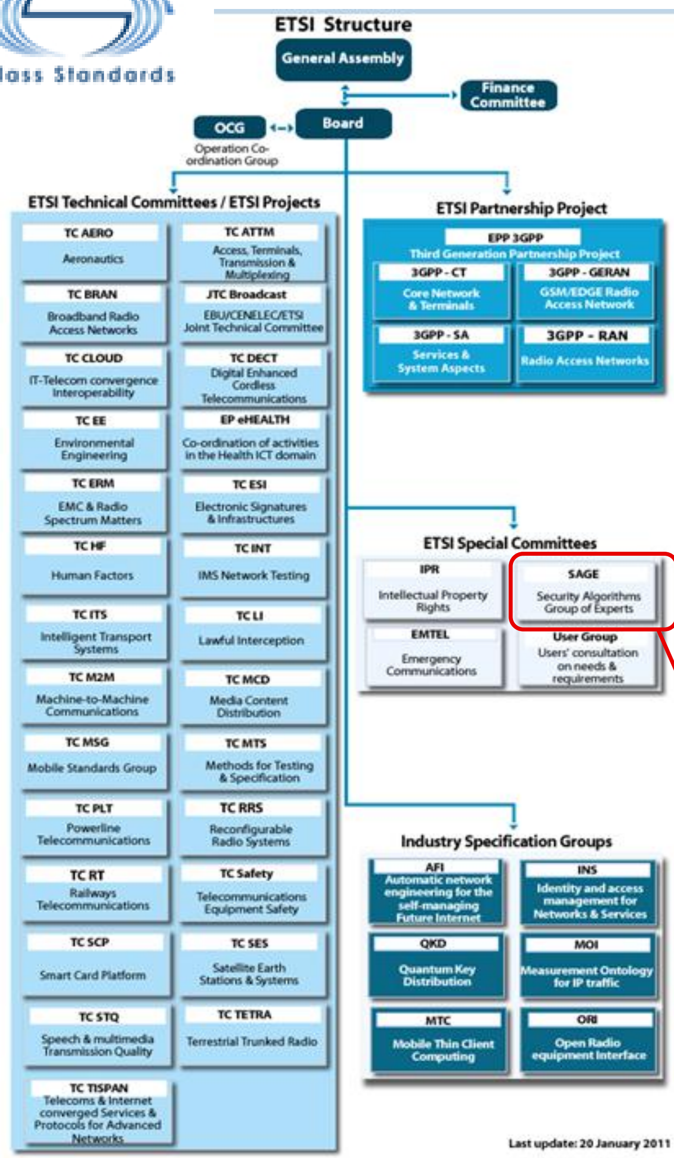
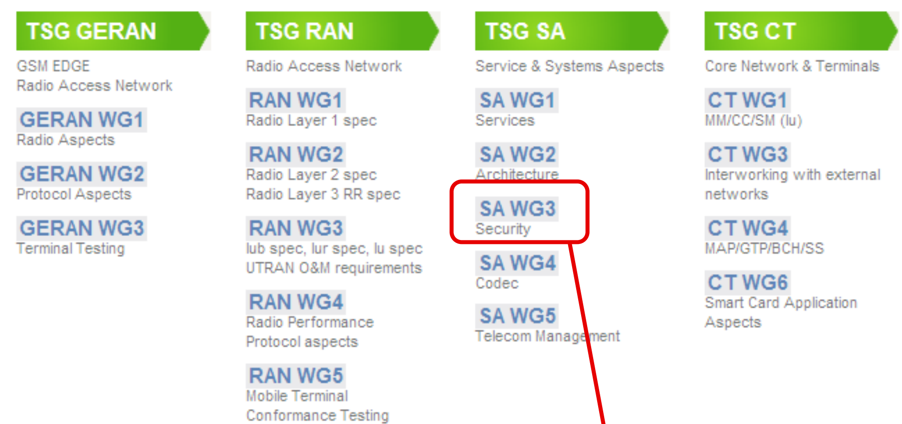
Choosing new authentication and key generation algorithms for mobiles

