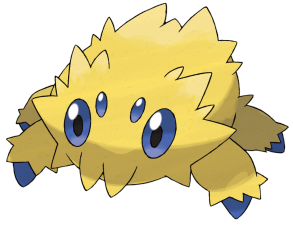
Joltik and Deoxys

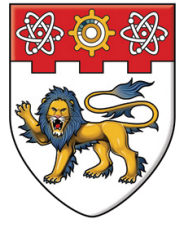
Jérémy Jean Ivica Nikolić Thomas Peyrin

Nanyang Technological University, Singapore

DIAC 2014 – August 23, 2014

http://www1.spms.ntu.edu.sg/~syllab/Joltik
http://www1.spms.ntu.edu.sg/~syllab/Deoxys





NANYANG TECHNOLOGICAL UNIVERSITY

Introduction

- Presentation of Joltik and Deoxys candidates.
- Together with Kiasu, they are different instances of the new TWEAKEY framework that we propose.
- Joltik and Deoxys share the same structure inside this framework.
- They use tweakable block ciphers (as Kiasu).
- Joltik: lightweight and hardware-oriented.
- Deoxys: fast and software-oriented (AES-NI).

Tweakable block ciphers for AEAD

Previous work on TBC:

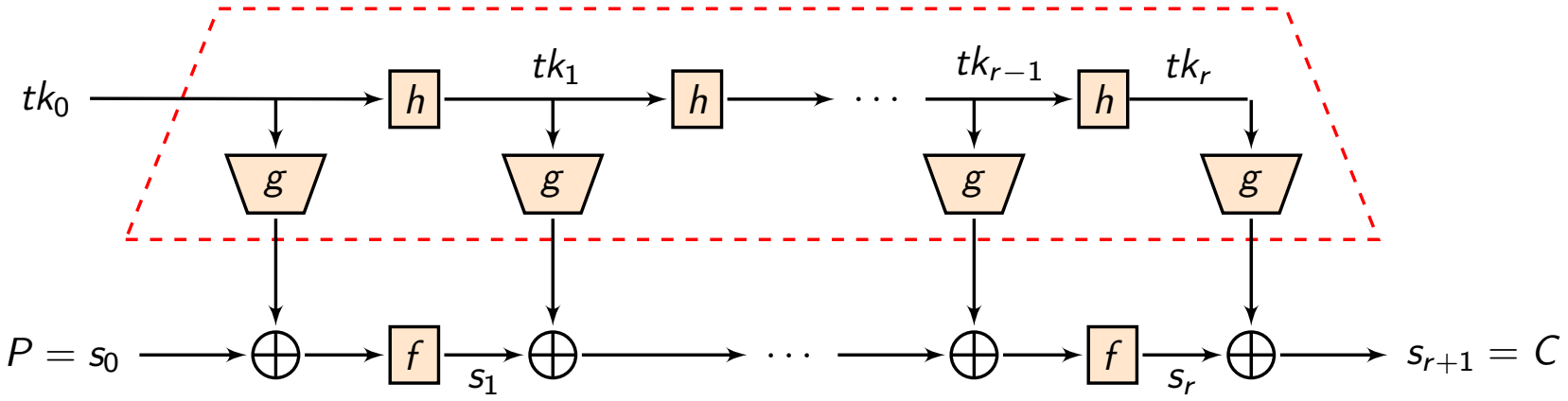
- Several known methods for TBC, e.g.: LRW, XEX.
- Drawback: birthday-bound security.

(new) The TWEAKEY **framework:** ***to appear at ASIACRYPT 2014***

- Unified approach to handle keys and tweaks.
- Standalone primitive to achieve a TBC.
- Tweak and key processed (almost) the same way.
- Only a framework $\Longrightarrow$ unsecured instances exist.
- Security reduction: regular block cipher with new key schedule.
- Particular subclass: Superposition-TWEAKEY (STK).
  $\Longrightarrow$ Precise the tweakey schedule.

