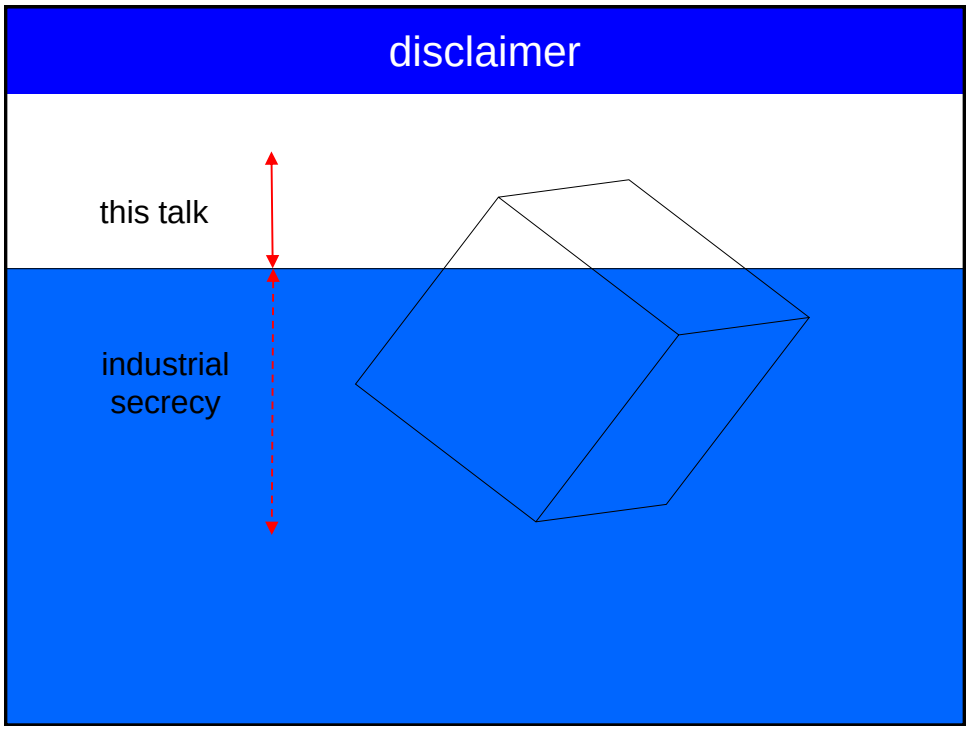
On White-Box Cryptography

Henri Gilbert
ANSSI, France

disclaimer



The Uneasy Relationship between White-Box Cryptography and the Security w/o Obscurity Principle

outline

1. White-Box Cryptography
   - informal definitions, related notions
   - seminal WB-AES implementation [Chow et al. 02]
2. Cryptanalytic toolbox for WB-AES and sisters
   - attacks on WB implementations based on networks of lookup tables
3. Alternative approaches and perspectives
   - can WB cryptography and security w/o obscurity be reconciliated?

White-Box Cryptography in a nutshell (1/2)

- aim of White-Box Crypto (WBC)
  - protect software implem. of crypto from full access attacks by insiders
  - «prevent the extraction of secret keys from the program » [CEJO02]
- cryptography as usual does not address this
  - keys and algorithms are assumed to reside in a trusted module
  - protect keys and information from outsiders' attacks
- examples of applications of WBC
  - protection of «Internet distribution of e-books, music and video » (DRM, e-services) against insiders, including « legitimate users »

  - potentially: protection of software only deployments of crypto against insiders, e.g. malwares, system intruders, etc.

white-box cryptography (WBC) in a nutshell (2/2)

- in their seminal papers [CEJO02-aes/des]

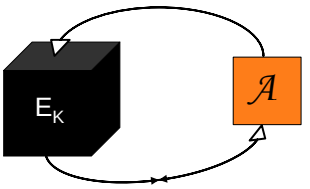
  Chow, Eisen, Johnson, and van Oorschot introduce two notions
  - WB attack context: «full visibility into software implem. and execution»
  - WB implemention: obfuscated description of a keyed instance of the cipher

  and specify WB implementations of AES and DES
  - WB-AES [CEJO02-aes]
  - WB-DES [CEJO02-des]
- this part outlines
  - 2 security models for WB crypto: weak WB and strong WB
  - the WB-AES implementation of [CEJO02-aes]

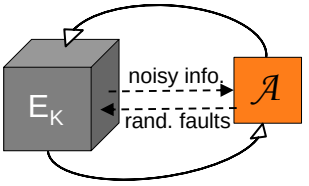
this talk exclusively addresses WB implementations of block ciphers

WB model / adversaries' resources

- Black Box

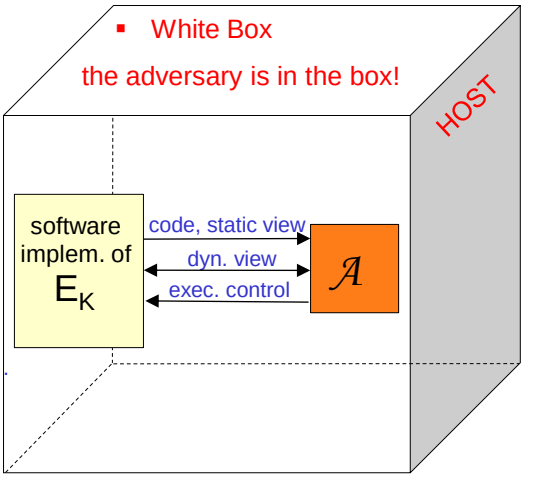


- Grey Box



- White Box

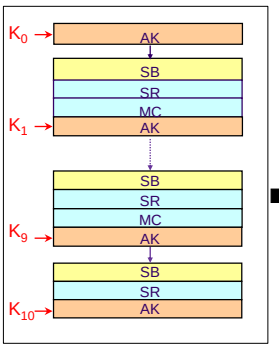
the adversary is in the box!