Steve Babbage
Vodafone Group R&D

14-18 January 2013

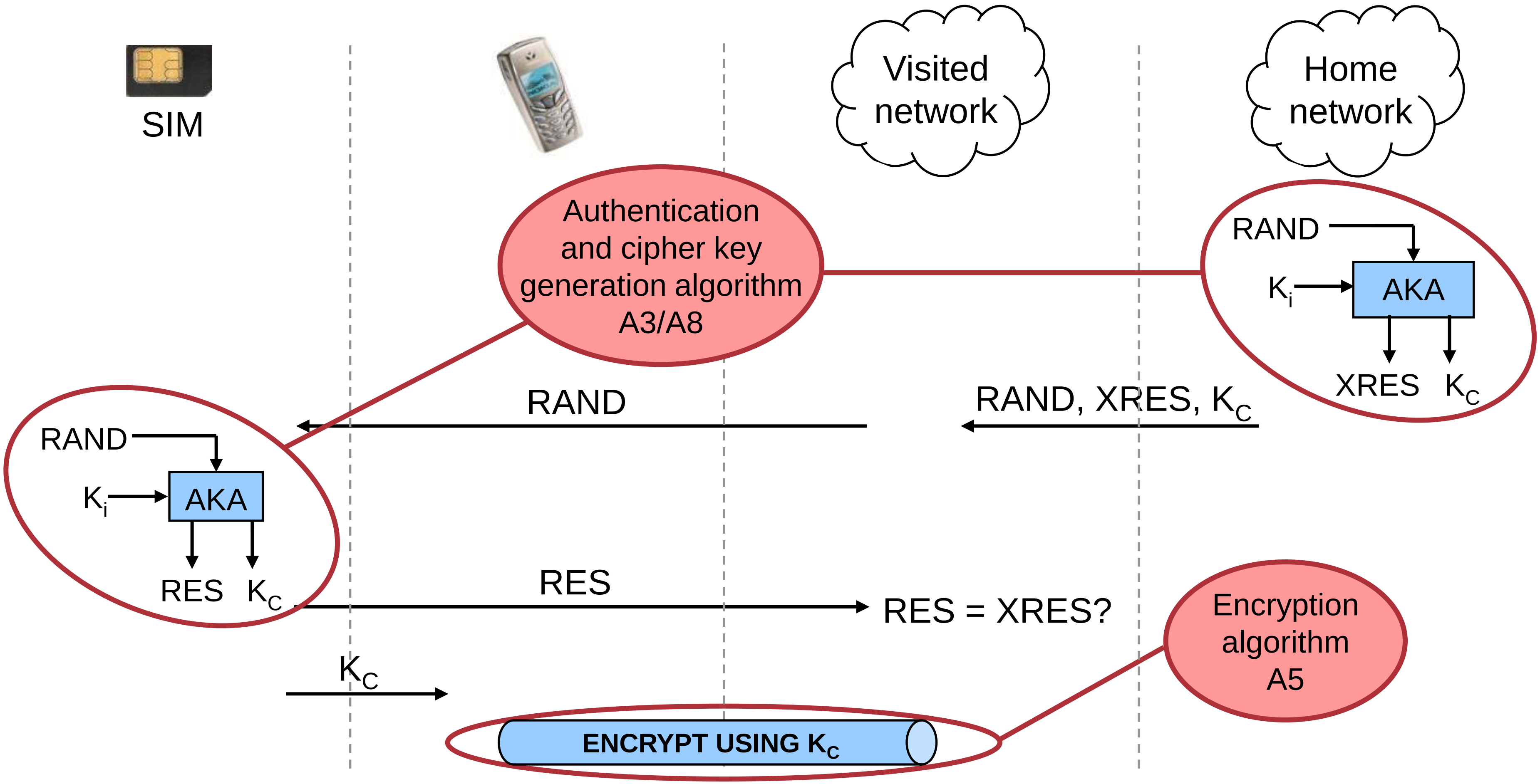
Standards groups





First generation

GSM security architecture



GSM security limitations

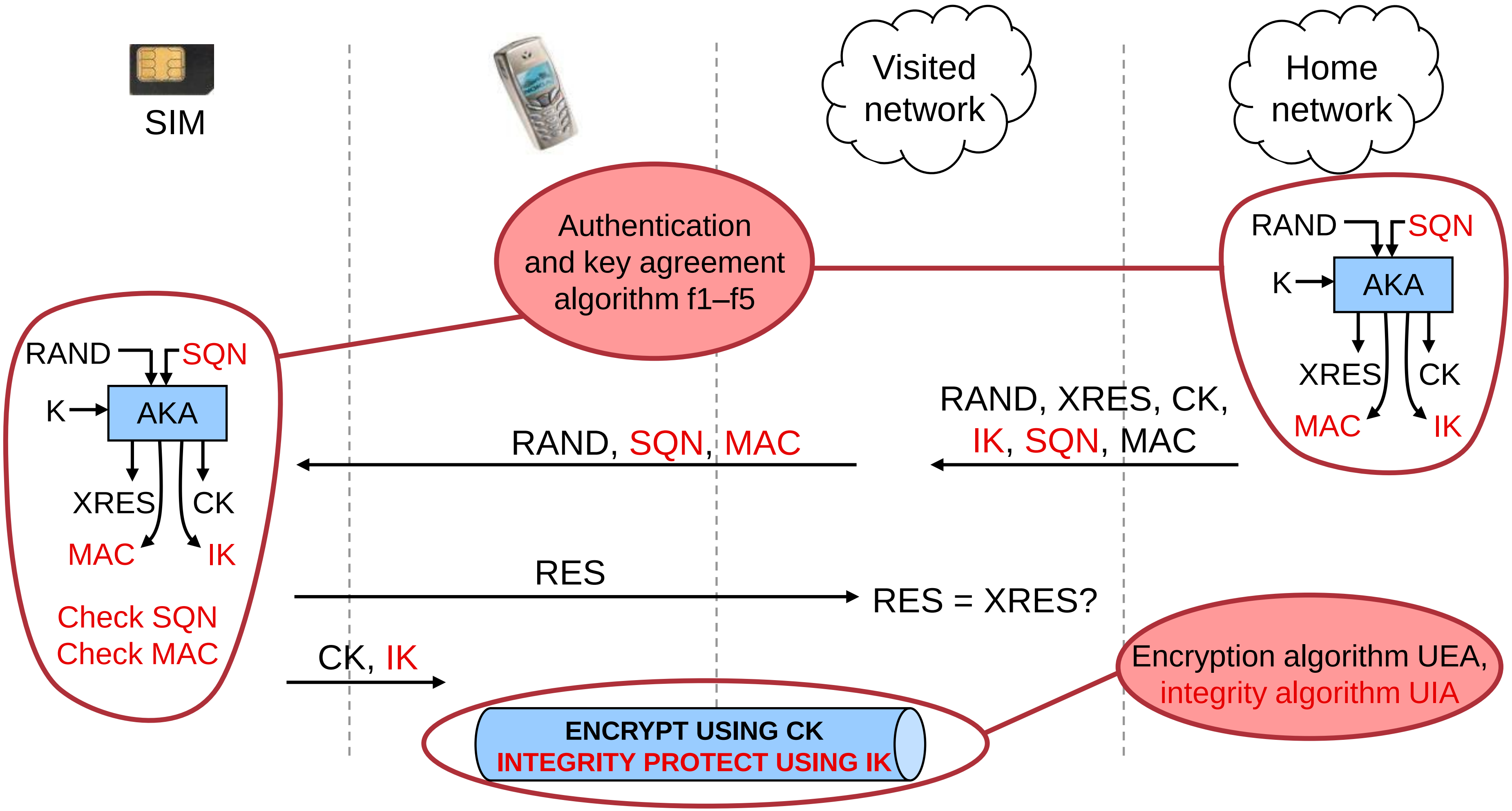
