KDM Security in the Hybrid Framework

Gareth T. Davies and Martijn Stam
gareth.davies@bristol.ac.uk

University of Bristol, UK

CT-RSA '14

28 February 2014

KDM Security in the Hybrid Framework

Results

Context

Investigated KDM Security in the context of Hybrid Encryption:

- Present a generic composition theorem for adaptive, KDM-secure hybrid encryption in the Random Oracle model.
- Proof method incorporates non-standard techniques that could be applicable in a wider context.

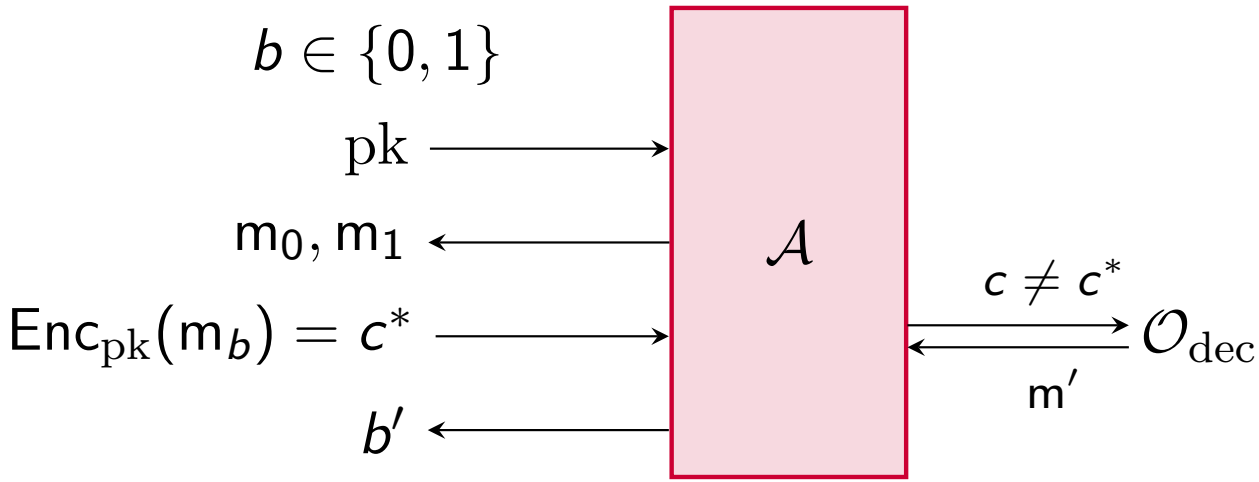
Motivation - KDM Security

Key-Dependent Message (KDM) Security involves an environment where the **adversary can receive encryptions of arbitrary functions of the secret key**, and it is a concern in many scenarios:

- Disk encryption systems (e.g. Bitlocker)
- Anonymous Credential Systems
- Formal Verification (Dolev-Yao proofs)

CCA Security

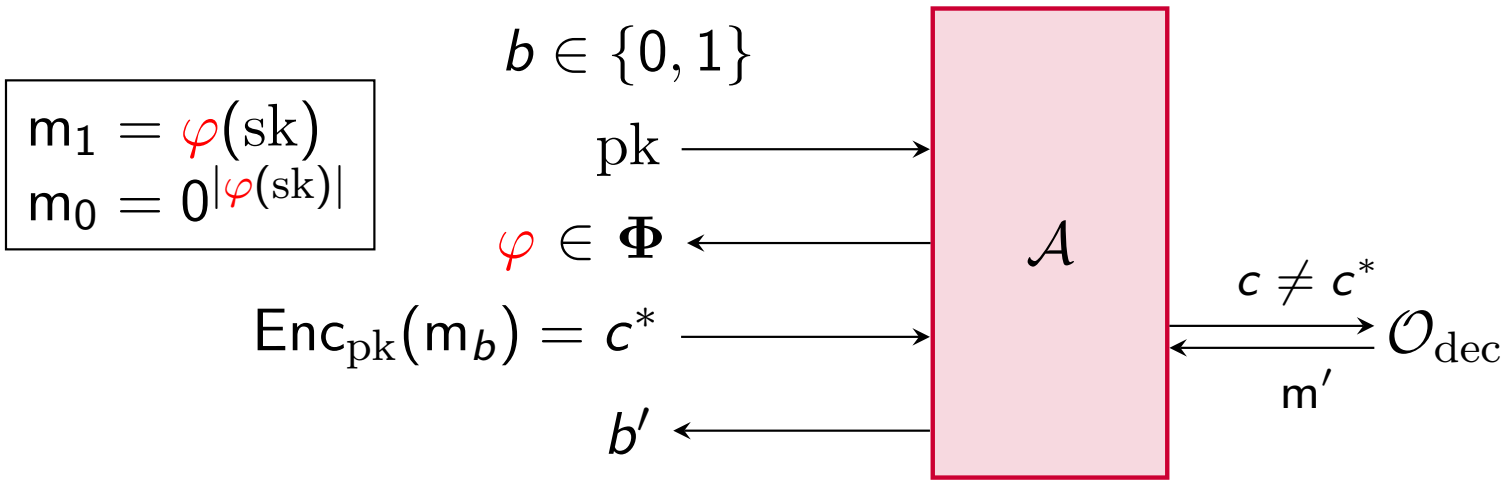
To define KDM security, first recall the definition of **IND-CCA security** (asymmetric encryption):