« choice of implementation is the sole remaining line of defense » [Chow et al. 02]

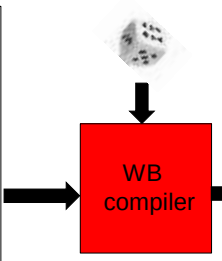
White-Box implementation WB(E_K)

keyed instance E_K

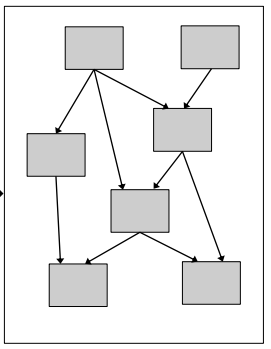


intelligible representation of E_K

e.g. the value of K



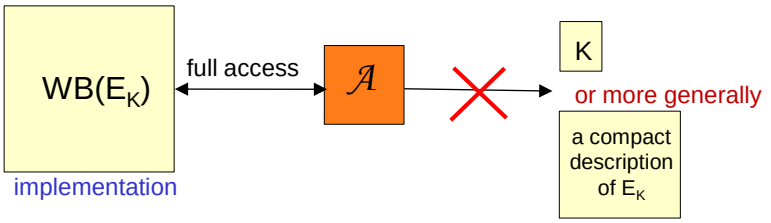
implementation WB(E_K)



obfuscated representation / program for E_K

adversaries' objectives: weak WB security

- weak white-box security (WWB) captures the requirements of [CEJO02]



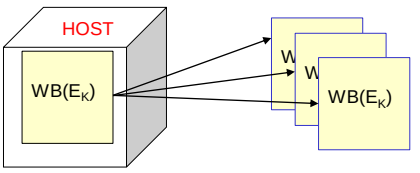
informal definition: (T, S)-incompressible implementation of E_K.
an adversary with full access to WB(E_K) must be unable to derive
an equivalent* representation of E_K of size lower than S in time T. **

* probabilistic variant: equivalent on a non-negligible fraction p of inputs

** if we drop the requirement that E_K must possess a short key, weak WB may become much easier to achieve by an using a block cipher E_K without compact description

discussion: code lifting

- WWB does not prevent code lifting
  i.e. the extraction / duplication / execution of WB(E_K)



- however code lifting
  - can be a lesser threat than key extraction due to the size of WB(E_K) - typically MBytes
  - can potentially be thwarted by complementary techniques not detailed here, e.g.
    - marking techniques for rendering WB(E_K) traceable
    - « node locking » e.g. using equipment indentifier as part of WB(E_K)
    - blurring the boundary between the WB(E_K) code for E_K and other parts of the code
      external encodings i.e. replacing WB(E_K) by WB(G o E_K o F) could facilitate this

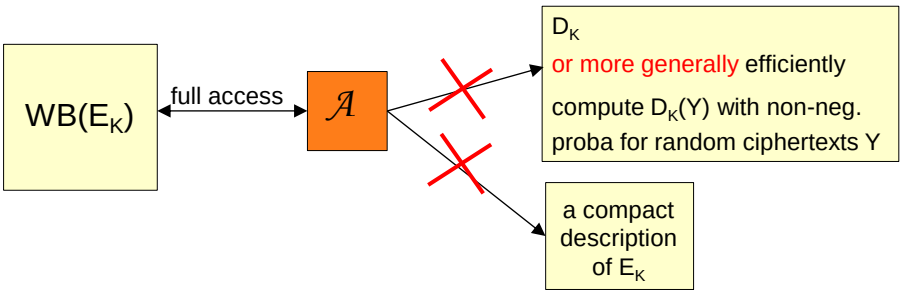
- conclusion on WWB: not a fully selfstanding notion but seems relevant in practice

adversaries' objectives: strong WB security

- strong white-box security (SWB)

  ~ resistance to plaintext recovery* + incompressibility of $WB(E_K)$

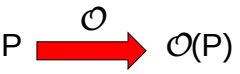
  * this requirement was included in formalisation efforts such as [SWP08, DLPR13]



- this implies that (P, S) = $(WB(E_K),D_K)$ is a public key encryption scheme
- of course SWB does not prevent code lifting that remains a major practical issue

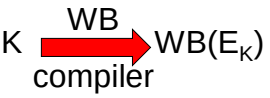
related notion: program obfuscation

whose extensive study was launched by the seminal paper of [Barak et al. 01]



- same functionality as P
- virtual black box property: ultimately aims at ensuring that « anything that can be efficiently computed from $\mathcal{O}(P)$, one could also compute given oracle access to P » [B+01]

suggests by analogy:



- same functionality as E_K
- white-box property [SWP08]: ultimately something like « any attack that can be achieved from full access to $WB(E_K)$ one could also achieve given oracle access to E_K »

- known impossibility results on obfuscation give no evidence that SWB is unimplementable
- [SWP08] and [DLPR13] give some evidence that WB is implementable for specific sym. cipher examples based on PK schemes [under some security definitions and assumptions]

digression: hypothetic extended WB features (1/2)

- traceable WB (could help thwarting code lifting)

  ~ an identifier ID – e.g. the owner's id – is embedded in WB (E_K) by the compiler

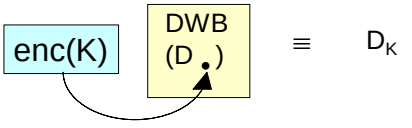
  ID must be recoverable by the compiler from any equivalent representation of $WB_{ID}(E_K)$ an adversary can produce

  - weak flavour: for a given key K the compiler produces one single implem. $WB_{ID}(E_K)$
  - strong flavour (traceable block cipher)
    - for a given key K the compiler produces several implementations $WB_{ID[i]}(E_K)$
    - the scheme must resist collusion attacks

digression: hypothetic extended WB features (1/2)

- dynamic WB (could help using WB with session keys)