> Key length

> One-way authentication

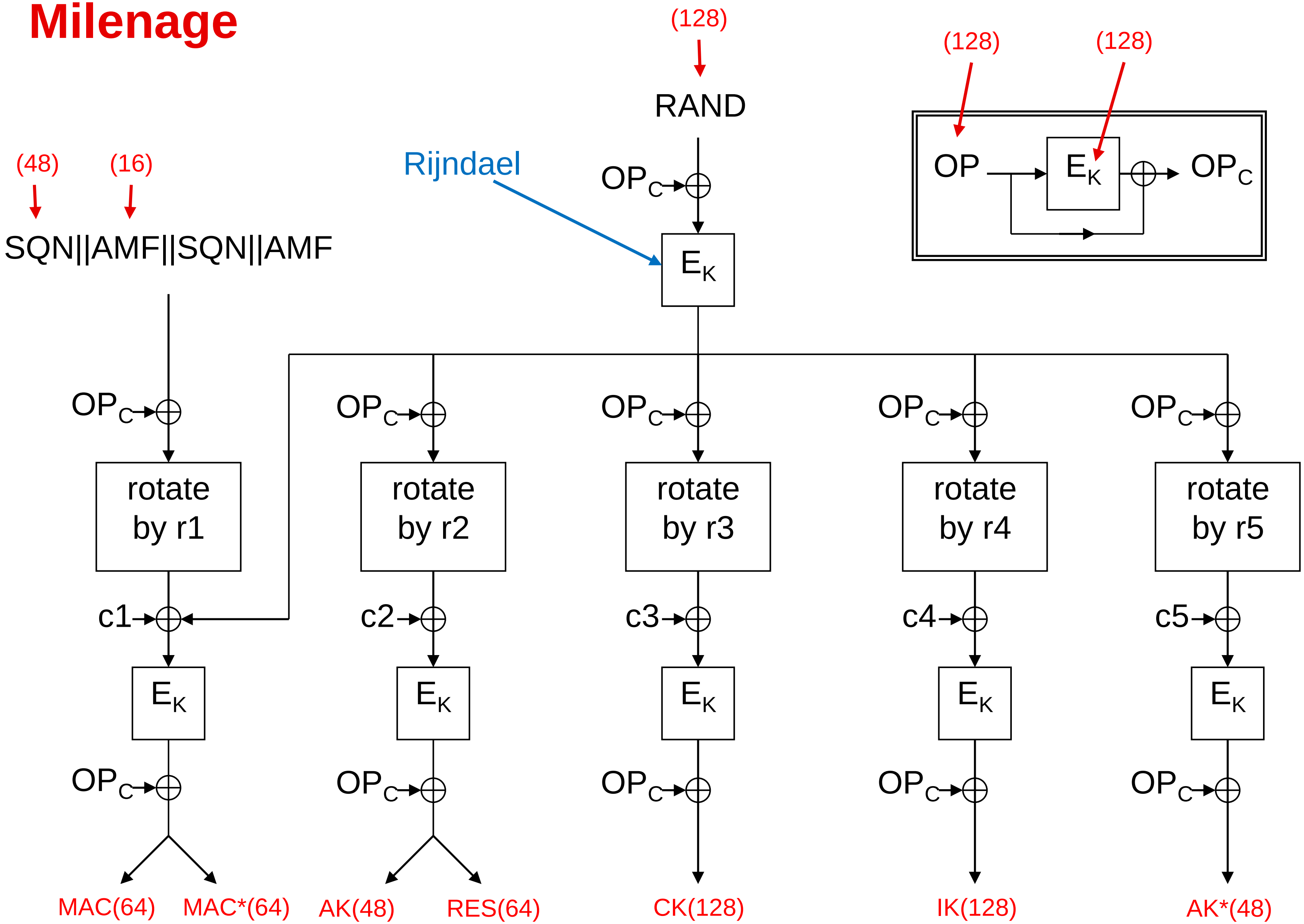
> Unprotected signalling

> A5/1, A5/2

UMTS security architecture (slightly simplified)



Milenage



(128)
RAND
(128)
(128)
OP
E_K
OP_C
Rijndael
(48)
(16)
SQN||AMF||SQN||AMF
OP_C
E_K
OP_C
OP_C
OP_C
OP_C
OP_C
rotate by r1
rotate by r2
rotate by r3
rotate by r4
rotate by r5
c1
c2
c3
c4
c5
E_K
E_K
E_K
E_K
E_K
OP_C
OP_C
OP_C
OP_C
OP_C
MAC(64)
MAC*(64)
AK(48)
RES(64)
CK(128)
IK(128)
AK*(48)

