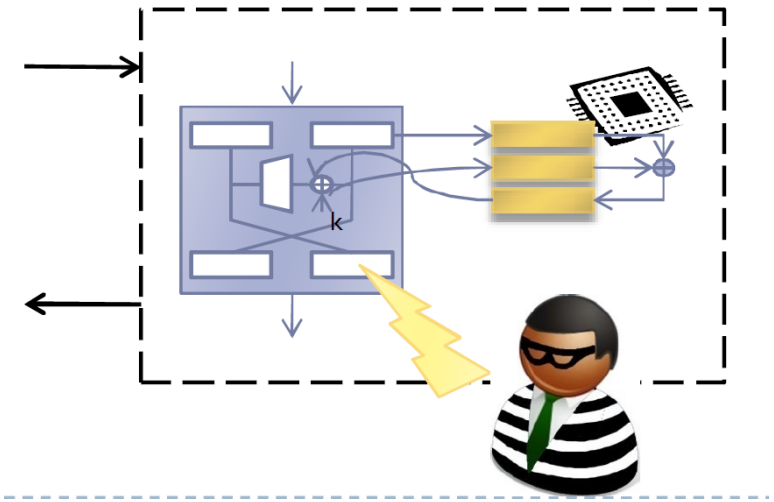


White box

- Malware, Trojans
- Memory leakage, side channels
- Critical weaknesses in OS and applications

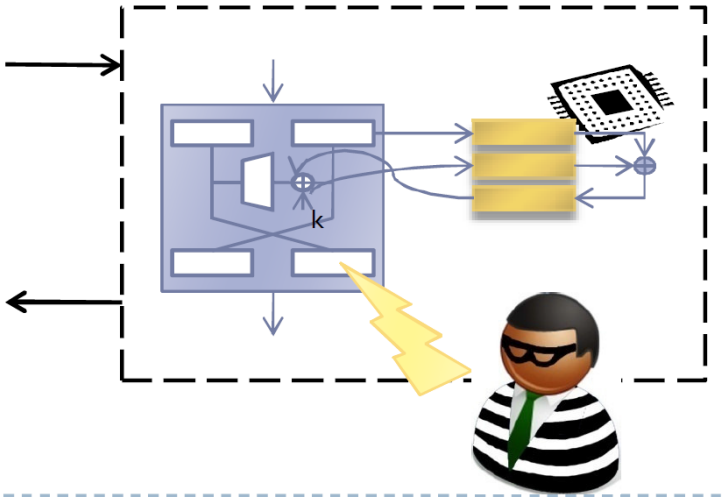
White Box: Attacker in Full Control

- What the whitebox attacker can do
  - Read memory/registers
  - Memory inspection
  - CPU call interception
  - Debugging
  - Reverse-engineering
  - Code tampering
  - Cache attacks
  - Inserting break-points
  - Force a system crash
  - Modification of internal variables
  - Dynamic analysis of the implementation
  - …



White Box: Attacker in Full Control

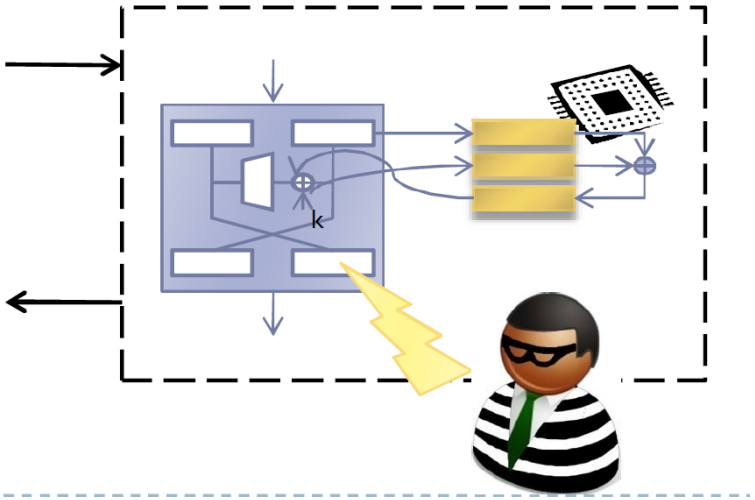
- Adversarial capacity
  - access to intermediate states
  - access to memories
  - access to execution
- Designer's goal
  - attain some residual security
- *Important note*
  - *We cannot protect against every adversary!*



White Box: Residual Security

- **Weak whitebox security**
  - It is difficult to recover the cipher's key
- **Strong whitebox security**
  - Weak whitebox security

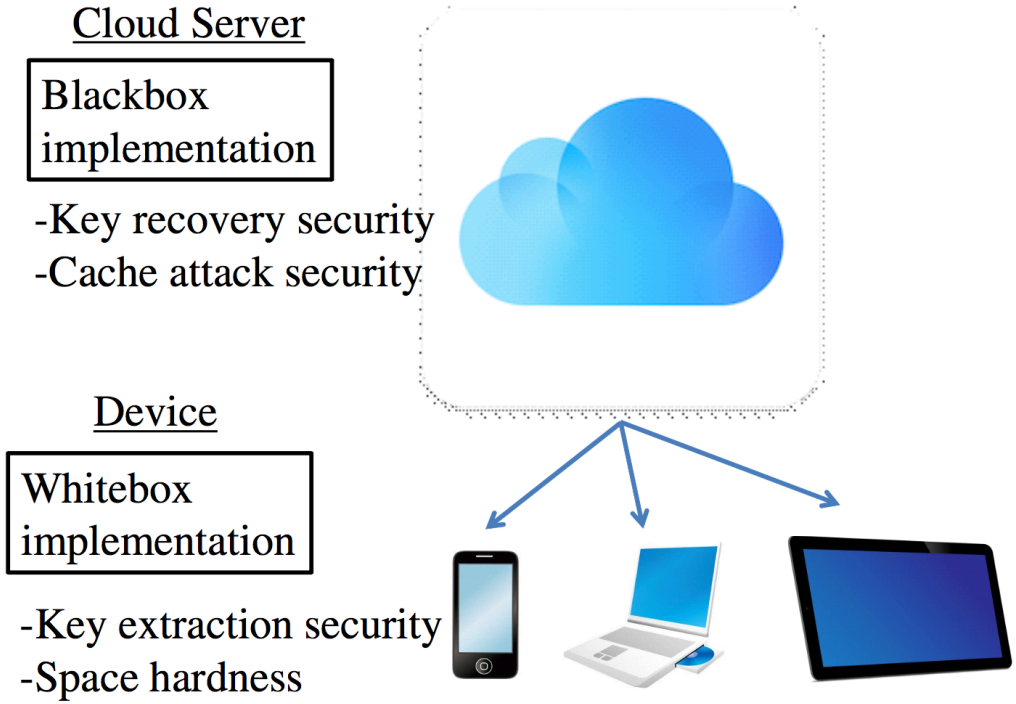
    +
  - It is difficult to encrypt given decryption functionality in WB
  - It is difficult to decrypt given encryption functionality in WB