The TWEAKEY framework

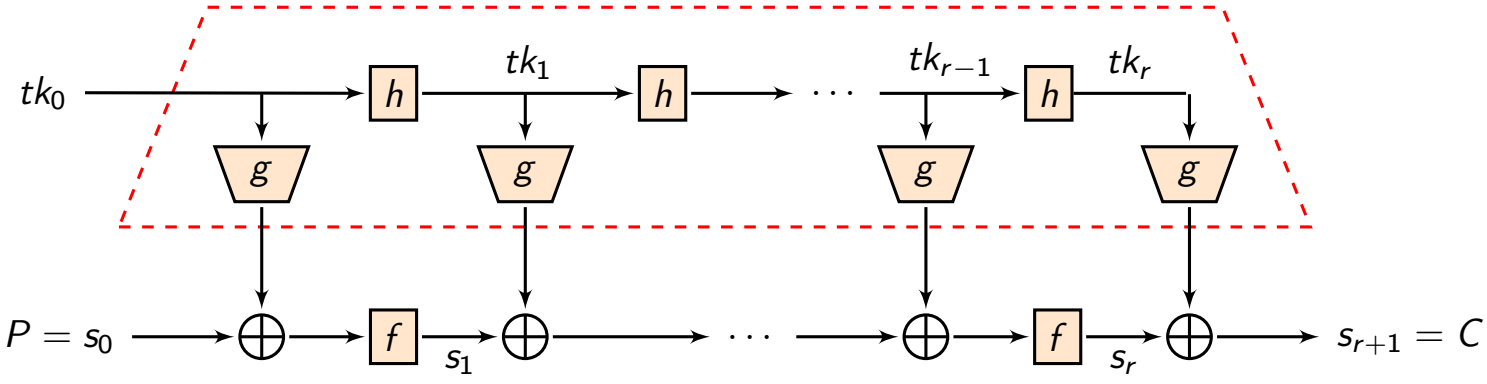
TWEAKEY generalizes the class of key-alternating (KA) cipher.



TWEAKEY

- The regular key schedule is replaced by a TWEAKEY schedule.
- An n-bit key n-bit tweak TBC have $2n$-bit tweakey and g compresses $2n$ to n bits.
- Such a primitive would be a TK-2 primitive (TWEAKEY of order 2).
- The same primitive can be seen as a $2n$-bit key cipher with no tweak (or $1.5n$-bit key $0.5n$-bit tweak, etc).

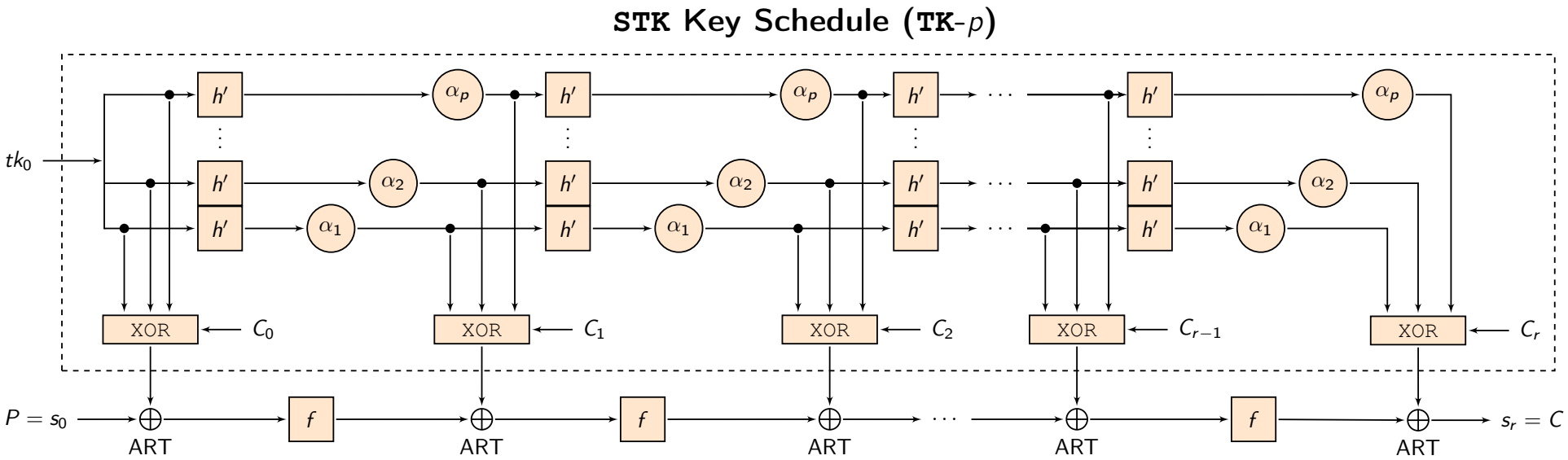
Towards the STK construction (Superposition-TWEAKEY)