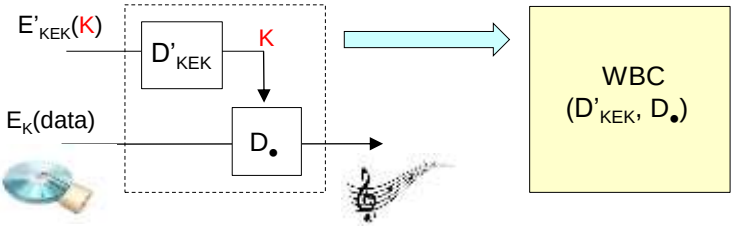
  ~ non-instantiated WB implem. of (D_K) or (E_K) parametrized by an encoded key



- WB composition (could help preventing content key leakage in DRM)

  ~ implement composition of crypto functions as not to leak intermediate values,

  e.g. content / payload decryption key K in DRM application



obfuscation method for WB- AES (sketch)

to get a WB implementation of a keyed block cipher instance E_K

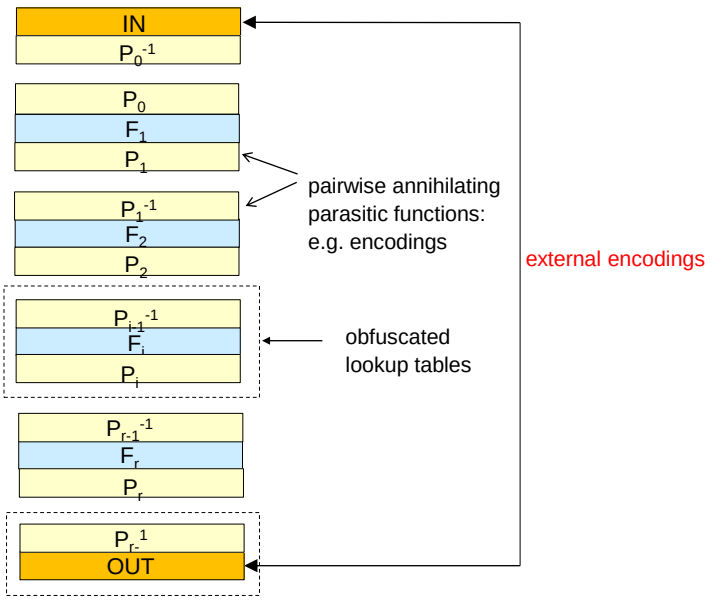
- decompose E_K into simple layers

  e.g. rounds or subrounds that can be split into lookup tables (and xors)

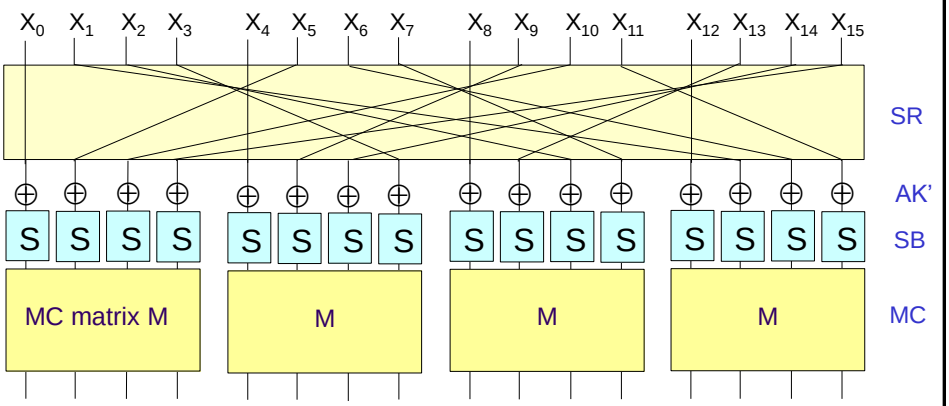
- compose these layers with pairwise annihilating parasitic functions
  - two kinds of parasitic functions
    - encodings: small linear or non-linear bijections
    - mixing bijections: large linear bijections
  - the lookup tables (and xors) that build up each layer are obfuscated by composition with these functions

- external encodings
  - implement OUT o E_K o IN instead of E_K - where OUT and IN are secret bijections
  - main motivation: avoid naked initial / final rounds w/o input / output encodings that would render WB(E_K) extremely insecure

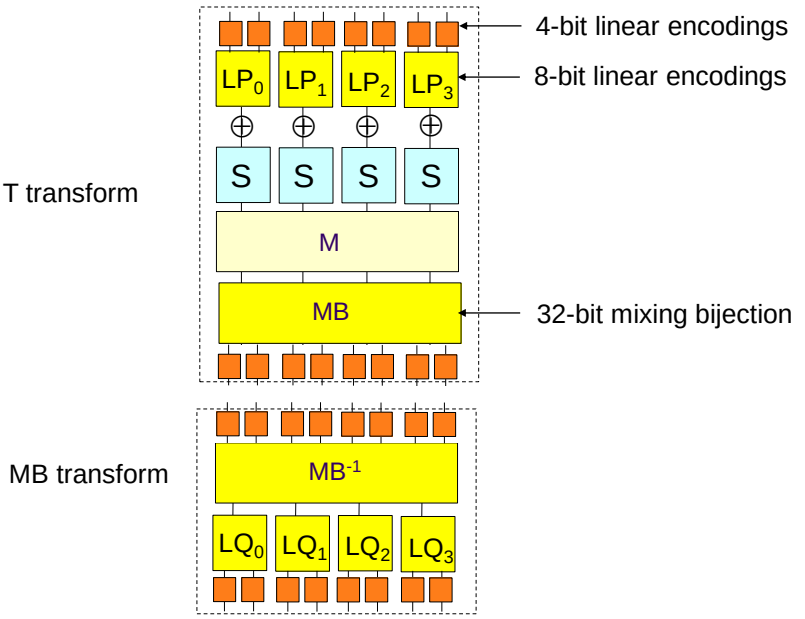
WB obfuscation method (rough sketch)