Part 2

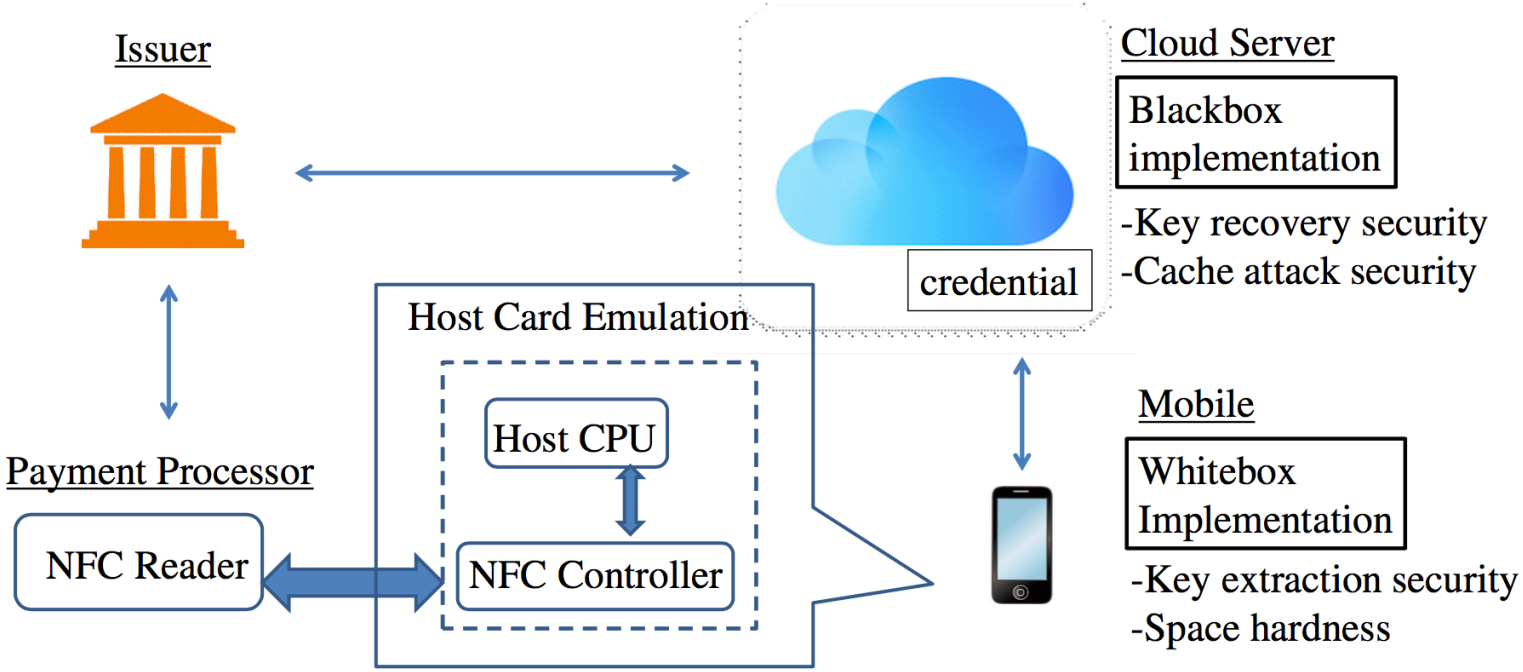
APPLICATIONS

Content Distribution



- **DRM in the cloud**
  - Cloud server encrypts for devices
  - Constant-time blackbox implementation in the cloud
  - Whitebox implementation on the device

Host Card Emulation in Cloud-based Mobile Payments

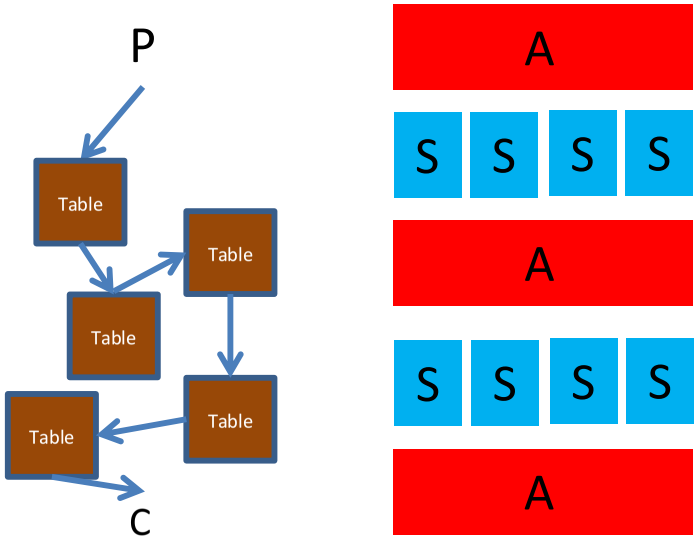


– HCE enables NFC transactions in pure software

– HCE supported from Android 4.4 KitKat on

Other Applications

- Authentication
- Mobile banking
- Governments and military
- Protection against mass-surveillance

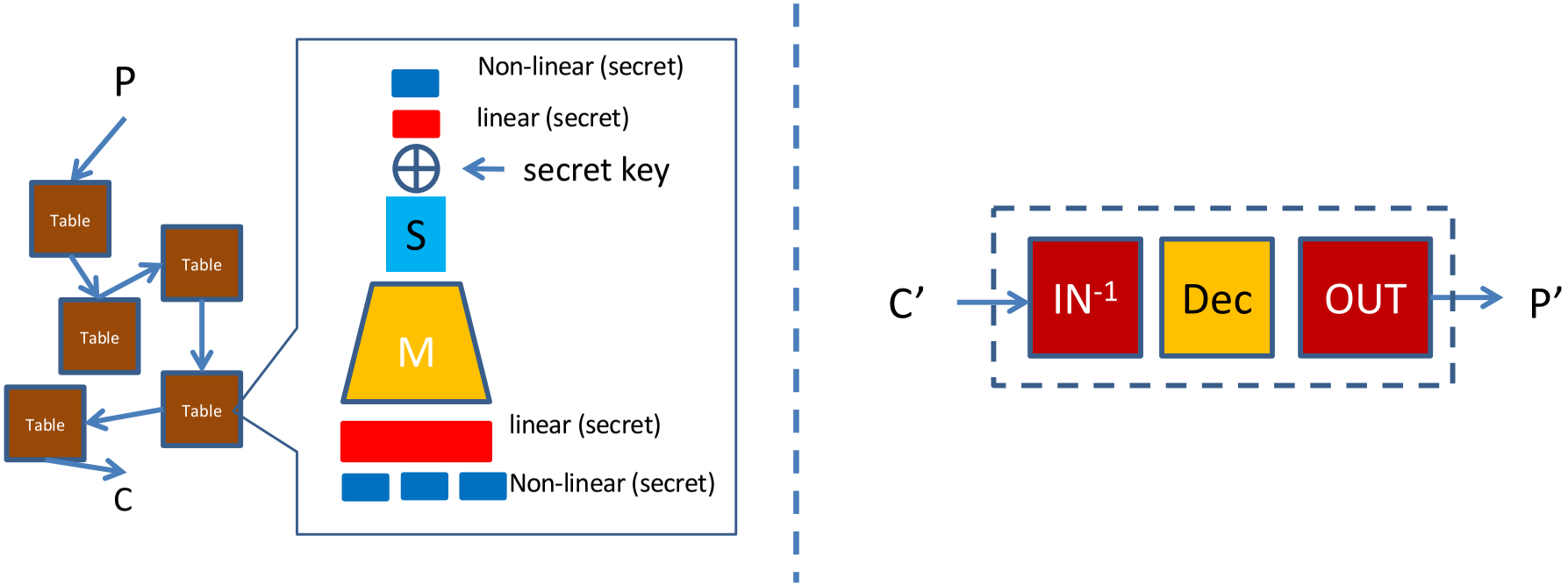


Part 3

EXISTING WHITEBOX SOLUTIONS

Traditional Approach: Tables

- Whitebox Implementation [C+02]
  - Encoded table
    - Convert computations of a cipher (e.g., AES and DES) into table-based ones and put key into table to protect it from WB attacker
  - External encoding
    - Add a secret permutation in the beginning and end of the cipher



Traditional Approach: Tables

- Whitebox AES implementations
  - 8-bit table based [C+02]
  - polynomial equations based [BCD06]
  - 16-bit table based [XL09]
  - dual AES table based [K10]

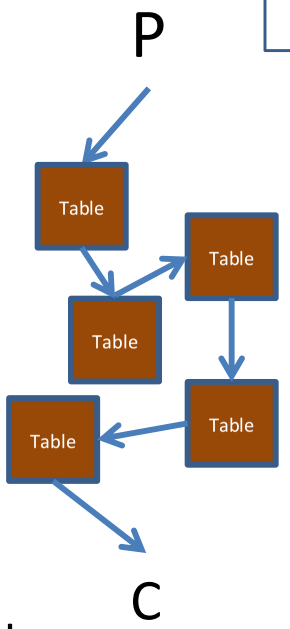
- Whitebox DES implementation
  - 8-bit table based [C+02]