Simplifications

- We would like to process the key and tweak inputs independently in the TWEAKEY schedule h and in the same way.
- The subtweakey addition of $g(tk_i)$ consists in XORing all the n-bit words of the tweakey state into the internal state.
- This would:
  - reduce the implementation overhead,
  - reduce the area footprint by reusing code,
  - simplify the security analysis.
- **But**: possible interactions between the XOR of n-bit tweakey words.

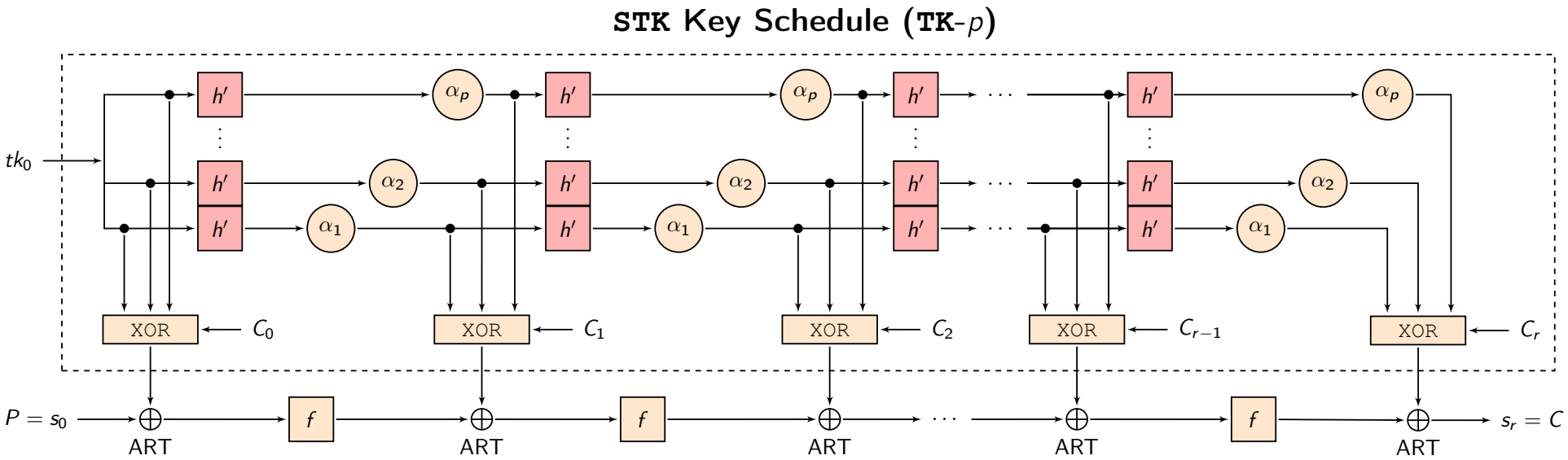
The STK construction



STK

- We consider c-bit nibbles in each (say p) n-bit tweakey words.
- The h function is replaced by n independent applications of a h' function, which is a nibble-wise substitution.
- To reduce the interaction of the tweakey words at the output of the g function, each nibble of the k-th tweakey word is multiplied by a value $\alpha_k \in GF(2^c)$.