$\mathcal{A}$ wins if $b' = b$, and the scheme is **IND-CCA-secure** if $\mathcal{A}$'s advantage is no better than guessing.

KDM Security

Now to define **KDM security** (asymmetric setting):



Scheme is **IND-KDM-CCA[Φ] Secure** if $\mathcal{A}$'s advantage is no better than guessing.

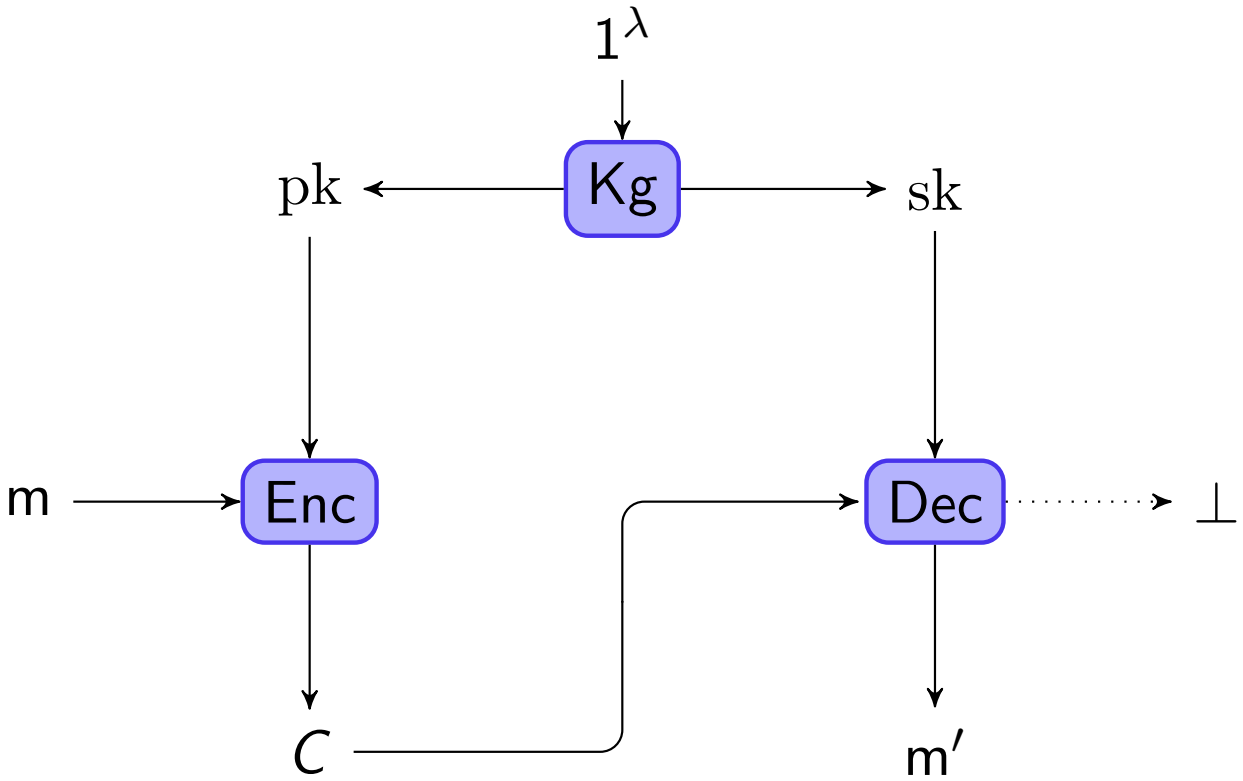
Goals for Cryptographers: KDM Security

- Investigate relations between KDM security and standard notions of security.
- Construct KDM-secure schemes for large function classes $\mathbf{\Phi}$.
- Prove existing schemes KDM-secure for reasonable $\mathbf{\Phi}$.