Traditional Approach: Tables

All published WB implementations of AES/DES are broken

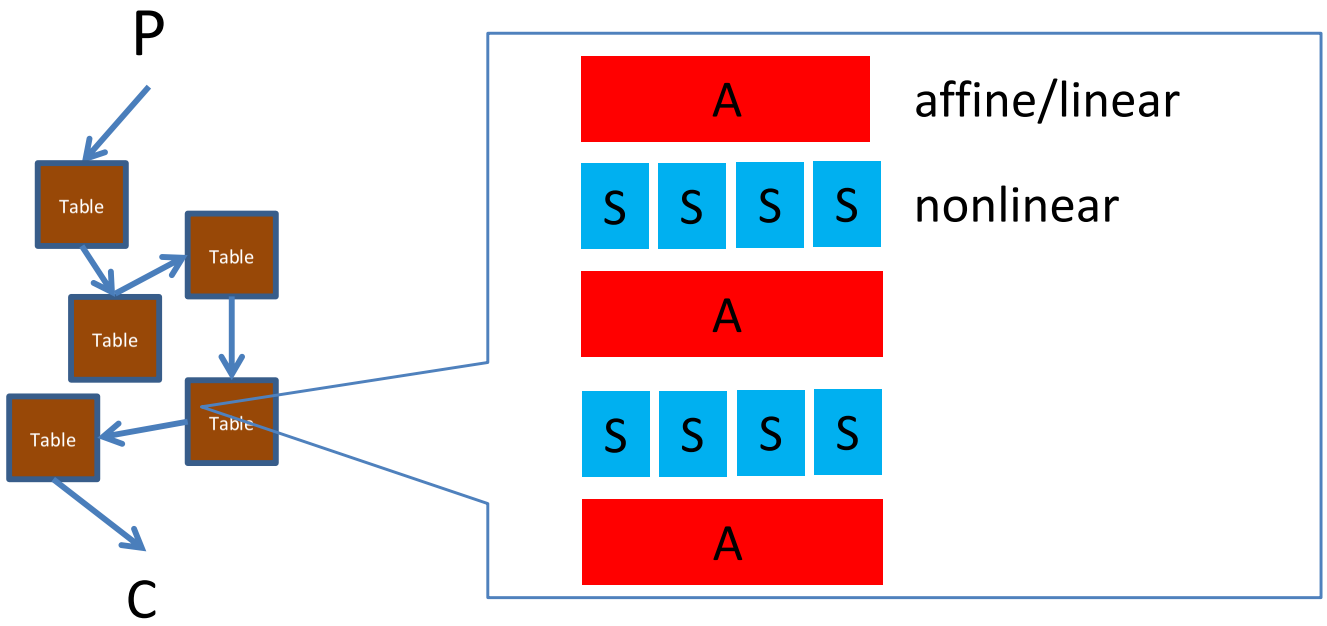
- Whitebox implementations of AES
  - 8-bit table based [C+02]
    - Practical attacks [BGE04][MGH08]
  - Polynomial equations based [BCD06]
    - Practical attacks [M14]
  - 16 bit table based [XL09]
    - Practical attacks [MRP12] [MGH08]
  - Dual AES table-based [K10]
    - Practical attacks [M14]
- Whitebox implementation of DES
  - 8 bit table based [C+02]
    - Practical attacks [W09]
- **Adhoc solutions, limited fundamental base**
- Most implementations are insecure even in gray box
  - DPA by Ruhr University Bochum, FSE'16
  - DCA by NXP, CHES'16
  - DFA by Riscure from BlackHat EU'15

- key extraction
- table decomposition



Dedicated Approach: ASASA

- Dedicated construction: ASASA construction [BBK14]
  - Table-based decomposition-hard problem
    - A: affine/linear bijective transform
    - S: nonlinear bijective transform



Dedicated Approach: ASASA

- Security
  - Hard to quantitatively evaluate
    - Generic attack: n-bit block (ASASA) and m-bit S-box
      - Time to compose : $2^{(n-m)m}$
        - » If m = 8, n = 16 : security 64 bits
  - Practically broken
    - key recovery [IDKL15, MDFK15]
    - code lifting (decomposition of table) [IDKL15, MDFK15]
  - At least 12 layers are needed to attain security [BK15]
  - The underlying problem needs more analysis