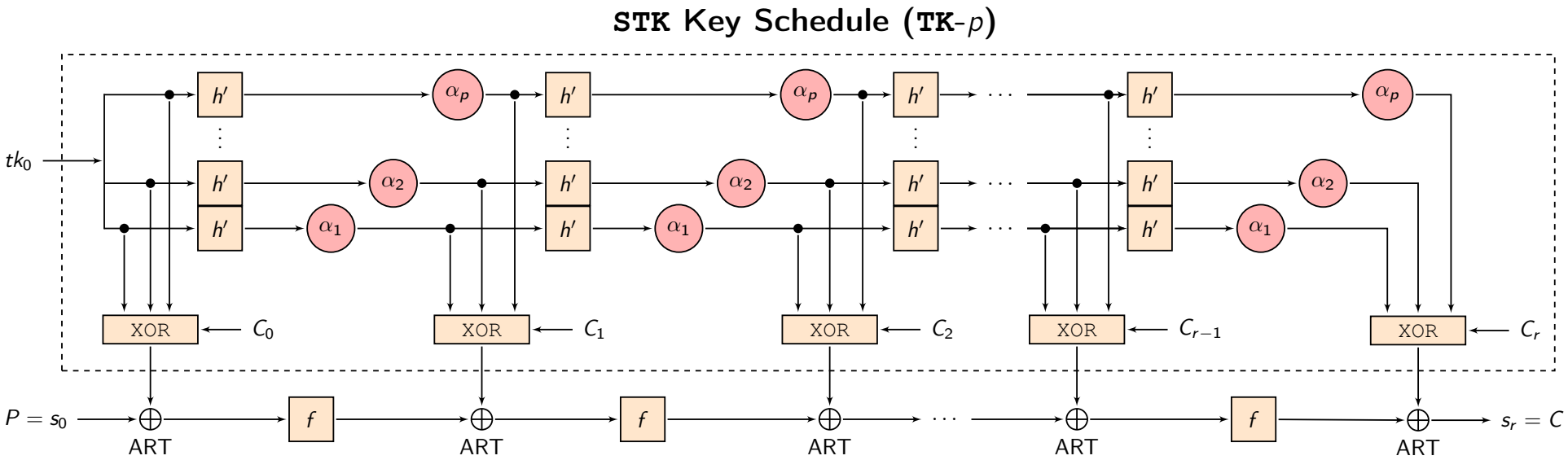
The STK construction



STK

- We consider *c*-bit nibbles in each (say *p*) *n*-bit tweakey words.
- The *h* function is replaced by *n* independent applications of a *h*′ function, which is a nibble-wise substitution.
- To reduce the interaction of the tweakey words at the output of the *g* function, each nibble of the *k*-th tweakey word is multiplied by a value $\alpha_k \in GF(2^c)$.

The STK construction



STK

- We consider *c*-bit nibbles in each (say *p*) *n*-bit tweakey words.
- The *h* function is replaced by *n* independent applications of a h' function, which is a nibble-wise substitution.
- To reduce the interaction of the tweakey words at the output of the *g* function, each nibble of the *k*-th tweakey word is multiplied by a value $\alpha_k \in GF(2^c)$.

The STK construction: rationale

Design choices:

- Multiplication in $GF(2^c)$ controls the number of cancellations at the output of g, when the subtweakeys are XORed to the internal state.
- Rely on a linear code to bound the number of cancellations.

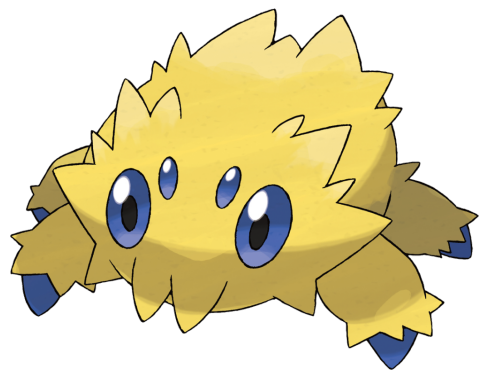
Security analysis:

- Simplified security analysis in STK.
- Easy analysis of the tweakey schedule (hard for AES).
- Possibility to reuse previous works and several existing tools searching for high-probability differential characteristics (easy to introduce limitations of the number of cancellations of differences).

Implementation:

- Very simple transformations: linear and lightweight.
- Multiplications constants chosen as 1, 2, 4, ... for efficiency.

Joltik



Lightweight and hardware-oriented candidate to CAESAR.

Joltik

- Two family of ciphers: Joltik$^{\neq}$ and Joltik$^{=}$.
- Joltik$^{\neq}$ assumes nonce-respecting users:
  - Rely on the ΘCB3 framework.
  - Full security.
  - Four recommended parameters (see submission).
- Joltik$^{=}$ allows nonce-repeating users.
  - Rely on the COPA mode.
  - Birthday-bound security.
  - Four recommended parameters (see submission).
- Exactly the same modes as Kiasu (see previous presentation).
- Rely on the Joltik-BC tweakable block cipher.