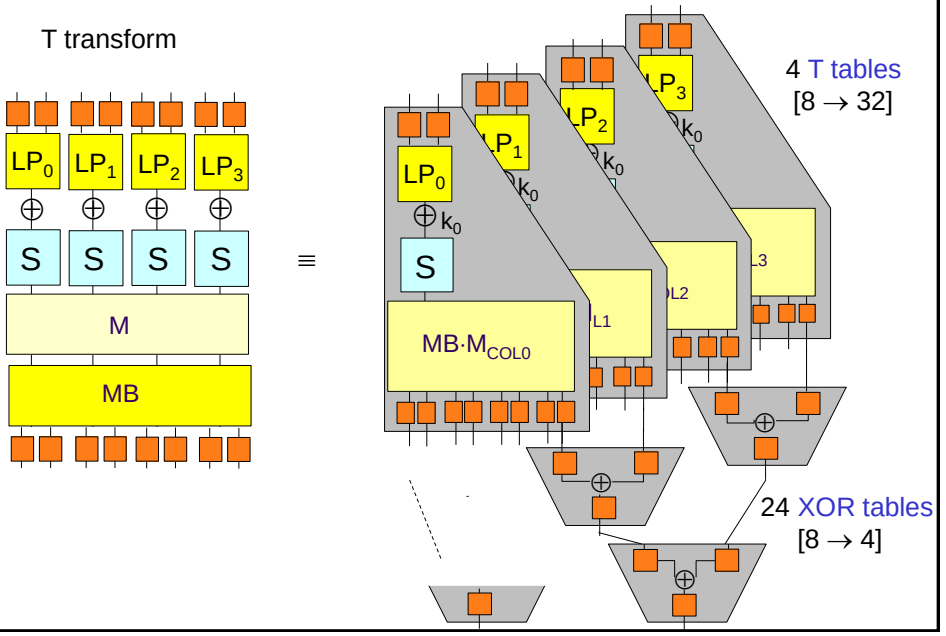
reminder: one AES round



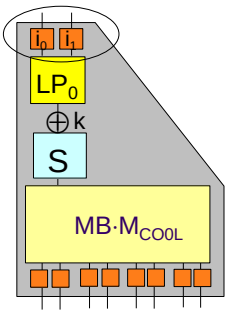
AES-WB obfuscating one quarter round



conversion into a network of tables (1/2)



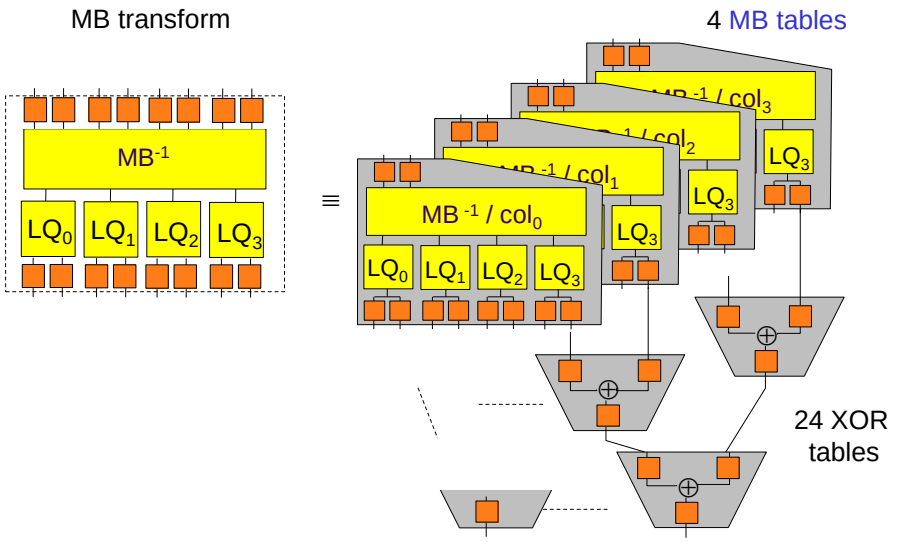
local security of T tables



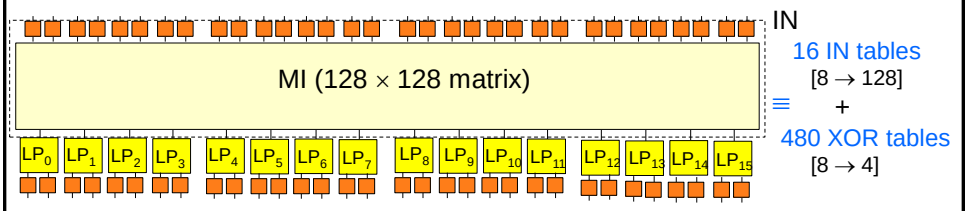
T-tables contain no information on the key byte k they embed

due to the masking effect
of the pair of 4-bit encodings (i_0, i_1)

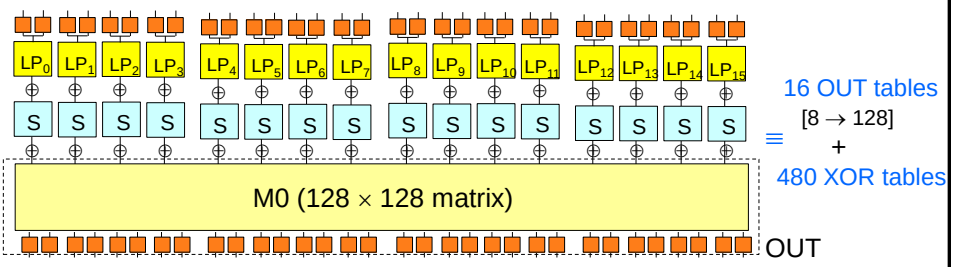
conversion into a network of tables (2/2)



extra tables for external encodings



[IN + output encodings]