Allow customising

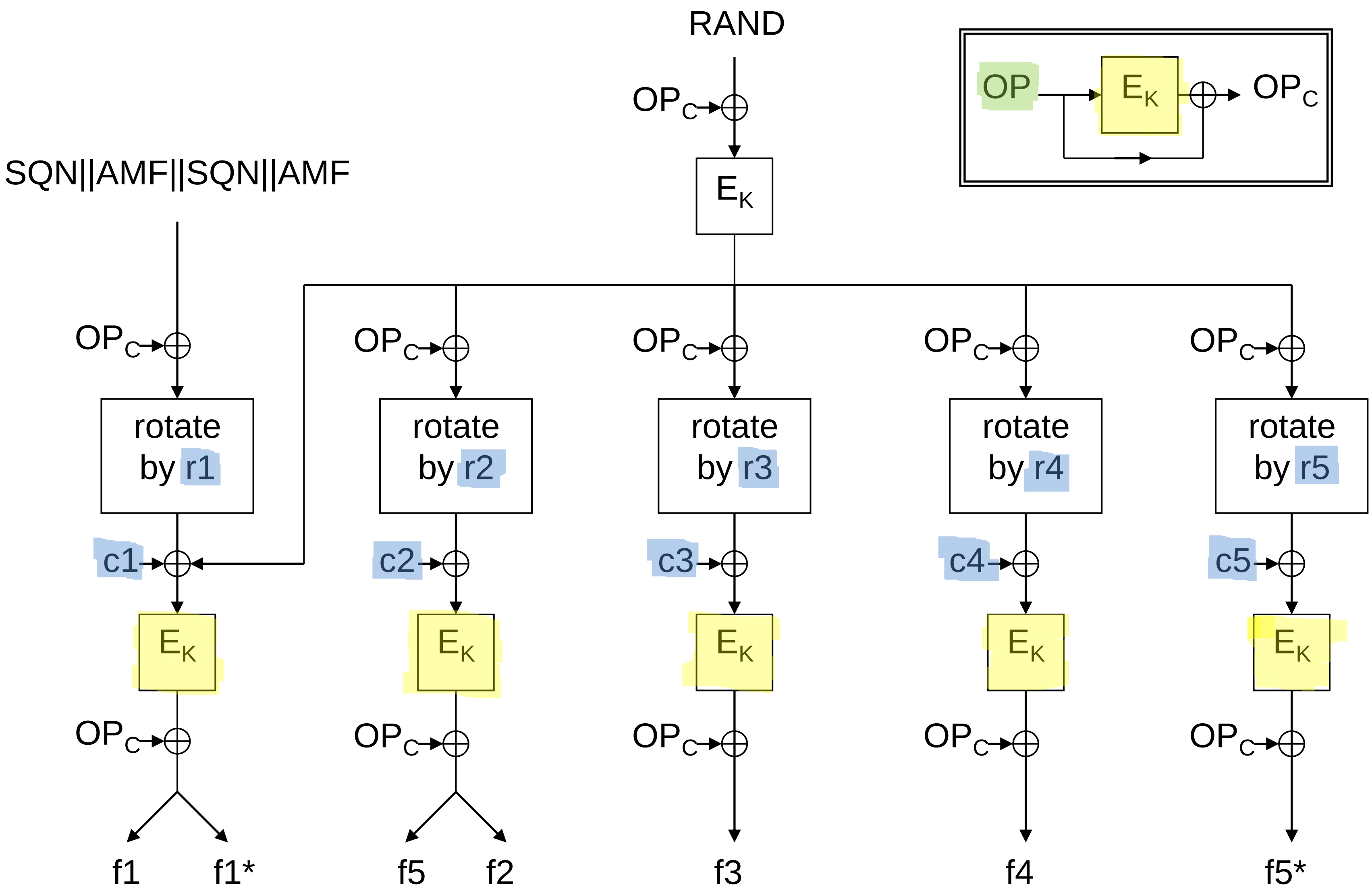
> To make each operator's implementation different

> To prevent USIMs for operators being interchangeable, either through trivial modification of inputs and outputs or by reprogramming of a blank USIM