Joltik-BC

- Instance of the STK construction.
- Two members: Joltik-BC-128 and Joltik-BC-192.
  - 128 bits for TK-2: $|key| + |tweak| = 128$ (2 tweakey words).
  - 192 bits for TK-3: $|key| + |tweak| = 192$ (3 tweakey words).
- AES-based design.
- Involutive MDS matrix in MixColumns $\Longrightarrow$ low decryption overhead.
- S-Box from the Piccolo block cipher (compact in hardware).
- Joltik-BC-128 has 24 rounds (TK-2).
- Joltik-BC-192 has 32 rounds (TK-3).
- TWEAKEY schedule:
  - h' is a simple permutation of the 16 nibbles.
  - Multiplications factor are: 1, 2 and 4 in $GF(16)/0x13$.
  - Constant additions to break symmetries (from LED cipher).

Security claims of Joltik (bits of security, $\log_2$)

Nonce-respecting user

	Joltik$^{\neq}$	Joltik$^{=}$
Confidentiality for the plaintext	k	$n/2$
Integrity for the plaintext	n	$n/2$
Integrity for the associated data	n	$n/2$

Nonce-repeating user

	Joltik$^{\neq}$	Joltik$^{=}$
Confidentiality for the plaintext	none	$n/2$
Integrity for the plaintext	none	$n/2$
Integrity for the associated data	none	$n/2$

Conjectured security of Joltik (bits of security, $\log_2$)

Nonce-respecting user

	Joltik$^{\neq}$	Joltik$^{=}$
Confidentiality for the plaintext	k	n
Integrity for the plaintext	n	n
Integrity for the associated data	n	n

Nonce-repeating user

	Joltik$^{\neq}$	Joltik$^{=}$
Confidentiality for the plaintext	none	$n/2$
Integrity for the plaintext	none	$n/2$
Integrity for the associated data	none	$n/2$

Implementations of Joltik$^{\neq}$

Software implementations

- vperm implementation (SSSE3 and avx2): about the same (expected) speed as LED.
- Projection for bitslice: about 9 cpb for 4KB messages.
- Similar numbers for other Joltik$^{\neq}$ parameters.
- Joltik$^{=}$ expected to be 2x slower.

Hardware implementations

- Estimations (see specs): (LED-128: about 1300GE)
  - 1500 GE for Joltik-BC-128 (TBC only),
  - 2000 GE for Joltik-BC-128 (TBC only),
  - 2100 GE for Joltik TK-2,
  - 2600 GE for Joltik TK-3.
- See estimations for Joltik$^{=}$ in the specs.

Deoxys

Fast and software-oriented candidate to CAESAR.

Deoxys

- Also two family of ciphers:
  - Deoxys$^{\neq}$ for nonce-respecting users,
  - Deoxys$^{=}$ for nonce-repeating users.
- Same modes as Joltik and Kiasu.
- Two sets of recommended parameters for each mode.
- Rely on the Deoxys-BC tweakable block cipher.

Deoxys-BC

- Also an instance of the STK construction.
- Two members: Deoxys-BC-256 and Deoxys-BC-384.
  - 256 bits for TK-2: $|key| + |tweak| = 256$ (2 tweakey words).
  - 384 bits for TK-3: $|key| + |tweak| = 384$ (3 tweakey words).
- The round function is **exactly** the AES round function (AES-NI).
- Deoxys-BC-256 has 14 rounds (TK-2).
- Deoxys-BC-384 has 16 rounds (TK-3).
- TWEAKEY schedule:
  - h' is the same permutation as Joltik.
  - Multiplications factor are: 1, 2 and 4 in the AES field.
  - Constant additions to break symmetries (RCON from AES KS).

Security claims of Deoxys (bits of security, $\log_2$)

Same as Joltik.

Nonce-respecting user

	Deoxys$^{\neq}$	Deoxys$^{=}$
Confidentiality for the plaintext	k	$n/2$
Integrity for the plaintext	n	$n/2$
Integrity for the associated data	n	$n/2$

Nonce-repeating user

	Deoxys$^{\neq}$	Deoxys$^{=}$
Confidentiality for the plaintext	none	$n/2$
Integrity for the plaintext	none	$n/2$
Integrity for the associated data	none	$n/2$

Conjectured security of Deoxys (bits of security, $\log_2$)

Same as Joltik.