[input encodings + round 10 + postwhitening + OUT]

WB-AES: summary

- $9 \times 4 \times 4$ = 144 T tables
- $9 \times 4 \times 4$ = 144 MB tables
- 16 IN tables
- 16 OUT tables
- $9 \times 4 \times 48 + 2 \times 480$ = 1728 XOR tables

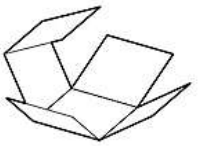
total size:
770 048 bytes

WB implementations and their analysis

WB implementations	Cryptanalysis
DES	
[Chow et al. 02] seminal DES-WB	[Jacob et al. 02, Link et al. 05] attacks on naked DES-WB]
	[Goubin et al 07, Wyseur et al. 07] attack on DES-WB
AES	
[Chow et al. 02] seminal AES-WB	[Billet et al. 04] BGE attack on AES-WB
[Bringer et al. 06] variant	[Michiels et al. 08] generic attack on SPN WB
[Xiao-Lai 09] variant	
[Karroumi 10] variant	[De Mulder et al. 10] attack on Bringer et al. variant [De Mulder et al. 12] attack on Xiao-Lai variant [Lepoint et al. 13] 2 improved attacks on AES-WB + attack on Karroumi's variant

❷ cryptanalytic toolbox for WB-AES and sisters

i.e. WB(E_K) implementations based on a network of lookup tables

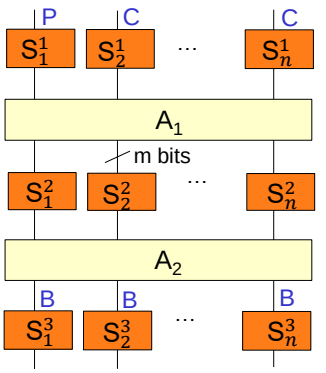


warming up: 2 multi-purpose tools from structural cryptanalysis

structural cryptanalysis of SASAS [BS01]

- problem: break a generic SPN that alternates
  - S layers: n secret bijective m-bit S-boxes
  - A layers: secret affine permutations
- initial step
  - encrypt appropriate 2^m-block structures
  - at the input of the last S layer all words have a zero sum (B property)

  → last S layer up to unknown affine bijections
- overall complexity: $T \approx n \cdot 2^{3m}$

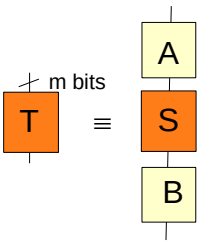


techniques for solving « subproblems » such as ASA and SAS are also introduced

e.g. Biham's low rank detection technique for ASA, differential technique for SAS

linear/affine equivalence algorithms [PGC98, BCBP03]

- LE and AE problems
  - given: two known m-bit bijections S and T
  - find: two linear / affine bijections A and B if any such that T = B o S o A

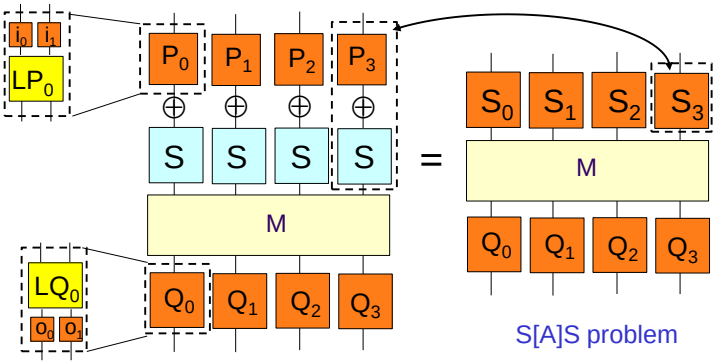


- solving technique: « to and fro », « needlework » …
  - partial assumptions for A values determine B on a subspace etc. …
- complexity : $O(m^3 2^m)$ for LE, $O(m^3 2^{2m})$ for AE [BCBP03]
  - i.e. low complexity for S-box sizes such as m = 8 or 16

BGE attack on WB-AES [Billet et al. 04]

- starting point

  if instead of considering isolated tables
  one considers a suitable composition of T, MB and XOR tables
  one gets an obfuscated quarter round with unknown byte encodings P_i, Q_i



S[A]S problem

- retrospective remark

  deriving candidate permutations S_i and Q_i is essentially a S[A]S problem!

BGE attack – outline

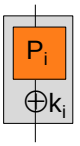
- phase 1: recover the « non-affine parts » of the P_i and Q_i
  - i.e. recover the Q_i and the P_i up to unknown affine permutations

  ~ solve the former S[A]S problem

  complexity < 2^{30}
- phase 2: fully recover the residual affine parts of the Q_i

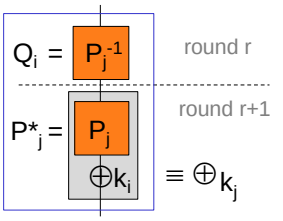
  and the residual affine parts the P_i up to unknown constants
  - i.e. recover the Q_i and the P^*_i =

    and reiterate this for round r+1



complexity < 2^{29}

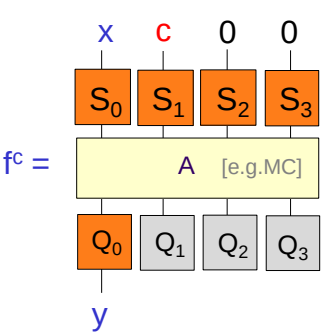
- phase 3: recover the k_i from
  - the Q_i or round r that are the P_j^{-1} of round r+1:
  - the P^*_j of round r+1