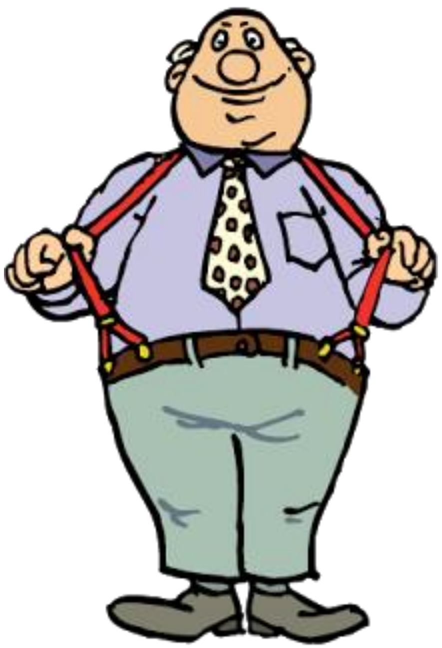
> To keep some algorithm details secret

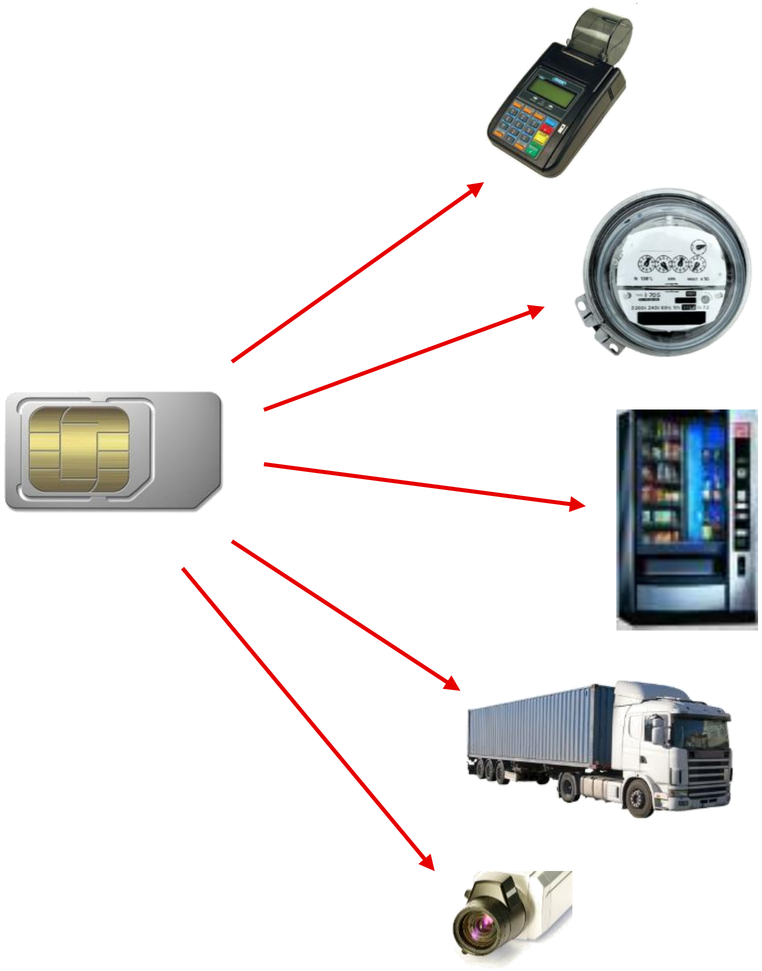
– Makes some attacks a little harder

Customising Milenage



Why do we want a second algorithm?





Design considerations

> Trusted kernel

– Don’t want to take several years

> Security proof, assuming the kernel is strong

> Can be a drop-in replacement for Milenage …

– … or can accommodate a 256-bit key

> Efficiency

– 8-bit out

– 16-bit, 32-bit in

> Fundamentally different from Milenage

> Side channel attacks

Candidates for a block cipher kernel

	ARIA	BLOWFISH	CAMELLIA	CAST	CLEFIA	IDEA	MARS	RC6	SEED	SERPENT	TEA, XTEA	TWOFISH
May fall with AES (XSL-type attacks)	x		x							x		
< 256 bit key						x			x		x	
Key-dependent S-box		x										x
32-bit multiplications							x					
IPR				x	x			x				