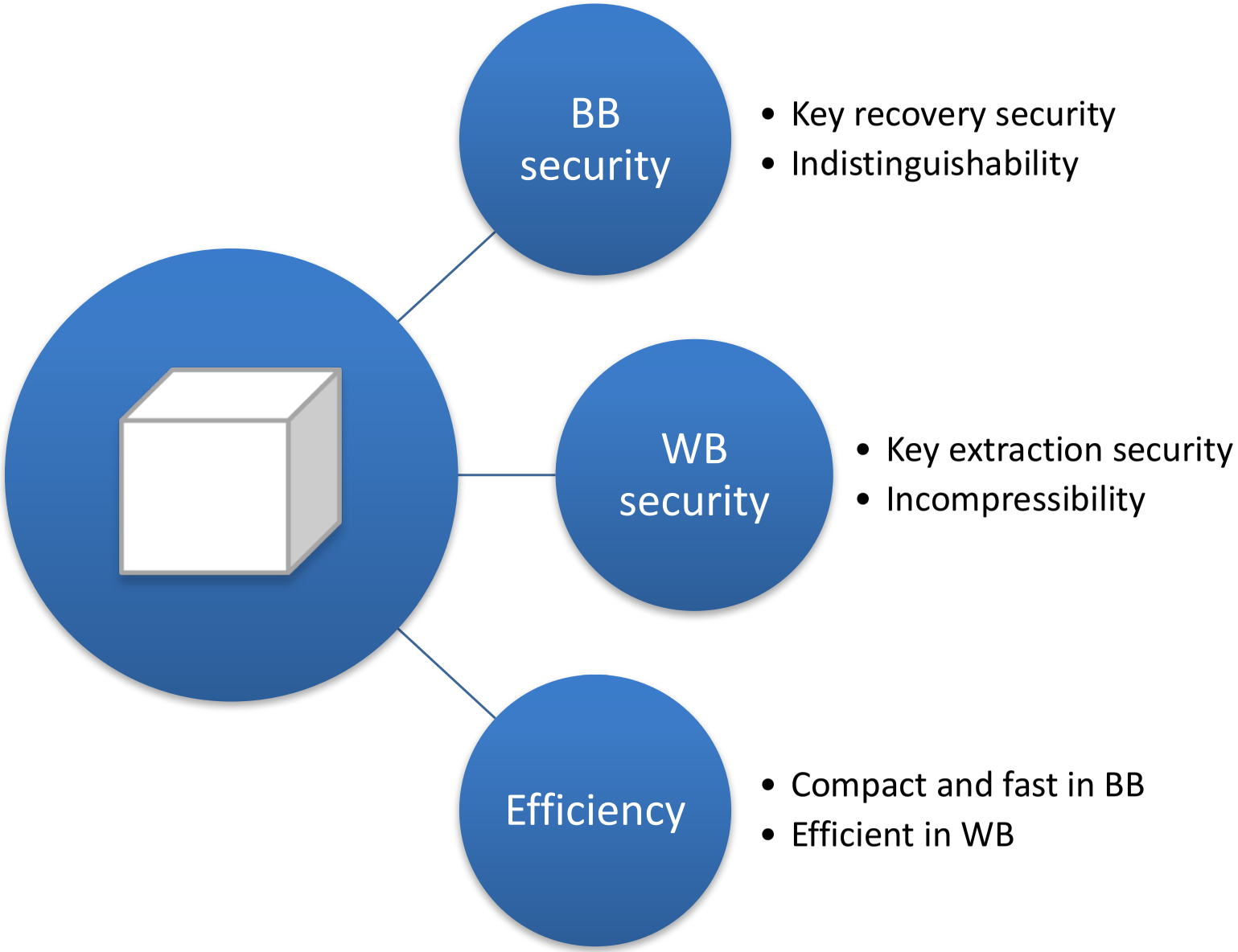
Existing Approaches

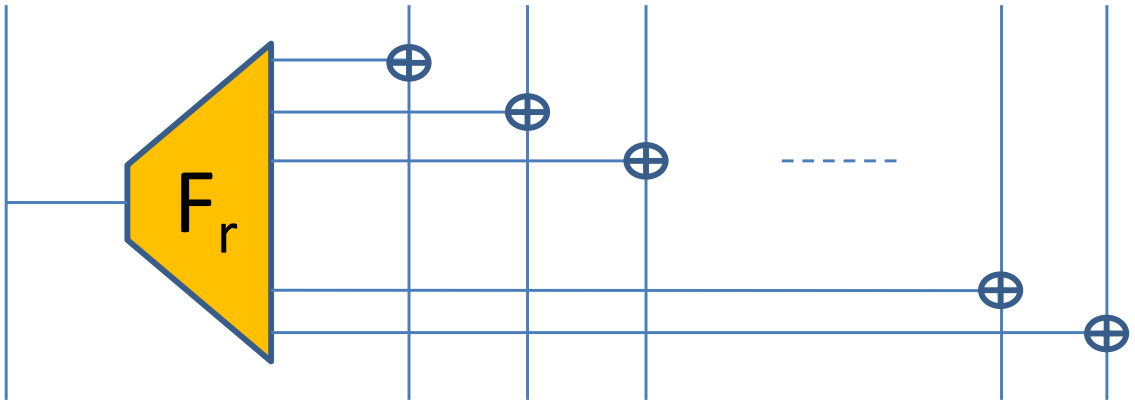
Summary of Practical Symmetric-Key Whitebox Proposals

	Blackbox		**Whitebox**	
	Key Recovery	Distinguishing	Key Recovery	Decomposition
WB-AES [C+02] and similar	Secure	Secure	Insecure [BGE04]	Insecure [BGE04]
ASASA [BBK14]	Secure?	Secure?	Insecure [IDKL15, MDFK15]	Insecure [IDKL15, MDFK15]

Any comparable approach with some security in the whitebox?

Challenge:
Robust Whitebox Cryptography



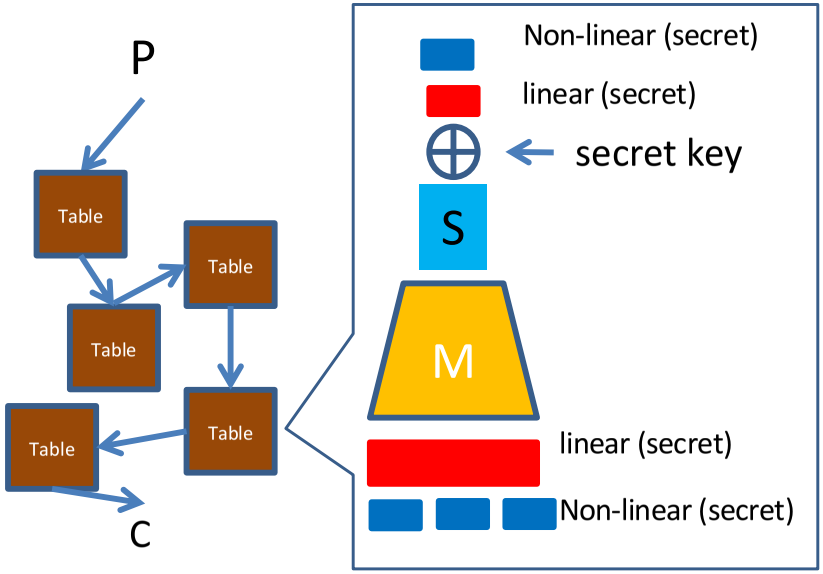


Part 4

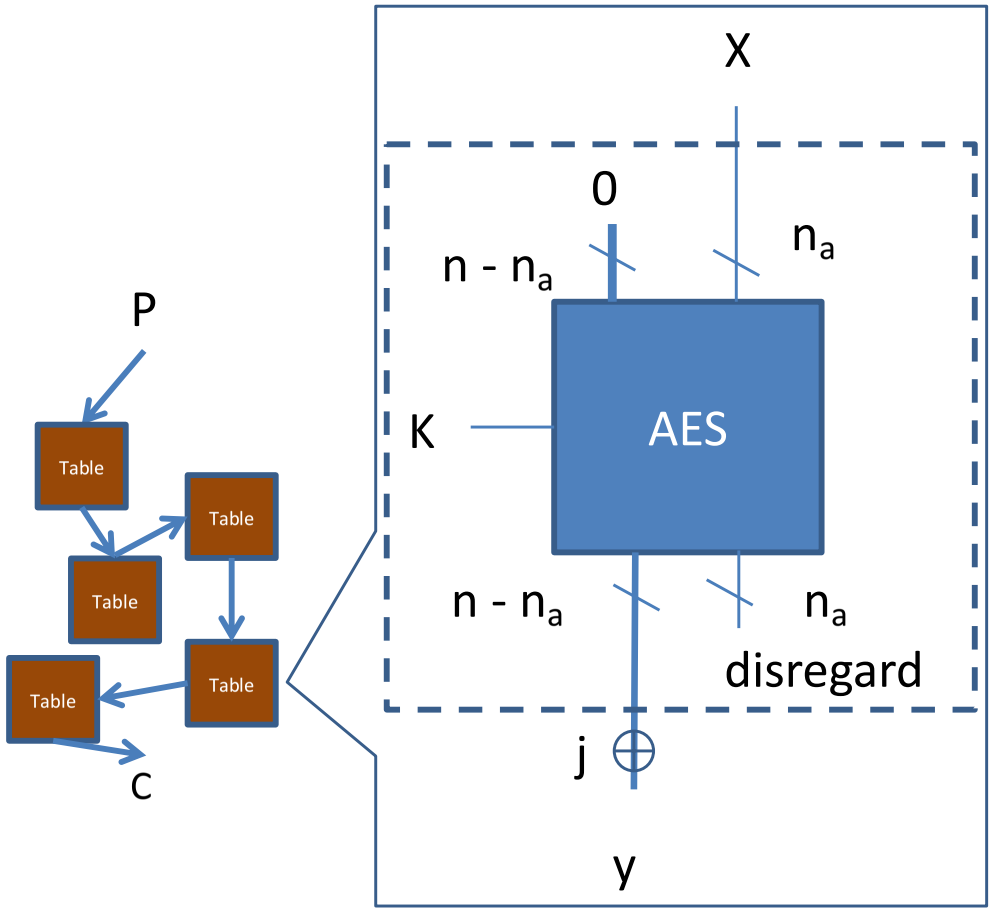
SPACE CIPHER (ACM CCS'15): AES-BASED WHITEBOX BLOCK CIPHER

What is Different?

Traditional WB solutions [C+02] and others



SPACEcipher



Design Goals

1. Security of the whitebox solution relies on a well-analyzed problem
   - key recovery problem for a block cipher, e.g. AES
2. No external encoding
   - executable in the stand-alone manner to be applicable in a wide range of environments
3. Multiple code (table) sizes if needed
   - Apply differently sized tables in different rounds

Security Requirements

- Security in the black box
  - **Key recovery resistance**
    - computationally hard to extract a key
  - **Distinguishing resistance**
    - computationally hard to distinguish it from random keyed perm.
- Security in the white box
  - **Key recovery resistance**
    - computationally hard to extract a key
  - **Space hardness (decomposition resistance)**
    - computationally hard to decompose internal component (table)
      - ($T/2$, 128)-space hardness
      - cf. (in)compressibility in SAC'13
      - cf. big-key symmetric encryption in CRYPTO'16 and key derivation in AC'16

What is Space Hardness?