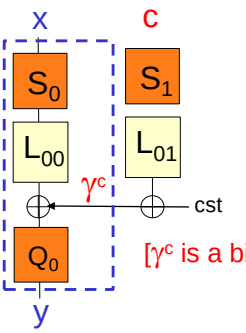
overall complexity ≈ 2^{30}

BGE phase 1 – yet another SAS technique

- aim: recover the non-linear part of the Q_i in the SAS function
- notation: lin.part of A = $\begin{pmatrix} L_{00} & L_{01} & L_{02} & L_{03} \\ L_{11} & L_{11} & L_{12} & L_{13} \\ L_{20} & L_{21} & L_{22} & L_{23} \\ L_{30} & L_{31} & L_{32} & L_{33} \end{pmatrix}$
- assumption: all m × m binary matrices L_{ij} are bijective [as for MC]
- we consider the 2^m [e.g. 256] permutations:



=

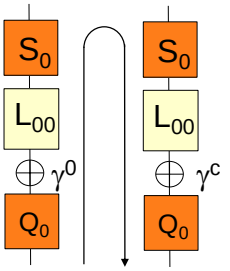


[γ^c is a bijection of c]

BGE phase 1- deriving a group G of permutations g^c

- to recover Q_0, we derive the following set G of 2^m permutations g^c

$g^c = f^c \circ (f^0)^{-1} \equiv$



$g^c = Q_0 \circ \oplus_\gamma \circ Q_0^{-1}$

where $\gamma = \gamma^c \oplus \gamma^0$

[conjugate of $\oplus_\gamma$ under Q_0]

- φ: $(G = \{g^c\}, \circ) \rightarrow (\{0,1\}^m, \oplus)$ $\quad g^c \mapsto \gamma$

  is a group isomorphism since $g^c \circ g^{c'} = Q_0 \circ \oplus_{\gamma \oplus \gamma'} \circ Q_0^{-1}$

claim 1: we can efficiently recover φ up to an unknown linear mapping

claim 2: this allows to recover Q_0 up to an unknown affine mapping

BGE phase 1 - group isomorphism technnique

- to construct an isomorphism ψ: $(G,\circ) \rightarrow (\{0,1\}^m,\oplus)$ we gradually
  - build a « basis » $(g_1, .., g_m)$ of $(G, \circ)$
  - express each element g of G as a product of basis elements

    $g = g_1^{\varepsilon 1} \circ g_2^{\varepsilon 2} \circ \cdots \circ g_8^{\varepsilon 8}$ [where $g_i^0 = id$, $g_i^1 = g_i$]
  - set $\psi(g) = \varepsilon_1 e_1 \oplus \varepsilon_2 e_2 \oplus \cdots \oplus \varepsilon_8 e_8$

    where $(e_1, .., e_m)$ denotes the canonical basis of $\{0,1\}^m$
- $\varphi = L \circ \psi$ where L is an unknown linear mapping
- ψ provides Q_0 up to an unknown affine mapping A of linear part L

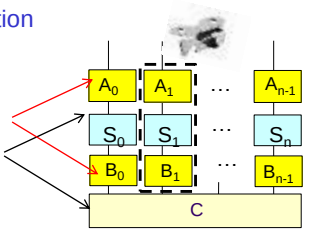
  $\exists A \; \forall g \quad g(0) = Q_0 \circ A(\psi(g))$ - this determines Q_0 up to A
- Tolhuizen's improvement [T12]
  - allows to accelerate the derivation of ψ and $Q_0 \circ A$ by a substantial factor

a more generic attack [MGH08]

- in [MGH08] Michiels, Gorissen, and Hollmann
  WB implementations under the techniques of [Chow et al.]
  of a generic class of SPN ciphers are shown to be vulnerable to a 3-step attack.

- step 1: the group isomorphism technique of BGE phase 1
  is shown to work - under a mild condition on the diffusion matrix
  → one recovers and peels off the non-linear part of all encodings

- step 2: derive the following equivalent representation
  of the round function with affine encodings



- step 3: use affine equivalence techniques
  from [BCBP03] on rounds r and r+1
  to get the r^{th} round key

improved WB-AES attaks [LRMRP13] (1/2)

- attack 1 [developed by De Mulder, Roelse, and Preneel]
  same starting point as BGE and similar overall structure, but:

  - in phase 1 Tolhuizen's speed-up [T12] of the isomorphism technique is applied
    → complexity: 2^{19} inst. of 2^{30} → the subsequent phase becomes time-critical

  - in phases 2 and 3 (derivation of affine encodings and round key bytes)
    - an accelerated testing technique for determining the Q_i is introduced
      → complexity: 2^{22} inst. of 2^{29}
    - a faster recovery procedure for the key bytes of round r+1 is also introduced

  - a phase 4 is added to tackle permutations of encoded bytes if any. It details
    - how to reorder the round key bytes before recovering the key
    - how to recover the external encodings

→ the overall complexity is reduced from 2^{30} to 2^{22}

improved WB-AES attaks [LRMRP13] (2/2)

- attack 2 [developed by Lepoint and Rivain]

  same starting point as BGE: S[A]S problem
  but entirely different structure and techniques

  - step 1: the secret mappings S_i are recovered

    a. exploit internal collisions on the input of [e.g.] Q_0

    to determine [e.g.] S_0 and S_1 up to 2 unknown bytes λ and μ

    $\rightarrow$ one gets S'_0 - the composition of S_0 with $x \mapsto \lambda \cdot x \oplus \mu$

    b. algebraic testing of the composition of S'_0 with trial permutations

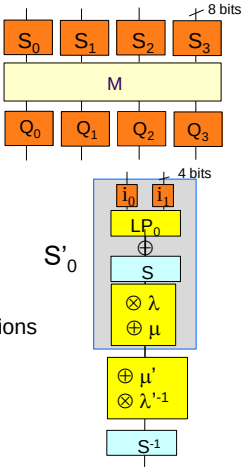
    $\rightarrow$ if $\mu'=\mu$ and $\lambda'=\lambda$, then the degree we get is only 3