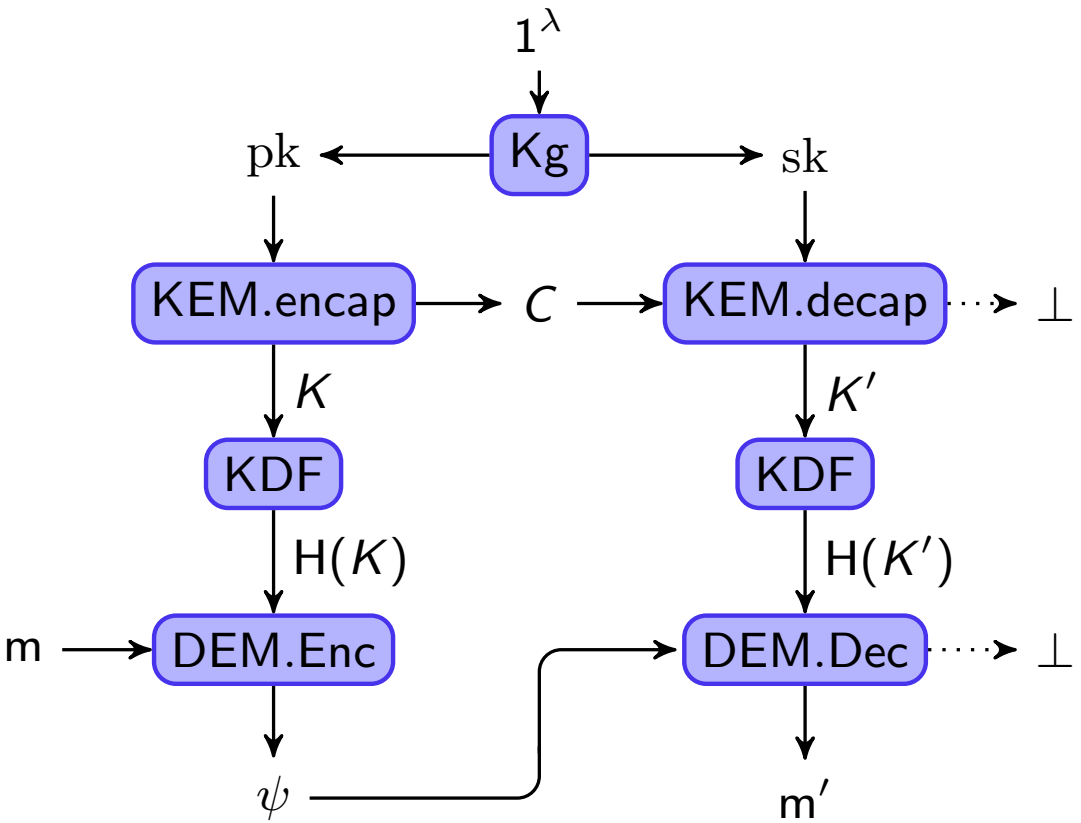
Prior Work

- Camenisch and Lysyanskaya EC'01 (anonymous credential systems) & Black, Rogaway and Shrimpton SAC'02 (definitions in ROM).
- Boneh et al. Crypto'08 presented the first scheme secure under chosen plaintext attacks in the standard model.
- Camenisch et al. EC'09 gave a scheme secure under active attacks in the standard model.
- Numerous schemes KDM-secure under a variety of number-theoretic assumptions.
- Negative results suggesting difficulty of acquiring generic statements.