Nonce-respecting user

	Deoxys$^{\neq}$	Deoxys$^{=}$
Confidentiality for the plaintext	k	n
Integrity for the plaintext	n	n
Integrity for the associated data	n	n

Nonce-repeating user

	Deoxys$^{\neq}$	Deoxys$^{=}$
Confidentiality for the plaintext	none	$n/2$
Integrity for the plaintext	none	$n/2$
Integrity for the associated data	none	$n/2$

Performances of Deoxys using AES-NI.

Benchmark of Deoxys$^{\neq}$ with 128-bit key 128-bit tweak (in cpb).

	1KB	2KB	4KB	8KB	64KB
Intel Haswell	2.12	1.74	1.55	1.46	**1.38**
Intel Sandy Bridge	2.37	1.85	1.59	1.43	**1.31**

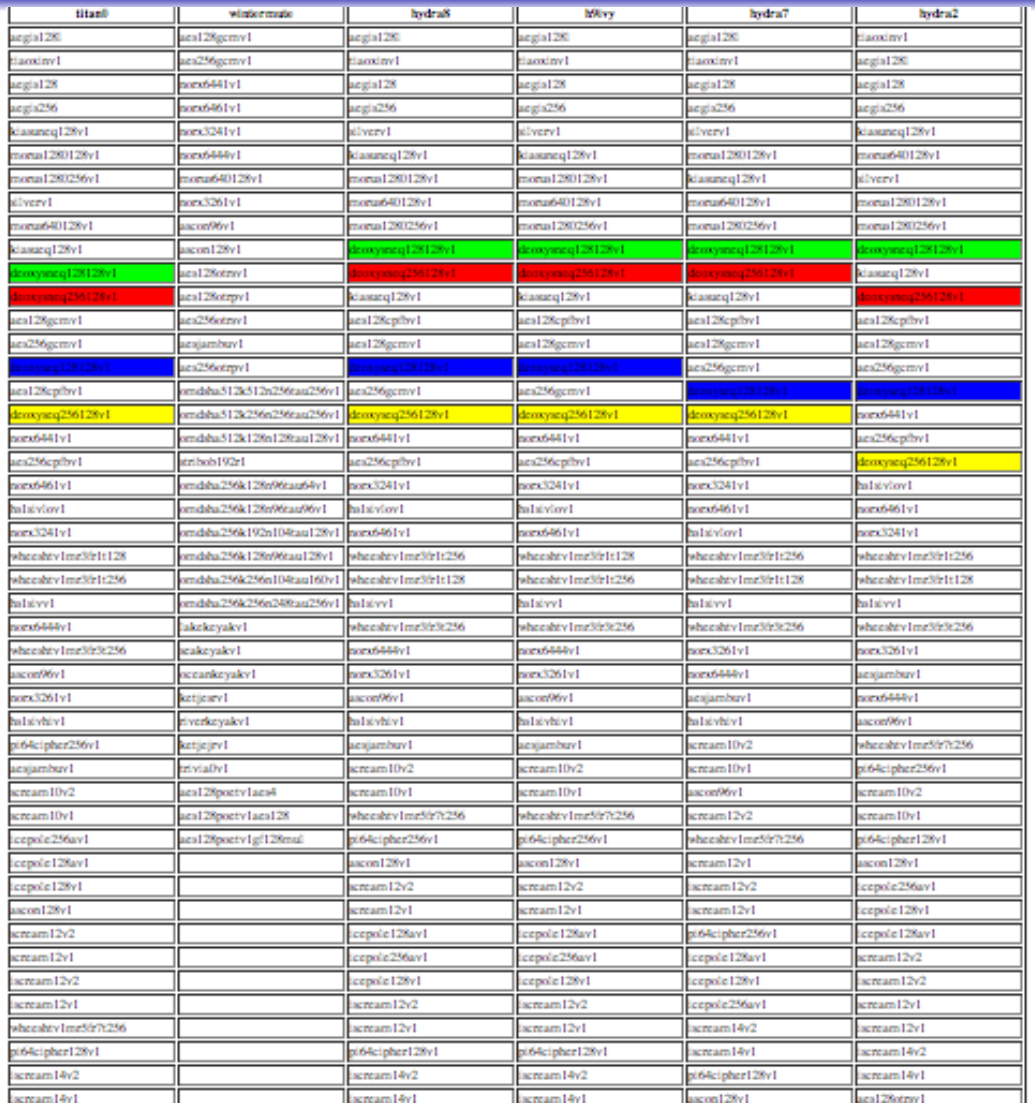
Benchmark of Deoxys$^{=}$ with 128-bit key 128-bit tweak (in cpb).