> No need for kernel to be invertible

> Hash functions are a natural alternative to block ciphers

Candidates for a hash function kernel

SHA-2

BLAKE

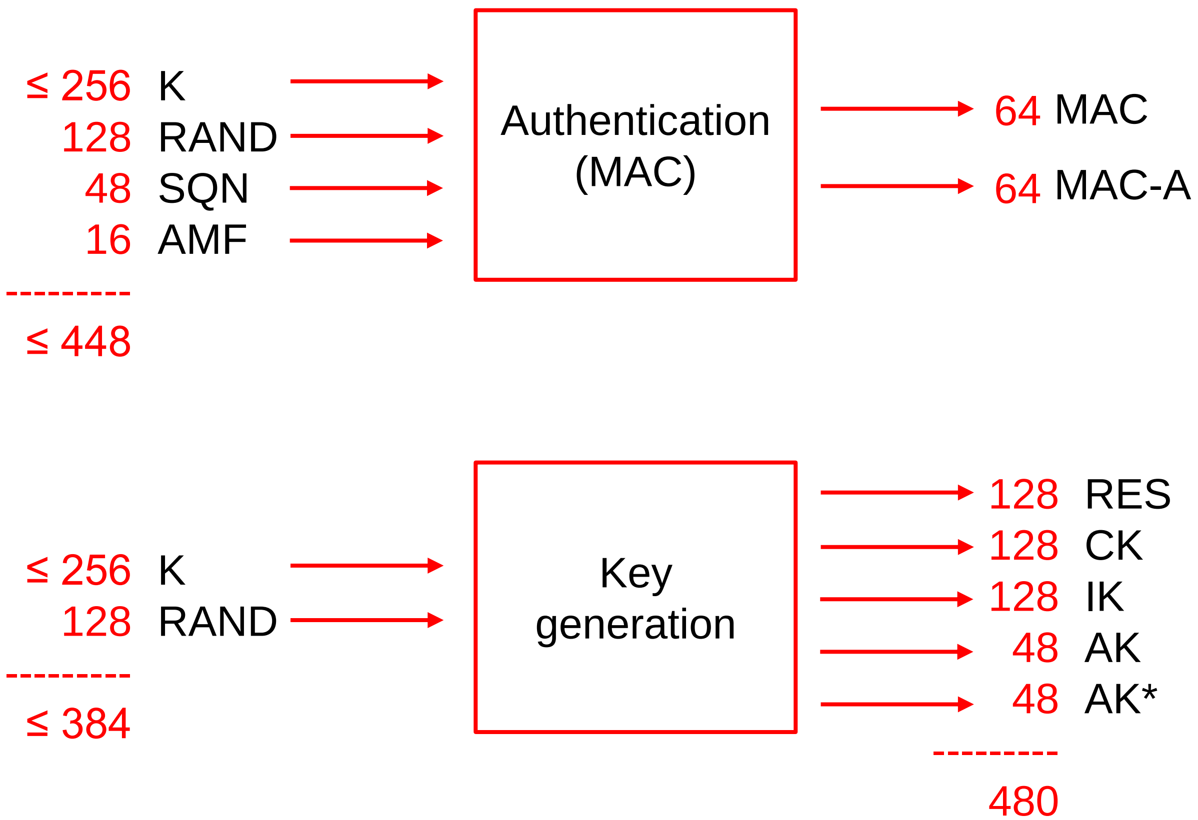
Grøstl - Too close to AES

JH

Keccak

Skein

Input and output sizes



Keyed MAC constructions

SHA-2

> With SHA-512, inputs fit into one block and outputs fit into one hash

> HMAC is the traditional, trusted choice

– Some precomputations possible, but still need two hash calls

> Simpler construction for fixed length input?

– Secret prefix, Hash (key || message)? No length extension attacks

KECCAK

> Documentation recommends Hash (key || IV-if-required || message) for MACs

Keccak considerations

> *On the security of the keyed sponge construction*, Bertoni / Daemen / Peeters / van Assche (SKEW 2011)

Theorem 1. *The advantage of distinguishing* KEYEDSPONGE$[f, K]$ *from a random oracle, with* f *a random permutation and* K *uniformly distributed, is upper bounded by:*

$$1 - \exp\left(-\frac{\frac{M^2}{2} + 2(M+1)(N+1)}{2^c}\right) + P_{key}(N) \ ,$$

where M *is the data complexity,* N *the time complexity and* $P_{key}(N)$ *the probability of guessing the key after* N *queries.*

> Parameter choices

- State size 1600: Capacity 512, Rate 1088
- State size 1600: Capacity 1024, Rate 576
- State size 800: Capacity 512, Rate 288

> Wait for the SHA-3 standard?

SHA-2 vs Keccak

SHA-2

>Better trusted?

>More implementations available

– For SHA-256, anyway

Keccak

>More efficient?

– 16-bit platforms

– Large chaining value, though

>Simpler MAC construction?

>Easier to protect against side channel attacks?

Thank you