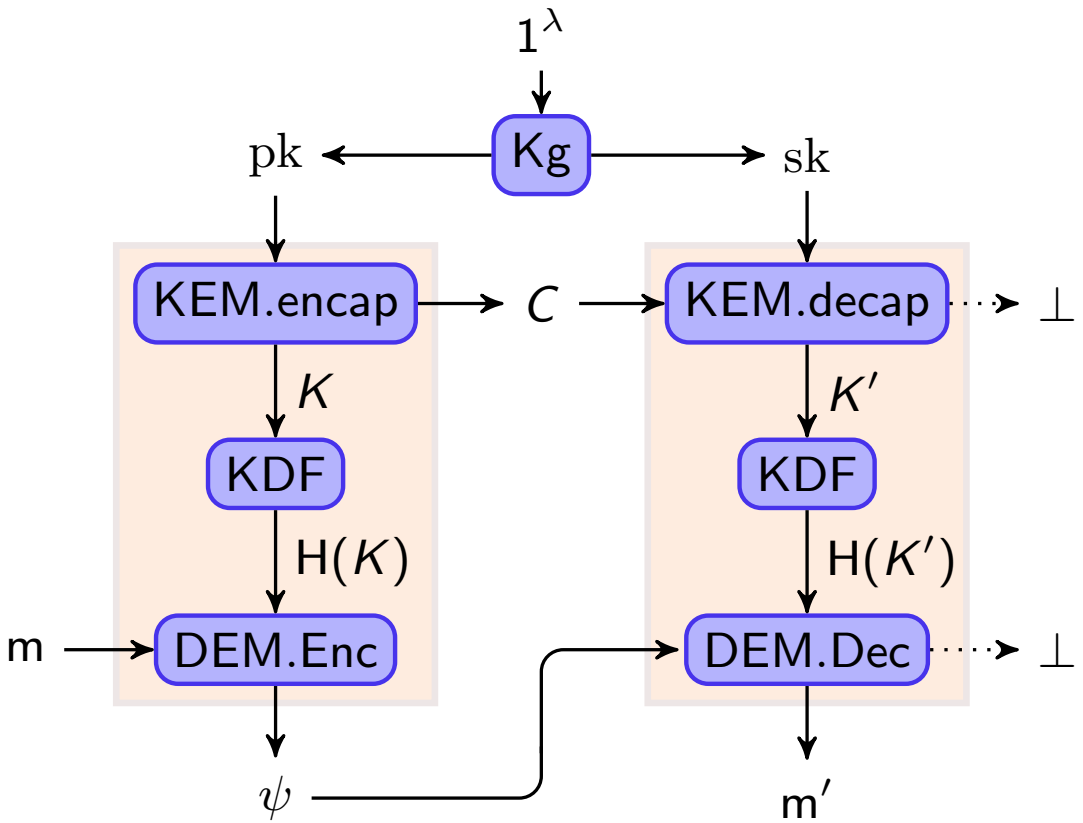
Public Key Encryption



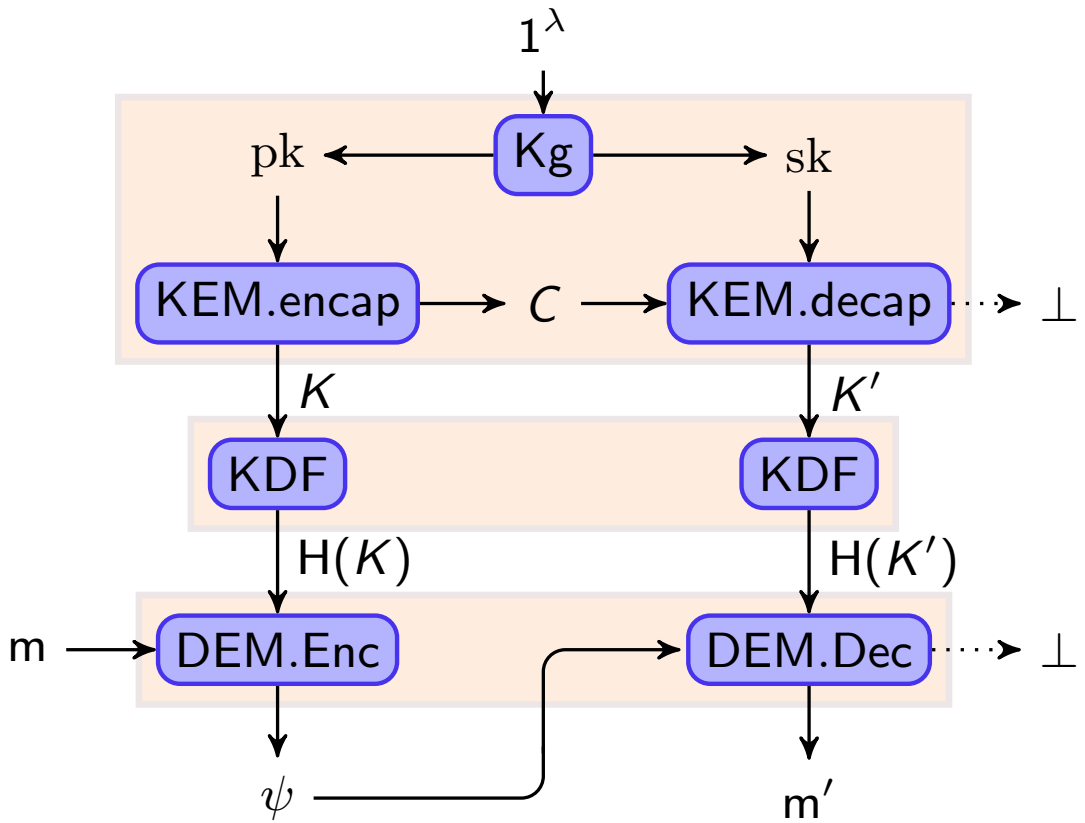
Hybrid Encryption: KEM-DEM Framework



Hybrid Encryption: KEM-DEM Framework



Hybrid Encryption: KEM-DEM Framework