	1KB	2KB	4KB	8KB	64KB
Intel Haswell	3.75	3.13	2.84	2.69	**2.56**
Intel Sandy Bridge	4.74	3.91	3.44	3.11	**2.80**

Notes:

- Benchmarks done in the $K_\Delta N_\Delta$ model.
- Fast non AES-NI implementations coming soon.
- Twice more TBC calls in Deoxys$^{=}$ to achieve nonce-misuse resistance.

Performances of Deoxys using AES-NI.



titan8	wintermute	hydra8	h9vy	hydra7	hydra2
aegis128l	aes128gcmv1	aegis128l	aegis128l	aegis128l	tiaoxinv1
tiaoxinv1	aes256gcmv1	tiaoxinv1	tiaoxinv1	tiaoxinv1	aegis128l
aegis128	norx6441v1	aegis128	aegis128	aegis128	aegis128
aegis256	norx6461v1	aegis256	aegis256	aegis256	aegis256
tiaoxneq128v1	norx3241v1	tl1vervl	tl1vervl	tl1vervl	tiaoxneq128v1
morus1280128v1	norx6444v1	tiaoxneq128v1	tiaoxneq128v1	morus1280128v1	morus640128v1
morus1280256v1	morus640128v1	morus1280128v1	morus1280128v1	tiaoxneq128v1	tl1vervl
tl1vervl	norx3261v1	morus640128v1	morus640128v1	morus640128v1	morus1280128v1
morus640128v1	ascon96v1	morus1280256v1	morus1280256v1	morus1280256v1	morus1280256v1
tiaoxneq128v1	ascon128v1	deoxysneq128128v1	deoxysneq128128v1	deoxysneq128128v1	deoxysneq128128v1
deoxysneq128128v1	aes128otrnv1	deoxysneq256128v1	deoxysneq256128v1	deoxysneq256128v1	tiaoxneq128v1
deoxysneq256128v1	aes128otrpv1	tiaoxneq128v1	tiaoxneq128v1	tiaoxneq128v1	deoxysneq256128v1
aes128gcmv1	aes256otrnv1	aes128cpfbv1	aes128cpfbv1	aes128cpfbv1	aes128cpfbv1
aes256gcmv1	aesjambuv1	aes128gcmv1	aes128gcmv1	aes128gcmv1	aes128gcmv1
deoxysneq128128v1	aes256otrpv1	deoxysneq128128v1	deoxysneq128128v1	aes256gcmv1	aes256gcmv1
aes128cpfbv1	sendblu512k512n256tau256v1	aes256gcmv1	aes256gcmv1	deoxysneq128128v1	deoxysneq128128v1
deoxysneq256128v1	sendblu512k256n256tau256v1	deoxysneq256128v1	deoxysneq256128v1	deoxysneq256128v1	norx6441v1
norx6441v1	sendblu512k128n128tau128v1	norx6441v1	norx6441v1	norx6441v1	aes256cpfbv1
aes256cpfbv1	triboh192r1	aes256cpfbv1	aes256cpfbv1	aes256cpfbv1	deoxysneq256128v1
norx6461v1	sendblu256k128n96tau64v1	norx3241v1	norx3241v1	norx3241v1	tu1ivlovl
tu1ivlovl	sendblu256k128n96tau96v1	tu1ivlovl	tu1ivlovl	norx6461v1	norx6461v1
norx3241v1	sendblu256k192n104tau128v1	norx6461v1	norx6461v1	tu1ivlovl	norx3241v1
wheeshtv1me3l9r1t128	sendblu256k128n96tau128v1	wheeshtv1me3l9r1t256	wheeshtv1me3l9r1t128	wheeshtv1me3l9r1t256	wheeshtv1me3l9r1t256
wheeshtv1me3l9r1t256	sendblu256k256n104tau160v1	wheeshtv1me3l9r1t128	wheeshtv1me3l9r1t256	wheeshtv1me3l9r1t128	wheeshtv1me3l9r1t128
tu1ivvl	sendblu256k256n248tau256v1	tu1ivvl	tu1ivvl	tu1ivvl	tu1ivvl
norx6444v1	takekzyakv1	wheeshtv1me3l9r3r256	wheeshtv1me3l9r3r256	wheeshtv1me3l9r3r256	wheeshtv1me3l9r3r256
wheeshtv1me3l9r3r256	keakzyakv1	norx6444v1	norx6444v1	norx3261v1	norx3261v1
ascon96v1	screamkzyakv1	norx3261v1	norx3261v1	norx6444v1	aesjambuv1
norx3261v1	ketjejarv1	ascon96v1	ascon96v1	aesjambuv1	norx6444v1
tu1ivhtvl	tvierkzyakv1	tu1ivhtvl	tu1ivhtvl	tu1ivhtvl	ascon96v1
pi64cipher256v1	ketjejrv1	aesjambuv1	aesjambuv1	scream10v2	wheeshtv1me5l9r7r256
aesjambuv1	trivia0v1	scream10v2	scream10v2	scream10v1	pi64cipher256v1
scream10v2	aes128pootv1acs8	scream10v1	scream10v1	ascon96v1	scream10v2
scream10v1	aes128pootv1acs128	wheeshtv1me5l9r7r256	wheeshtv1me5l9r7r256	scream12v2	scream10v1
icepole256av1	aes128pootv1gf128mul	pi64cipher256v1	pi64cipher256v1	wheeshtv1me5l9r7r256	pi64cipher128v1
icepole128av1		ascon128v1	ascon128v1	scream12v1	ascon128v1
icepole128v1		scream12v2	scream12v2	scream12v2	icepole256av1
ascon128v1		scream12v1	scream12v1	scream12v1	icepole128v1
scream12v2		icepole128av1	icepole128av1	pi64cipher256v1	icepole128av1
scream12v1		icepole256av1	icepole256av1	icepole128av1	scream12v2
iscream12v2		icepole128v1	icepole128v1	icepole128v1	iscream12v2
iscream12v1		iscream12v2	iscream12v2	icepole256av1	scream12v1
wheeshtv1me5l9r7r256		iscream12v1	iscream12v1	iscream14v2	iscream12v1
pi64cipher128v1		pi64cipher128v1	pi64cipher128v1	iscream14v1	iscream14v2
iscream14v2		iscream14v2	iscream14v2	pi64cipher128v1	iscream14v1
iscream14v1		iscream14v1	iscream14v1	ascon128v1	aes128otrnv1