DEFINITION 1 ((M, Z)-SPACE HARDNESS). *The function F is a (M, Z)-space hard implementation of a block cipher E_K if it is infeasible to efficiently compute encryption/decryption with probability of more than 2^{-Z} for any plaintext/ciphertext given the code (table) whose size is less than M in whitebox environments.*

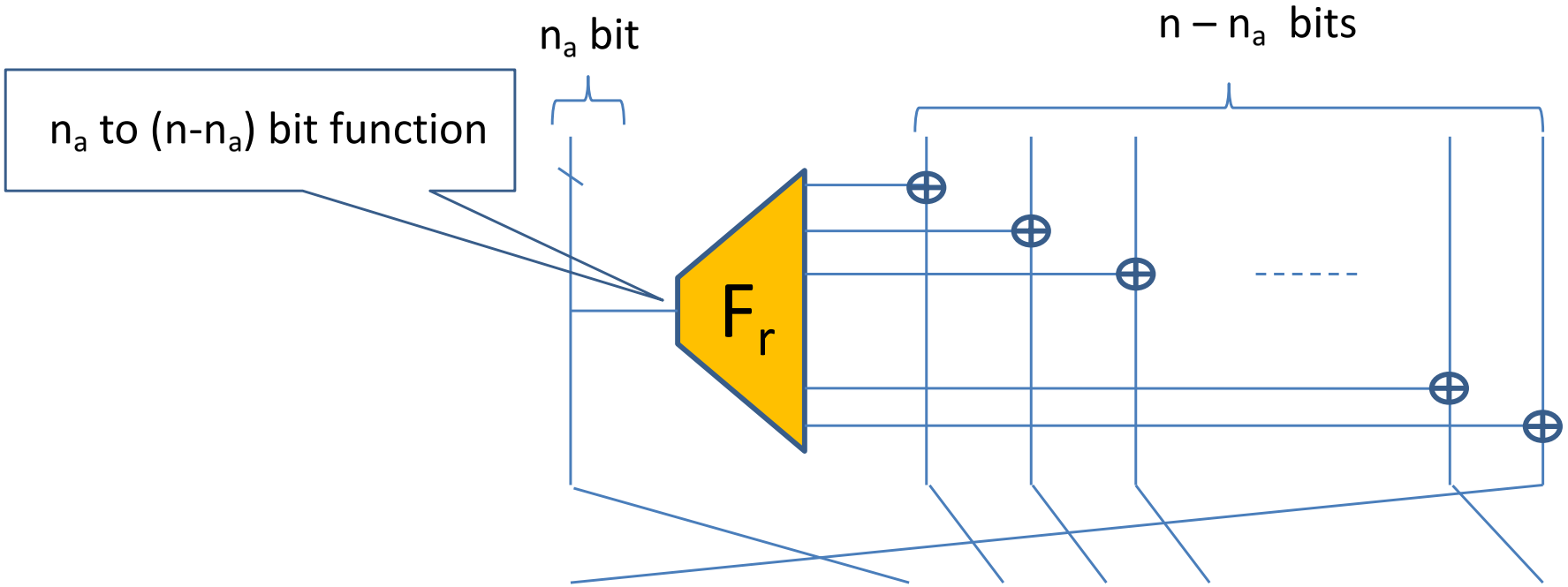
E.g., (**T/2**, 128)-space hardness :

An attacker needs to obtain at least half of the total table size to compute any plaintext or ciphertext with probability of 2^{-128}

It enables us to quantitatively evaluate security of code lifting attacks by the amount of required code (table) size to be isolated from white-box environments for an attacker.

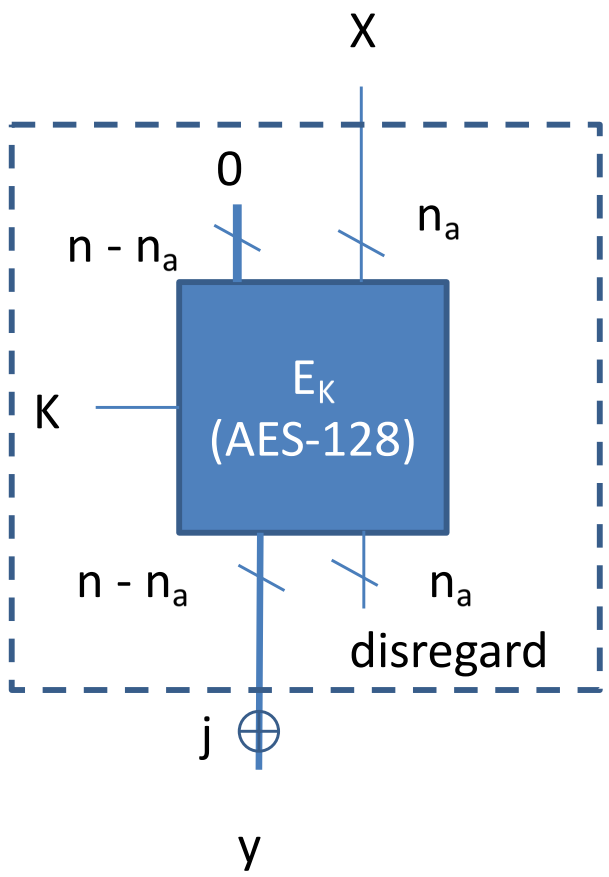
Unbalanced Target-Heavy Feistel Network

- Block size : n
- #branches: l
- Size of each line : n/l bit
- Function (Table) size: n_a to $(n - n_a)$ bits



The F-function

- n_a to (n- n_a)-bit function
  - based on well-analyzed block cipher E_k
    - e.g., AES, PRESENT, etc
  - $y = F_r(X) = trunc_{n\text{-}na}(E_k(i \,||\, X)) \,\hat{}\, j$
    - i = 0, j = r (excluded from table)
      - Same F-function w/ round constants



$trunc_x(Y)$: output x bit of Y , x < n

Example: SPACEcipher-X

- 4 variants with differently sized F-functions

  - SPACECIPHER-8 : $n = 128,\ \ell = 16,\ R = 300,\ n_a = 8,\ F_8^r : \{0,1\}^8 \rightarrow \{0,1\}^{120}$
  - SPACECIPHER-16 : $n = 128,\ \ell = 8,\ R = 128,\ n_a = 16,\ F_{16}^r : \{0,1\}^{16} \rightarrow \{0,1\}^{112}$
  - SPACECIPHER-24^{*3} : $n = 128,\ \ell = 16,\ R = 128,\ n_a = 24,\ F_{24}^r : \{0,1\}^{24} \rightarrow \{0,1\}^{104}$
  - SPACECIPHER-32 : $n = 128,\ \ell = 4,\ R = 128,\ n_a = 32,\ F_{32}^r : \{0,1\}^{32} \rightarrow \{0,1\}^{96}$

Security in the White Box

- Key extraction in WB
  - Relies on the block cipher security in BB
    - What an WB attacker can do is to know/choose input and output of table
    - A subset of attacks on AES possible only

Security in the White Box

- Space hardness (decomposition)
  - (**T/2**, 128)-space hardness
    - An attacker needs to obtain at least half of the total table size to compute any plaintext or ciphertext with probability of more than 2^{-128}

Trade-off between M and T

Cipher	M	T
$K = 128$		
SPACECIPHER-8	2.85 KB	3.84 KB
SPACECIPHER-16	459 KB	918 KB
SPACECIPHER-24	109 MB	218 MB
SPACECIPHER-32	25.8 GB	51.5 GB

T : total table size
M: code isolated

Security in the Black Box

- Evaluation against distinguishing attacks

	R	G	F	D	L	ID	I
SPACEcipher-8	300	47	17	152	17	34	19
SPACEcipher-16	128	23	9	44	9	18	12
SPACEcipher-24	128	-	15	32	6	30	17
SPACEcipher-32	128	11	5	14	5	10	10

G : Generic attack F : Full Diffusion
D : Differential attack, L : Linear attack
ID : Impossible differential attack, I : Integral attack

Performance in white box

	Performance	Table size
SPACEcipher-8	300 TL	3.84 KB
SPACEcipher-16	128 TL	918 KB
SPACEcipher-24	128 TL	218 MB
SPACEcipher-32	128 TL	51.5 GB
ASASA-1 [3]	64 TL	8 MB
ASASA-2 [3]	64 TL	384 MB
ASASA-3 [3]	25 TL	20 GB
AES(Chow et al) [11]	3008 TL	752 KB
AES(Xiao-Lai) [37]	80 TL	20.5 MB
AES(Black-box) [13]	160 TL	4 KB