    $\rightarrow$ this is easy to test and provides λ and μ and thus S_0

  - step 2: the next round bytes k_i are recovered

    the Q_i and their inverses are easy to derive from the S_i

    $\rightarrow$ a simple algebraic test then provides the next round key bytes



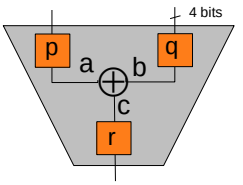
$\rightarrow$ more simple structure + overall complexity reduced to $\approx 2^{22}$

another Achille's heel of WB-AES

- XOR tables are vulnerable to SAS attacks

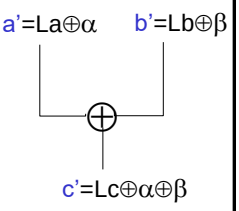
  - for instance to the isomorphism technique of [BGE04]

    as pointed out in [Wyseur 07]



  - this typically provides p, q, and r

    up to an unknown 4-bit linear bijection L and 2 nibbles α, β



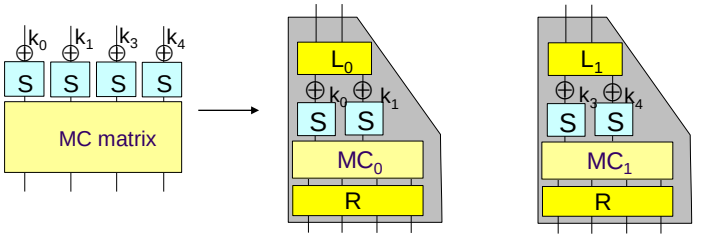
  - one is left with a partly de-obfuscated AES representation

  - [to be further checked]

    this might lead to a speed-up of the former attacks

[XL09] variant and the attack of [De Mulder et al. 12]

- some features of the [XL09] scheme
  - larger size: ≈ 20 Mbytes, exclusive use of larger linear encodings, no xor tables
  - quarter rounds are reflected in pairs of 16-bit to 32-bit tables (T-tables)



  - R, L inverses and ShiftRows embedded in one 128 x 128 matrice per round

- some features of the attack of [MRP12]
  - linear equivalence techniques partly inspired from [BCBP03] are applied to T tables
  - the attack recovers the key bytes +the external encodings
  - workfactor: about 2^{32}
  - with affine encodings the scheme would remain vulnerable to the generic attack of [MGH08]

summary on WB-AES/DES-like implementations

- networks of lookup tables seem vulnerable

  to the combined used of structural cryptanalysis techniques such as
  - SAS attack methods [BS01 and internal collisions, isomorphism technique…]
  - linear/affine equivalence algorithms, algebraic testing, etc.

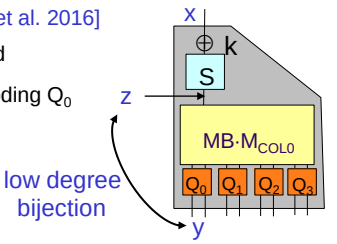
  inputs to tables embedding NL components of E_K (S-boxes, etc.) seem to be a bottleneck

- w/o external encodings this vulnerability is considerably amplified
  - local security is typically lost
  - this could at least partly explain why standard grey box analysis (e.g. DPA) tools seems « blindly applicable » to naked public WB implementations as reported for instance in [Bos et al.; Sanfelix et al.; Sasdrich et al. 2016]
  - assume for instance that a naked T-table is used

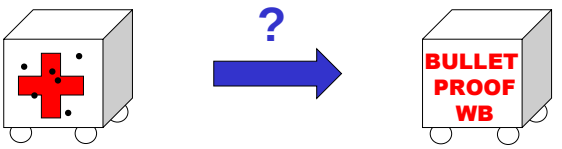
    for many values of MB and the WB-AES encoding Q_0

    some output bits y_j are correlated

    to some bits z_i of the « sensitive » byte z



❸ alternative approaches and perspectives



can white box crypto and security w/o obscurity be reconciliated ?

I will consider this issue for:

- WWB: incompressible white box
- SWB: strong (plaintext rec. resistant + incompressible) white box

perspectives for Weak WB cryptography (1/2)

[i.e. a compact description of E_K, e.g. K should be difficult to extract from WB(E_K)]

A. open question if we require that E_K be a common cipher, e.g. AES
   - alternatives to networks of LUTS might potentially come from the combined use of polynomials with GF(2^n) coefficients and multivariate techniques
     - lesser constraints on number of input variables than for LUTS
     - one attempt along these lines [BCD06] was attacked in [De Mulder et al. 2010] but it would be premature to draw general conclusions

perspectives for Weak WB cryptography (2/2)

[i.e. a compact description of E_K, e.g. K should be difficult to extract from $WB(E_K)$]

B. seems achievable if we relax the requirement that E_K be a common cipher and do not require that a compact and fast implementation of E_K still exists

- looks simple at first glance: use for instance large random or pseudo-random S-boxe
- it might be be worth investigating constructions with strong security arguments

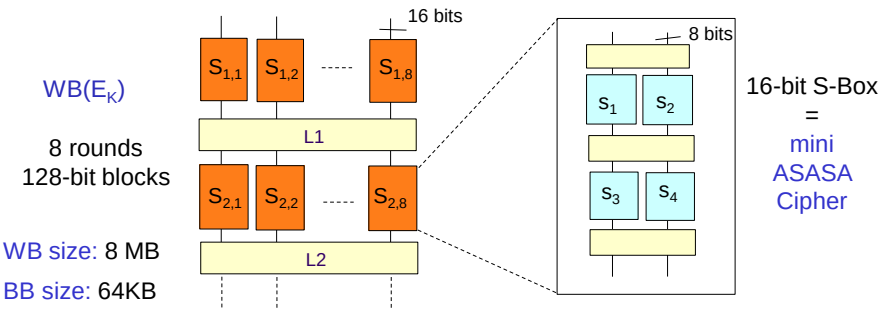
C. seems still achievable with some extra caution if we relax the requirement that E_K be a common cipher but require that outside from the WB implementation a compact and not too slow implementation of E_K still exists