Deoxys in the top 10% of AES-NI implementations on SUPERCOP.

Source: http://www1.spms.ntu.edu.sg/~syllab/speed/.

Security analysis

- We have scrutinized the security of the TWEAKEY framework, and devised the STK subclass.
  $\Longrightarrow$ Provide bounds on the number of differences introduces by the tweakey schedule.
- This bound can easily be used in existing differential characteristic search tools.
- We conducted a differential analysis, and selected the number of rounds such that:
  - Joltik-BC has 8 rounds of security margin,
  - Deoxys-BC has 4 rounds of security margin.
- Also in the submission documents: analysis against MITM strategy.

Conclusion

- We propose the TWEAKEY framework to design easy-to-analyze tweakable block ciphers (more in an upcoming ASIACRYPT 2014 paper).
- We instantiate this framework to get two TBC:
  - Joltik-BC, which is lightweight and hardware-oriented,
  - Deoxys-BC, which is fast and software-oriented.
- We plug these two ciphers into two different modes to achieve AEAD schemes:
  - one mode similar to OCB3 for nonce-respecting users,
  - one mode similar to COPA to achieve nonce-misuse resistance.

Conclusion

- We propose the TWEAKEY framework to design easy-to-analyze tweakable block ciphers (more in an upcoming ASIACRYPT 2014 paper).
- We instantiate this framework to get two TBC:
  - Joltik-BC, which is lightweight and hardware-oriented,
  - Deoxys-BC, which is fast and software-oriented.
- We plug these two ciphers into two different modes to achieve AEAD schemes:
  - one mode similar to OCB3 for nonce-respecting users,
  - one mode similar to COPA to achieve nonce-misuse resistance.

Thank you!