Target

L1 cache

L3 cache

RAM

HDD

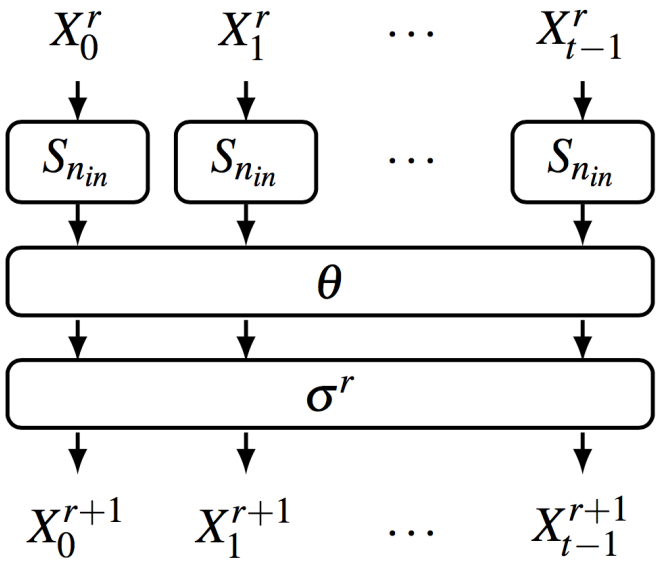
ASASA-1 : S layer consists of 8×16-bit

ASASA-2 : S layer consists of 24-bit $+ 6 \times 16$-bit $+ 8$ bit

ASASA-3 : S layer consists of 4×28-bit $+ 16$-bit

Performance in black box

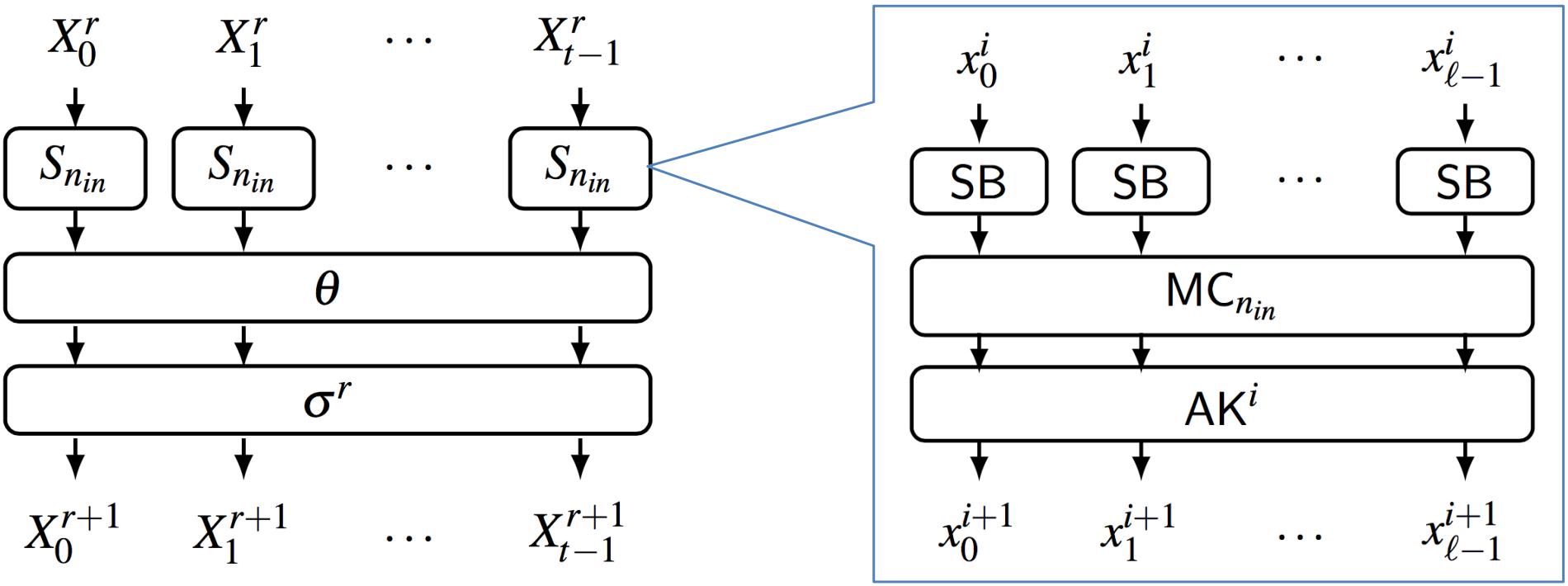
- Implementation without tables is possible by decomposing the tables
- Underlying internal block cipher can be freely chosen depending on user requirements
  - a wide range of implementations in the black box are thinkable
  - For example:
    - S/W lightweight block cipher such as PRIDE and SIMON/ SPECK
      - Implementation with very small size of RAM and code is possible
  - AES-128
    - Optimization for speed by AES-NI and bit sliced implementations



Part 5

SPN BOX: DEDICATED WHITEBOX CIPHER

Design: Nested SPN



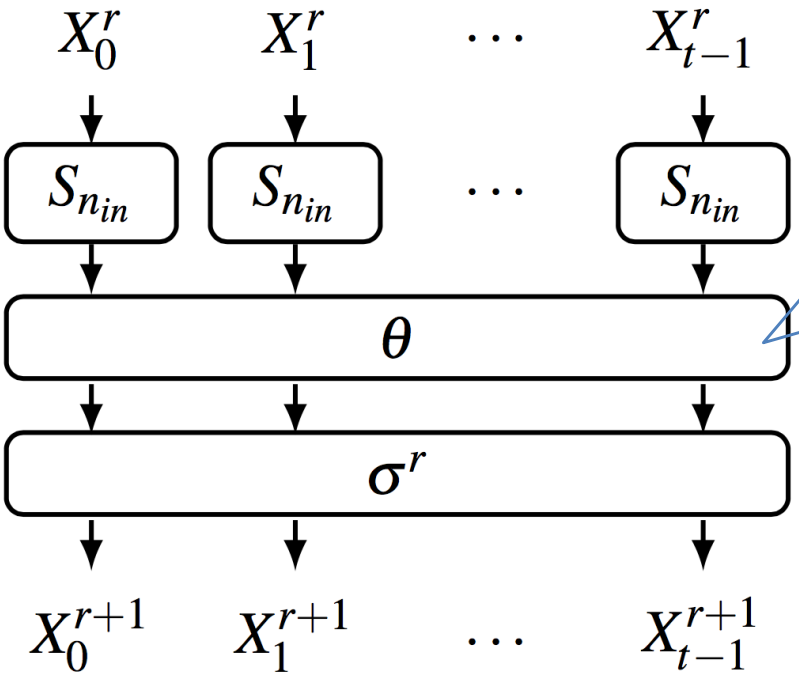
Outer block cipher:

- 120- or 128-bit block
- 10 rounds
- MDS matrix

Underlying block cipher:

- 8-, 16-, 24- or 32-bit block
- 16, 20, 32 or 64 rounds
- AES S-box
- AES MixColumn based MDS diffusion

Design: Diffusion in the Outer Cipher



$$M_{32} = \mathrm{cir}\,(1_x, 2_x, 4_x, 6_x)$$
$$M_{24} = \mathrm{cir}\,(1_x, 2_x, 5_x, 3_x, 4_x)$$
$$M_{16} = \mathrm{had}\,(1_x, 3_x, 4_x, 5_x, 6_x, 8_x, \mathtt{b}_x, 7_x)$$
$$M_8 = \mathrm{had}\,(08_x, 16_x, 8\mathtt{a}_x, 01_x, 70_x, 8\mathtt{d}_x, 24_x, 76_x, \mathtt{a}8_x, 91_x, \mathtt{ad}_x, 48_x, 05_x, \mathtt{b}5_x, \mathtt{af}_x, \mathtt{f}8_x)$$