- case study 1: candidate memory-resistant block ciphers of [BBK14]
- more recently, two other families of candidate memory-resistant block ciphers were proposed: SPACE and N-SPACE [BI15]

WWB block ciphers family of [BBK14]

- WB-ASASA approach: use ASASA[…] S-boxes to thwart structural cryptanalysis

for instance:



- BB-ASASA: a non-WB block cipher with an ASASA structure is also proposed (128-bit blocks, secret bijective 8-bit S-Boxes and 128-bit L layers)

advances in structural cryptanalysis triggered by [BBK14]

- [Minaud, Derbez, Fouque, Karpman 2015]
  - generic algebraic attack on ASASA structure that breaks BB-ASASA
  - « differential-linear » attack on mini ASASA block ciphers, e.g. n=16 bits
- [Dinur, Dunkelman, Kranz, Leander 2015]
  - structural attacks on mini ASASA blockciphers
    (e.g. n = 16 to 24 bits, generic bijective 8-bit S-boxes)
- [Biryukov, Khovratovich 2015]
  - attacks on blockciphers with up to 9 secret layers, e.g. SASASASAS
    with sufficiently small S-boxes, using [Boura, Canteaut 2011] and other techniques

- these results represent a break-through in structural cryptanalysis
- they invalidate many WB-ASASA instances, but not the WB approach of [BBK14]
  provided that sufficiently complex structures ASASASA[++] are used

perspectives for SWB cryptography

$(WB(E_K), D_K)$ must be a PKE scheme, $WB(E_K)$ must be incompressible

A. challenging open question if we require that E_K be a common block cipher, e.g. AES
   - potential direction: use of polynomials with GF(2^t) coefficients and multivariate techniques ?
   - far from reach from the current state of the art of multivariate cryptography

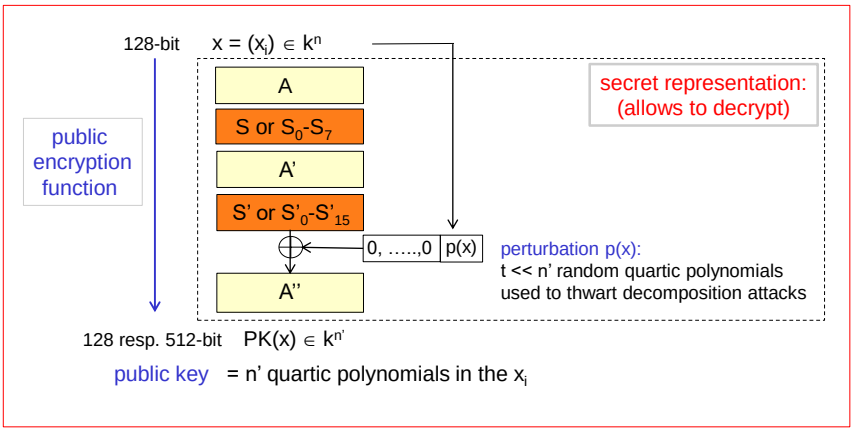
B. seems achievable if we relax the requirement that E_K be a common block cipher
   - PKE does not automatically provide a compression resistant solution
   - candidate SWB implementation [DLPR13]: one-time random blinding of an RSA exponent e into a larger equivalent exponent $e'=e+r\cdot\varphi(N)$. [Caveat: security is lost if e and d are not both kept secret, or if e is blinded twice, etc.]

C. exciting open question if we require that E_K be based on block cipher ingredients
   - this would solve a long standing challenge:
     design a public key encryption scheme with block cipher features
   - case study 2: the two asymmetric / strong WB design attempts of [BBK14]

case study 2: public key encryption schemes of [BBK14]

- two PK encryption / strong WB schemes were proposed in [BBK14]

  in both cases: ASASA structure with secret k-affine A layers, secret k-quadratic S layers
  - χ-scheme: k = GF(2); one large S-box per S layer
  - ASASA scheme with expanding S-boxes: k=GF(16); several S-boxes per S layer



attacks on the PK schemes of [BBK14]

- [Minaud, Derbez, Fouque, Karpman 2015]

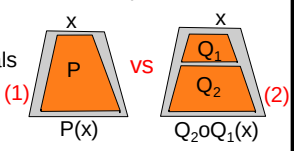
  key recovery attack on the χ-scheme
  - low complexity $\approx 2^{39}$
  - an adaptation of the generic algebraic attack on ASASA from the same paper

- [G., Treger, Plût 2015]

  key recovery attack on the ASASA scheme with expanding S-boxeq
  - low complexity ≈ rank computation for 2^{26} matrices of size 64×96 over F_{16}
  - starting point: an algebraic distinguisher between

    (1) a random input expanding system P of quartic polynomials

    (2) a composition of quadratic polynomials Q_1, Q_2



- none of these attacks necessarily invalidates all ASASA PK schemes

  however they teach that low diffusion and/or expanding S layers should be avoided

conclusions

- no overoptimism for current situation
  - WB crypto still largely remains a subarea of security by obscurity
  - can provide: some extra protection in systems where the single alternative is essentially deploying the algorithms and secret keys « in plain » in software
  - caution: WB should not be viewed as a « drop-in replacement » of solutions based on the combination of tamper-resistant modules (e.g. smart cards, TPM) and grey-box implementation techniques, that are more mature
- more optimistic view of future perspectives
  - provided that a sufficient open research effort is spent on WB crypto
    - definitional issues, security models, links with other notions, etc.
    - usual triptych: candidate constructions / security arguments / cryptanalysis

    answers to some of the former open questions seem reachable
  - side benefits might include
    - extending the toolbox of structural and algebraic cryptanalysis (this has begun)
    - extending the toolbox of grey-box security techniques
    - bringing symmetric crypto closer to asymmetric crypto