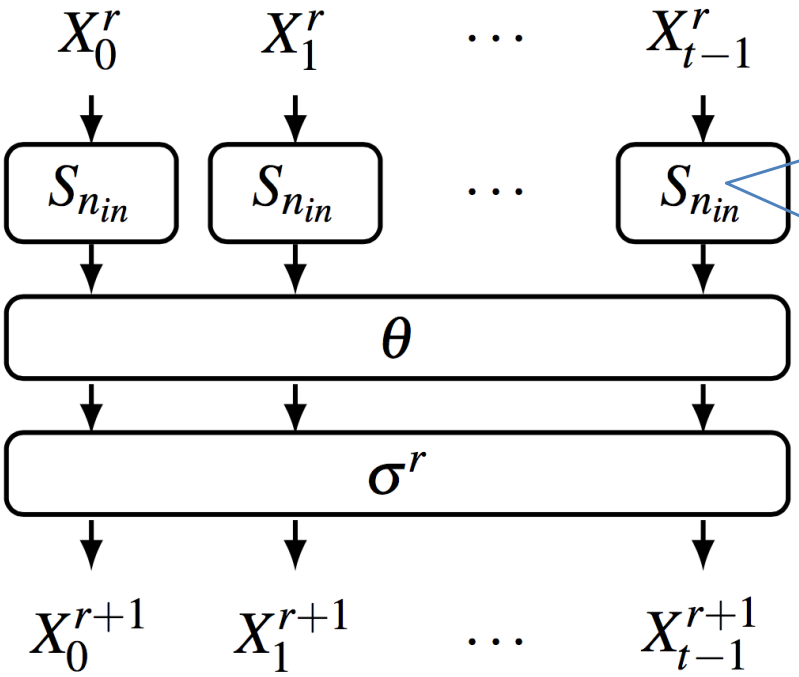
Outer block cipher:

- 120- or 128-bit block
- 10 rounds
- MDS matrix

Matrix:

- M_{32}, M_{16} and M_8 are involutions
- M_{32} and M_{16} used in Anubis and Khazad
- M_8 is an optimized involutory Hadamard-Cauchy matrix from FSE'15

Design: Diffusion in the Inner Cipher



$$A_{32} = \mathrm{cir}\,(2_x, 1_x, 1_x, 3_x)$$

$$A_{24} = \begin{pmatrix} 2_x & 1_x & 1_x \\ 3_x & 2_x & 1_x \\ 1_x & 3_x & 2_x \end{pmatrix}$$

$$A_{16} = \begin{pmatrix} 2_x & 1_x \\ 3_x & 2_x \end{pmatrix}$$

Outer block cipher:

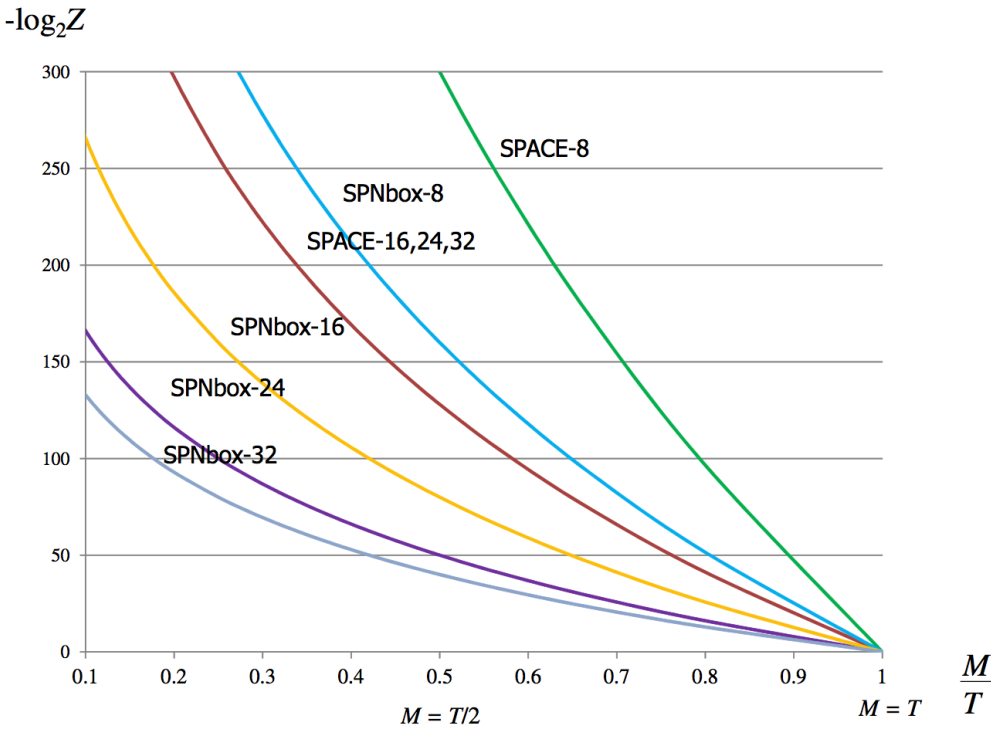
- 120- or 128-bit block
- 10 rounds
- MDS matrix

Matrix:

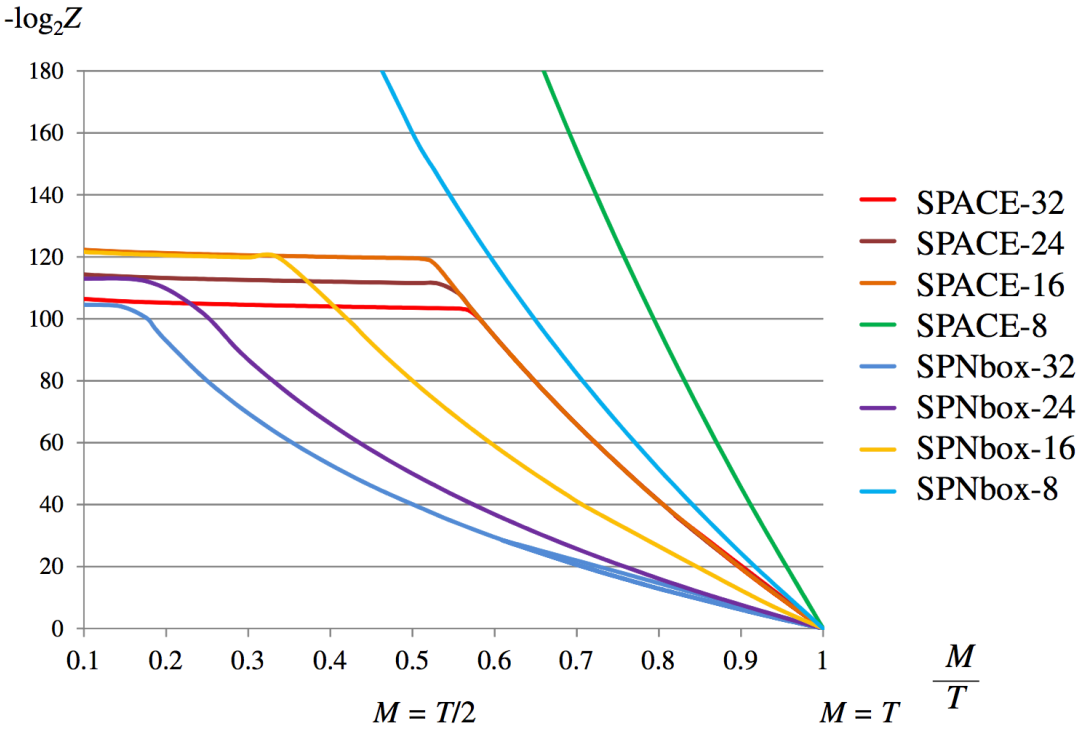
- A_8 is the identity
- All matrices are submatrices of the AES MixColumn transform

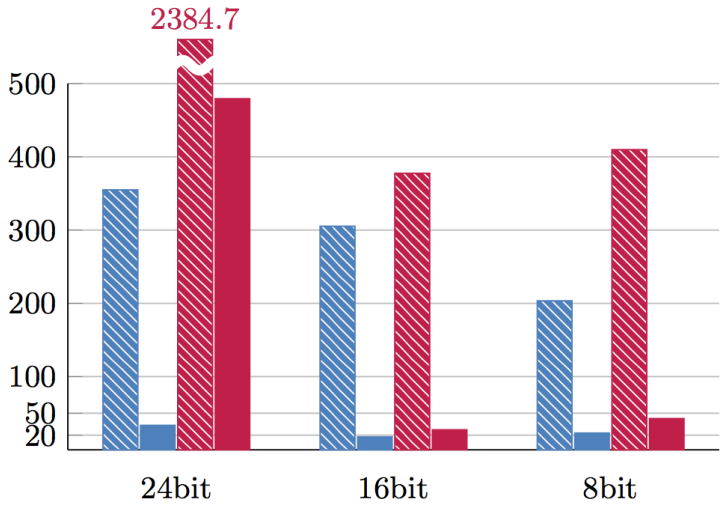
Weak Space Hardness

Before the plaintext is given, the attacker can copy tables non-adaptively



Before the plaintext is given, the attacker can copy tables adaptively

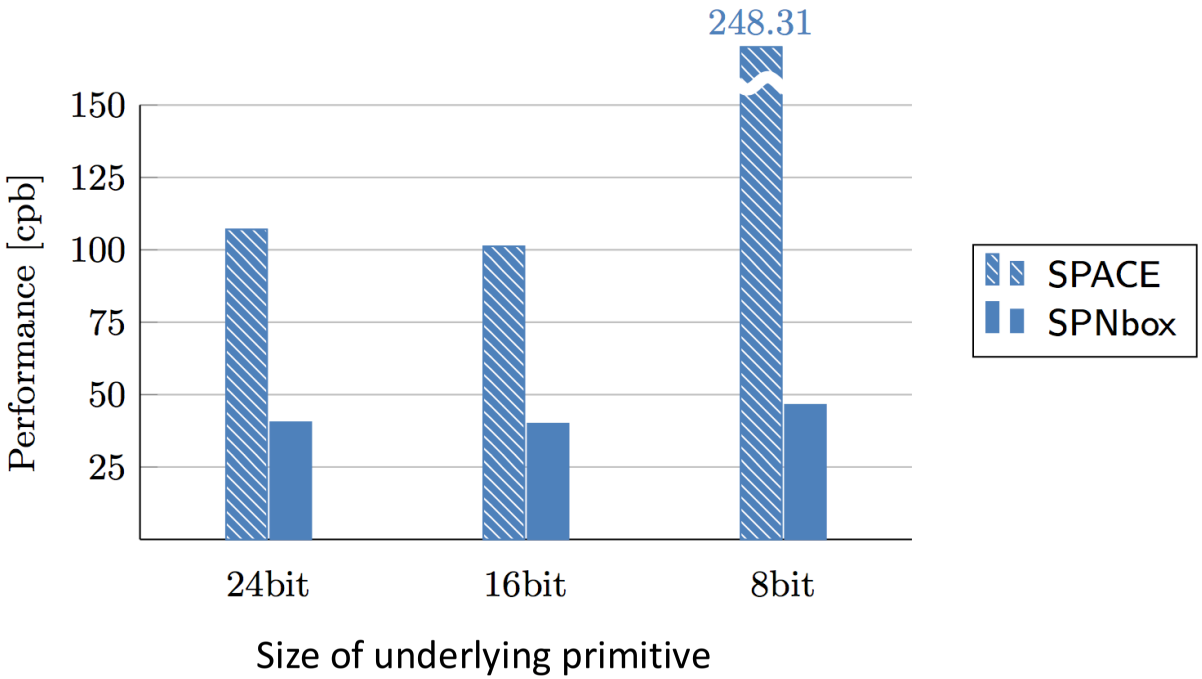




Part 6

IMPLEMENTATION STUDY

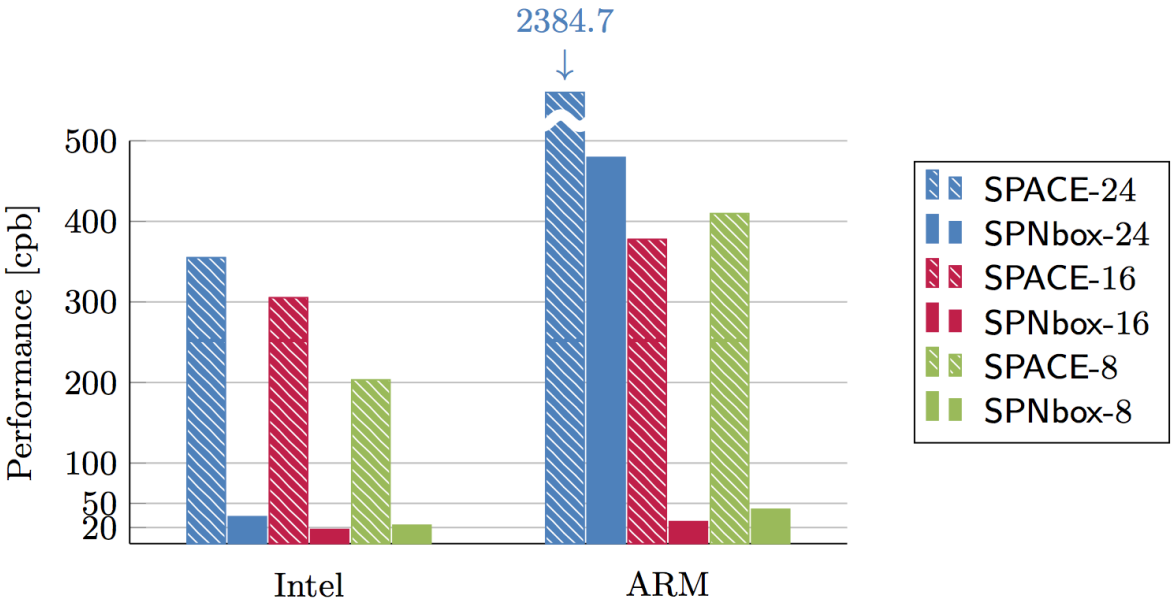
Blackbox Implementation



Algorithm	Rounds (outer)	Rounds (inner)	Performance [cpb]
SPNbox-32	10	16	15.09
SPNbox-24	10	20	40.48
SPNbox-16	10	32	39.98
SPNbox-8	10	64	46.49
SPACE-32	128	10	101.02
SPACE-24	128	10	107.01
SPACE-16	128	10	101.21
SPACE-8	300	10	248.31

Constant-time BB performance on Intel Skylake with AES-NI, Intel Core i7-6700, at 3400 MHz with disabled TurboBoost and disabled hyperthreading, averaged over 100000 repetitions (lower is better)

Whitebox Implementation



Algorithm	Rounds (outer)	Table size	Performance [cpb] Intel	ARM
SPNbox-32	10	17.2 GB	184.56	—
SPNbox-24	10	50.3 MB	33.48	479.38
SPNbox-16	10	132 KB	17.59	27.37
SPNbox-8	10	256 B	22.93	42.66
SPACE-32	128	51.5 GB	5535.01	—
SPACE-24	128	218 MB	354.86	2384.74
SPACE-16	128	918 KB	305.11	377.51
SPACE-8	300	3.84 KB	203.19	409.57

WB performance on Intel Skylake i7-6700 and ARMv8 Cortex-A57 (Samsung Galaxy S6)

Conclusions I

- Secure AES-based WB cipher: SPACEcipher
  - Security = key recovery, so weak WB security
  - Same algorithm, different possible space requirements
  - Key extraction in WB bases directly on AES key recovery
- Secure dedicated WB cipher: SPNbox
  - Weak WB security
  - Higher performance than SPACEcipher
  - Key extraction in WB bases on the security of a dedicated cipher

Conclusions II

- Other efficiency/space-hardness tradeoffs possible
  - Up to 2-7x speedup for SPACEcipher
  - Up to 2x speedup for SPNbox
- More detailed and further provable settings possible
  - Cf. big-key symmetric encryption, CRYPTO'16
  - Cf. strong space-hardness, see this paper
  - Cf. key derivation in the next talk

Performance Comparison

P.-A. Fouque, P. Karpman, P. Kirchner, B. Minaud “Efficient and Provable White-Box Primitives”, next talk [FKK+16]

	Whitebox, cycles per call	Blackbox, cycles per call
Puppycipher-16 [FKK+16]	2960	4140
Hound-16 [FKK+16]	2300	3520
Coureurdesbois-16 [FKK+16]	3190	3100
SPNbox-16, here	**281**	**640**
Puppycipher-24 [FKK+16]	27570	6760
Hound-24 [FKK+16]	26540	5490
Coureurdesbois-24 [FKK+16]	17360	4470
SPNbox-24, here	**502**	**607**

[FKK+16]: Xeon E5-1603v3 (Haswell)
Ours: i7-6700 (Skylake)