KEM: IND-CCA2

+

KDF: balanced

+

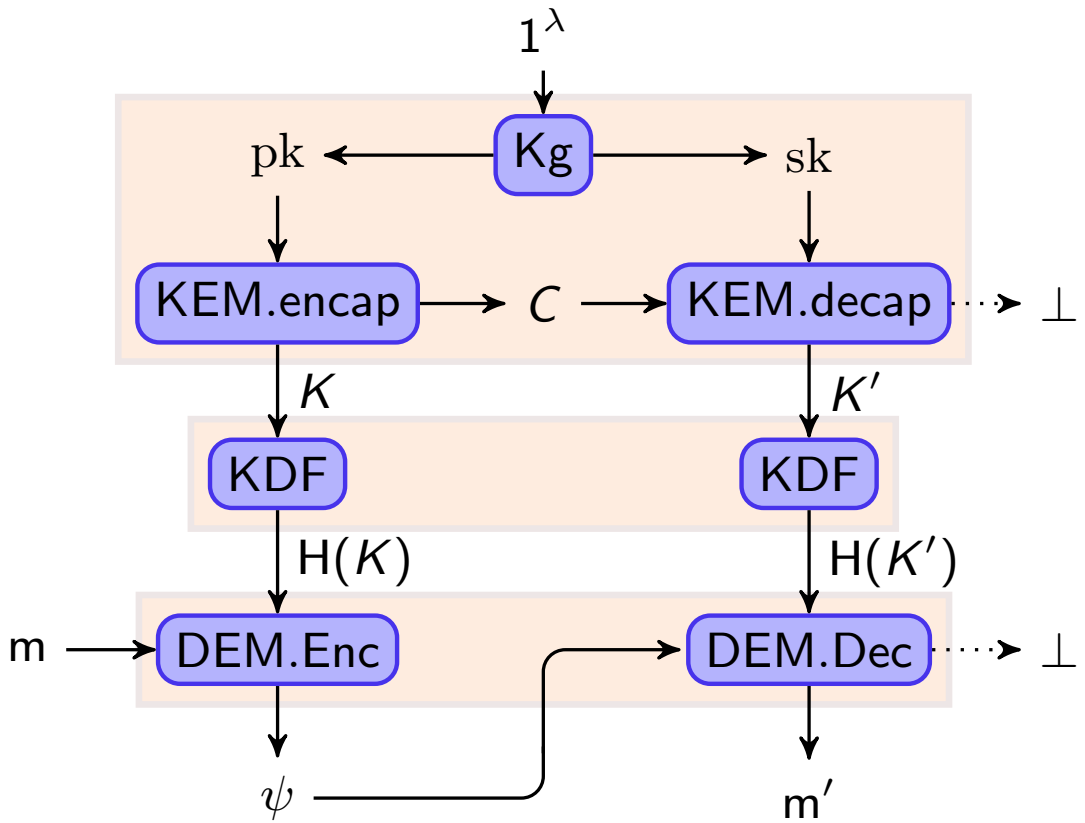
DEM: IND-CCA2

⇓

PKE: IND-CCA2

Shoup EC '00; Cramer and Shoup EC '02

Hybrid Encryption: KEM-DEM Framework



KEM: 2-Universal HPS

+

KDF: balanced

+

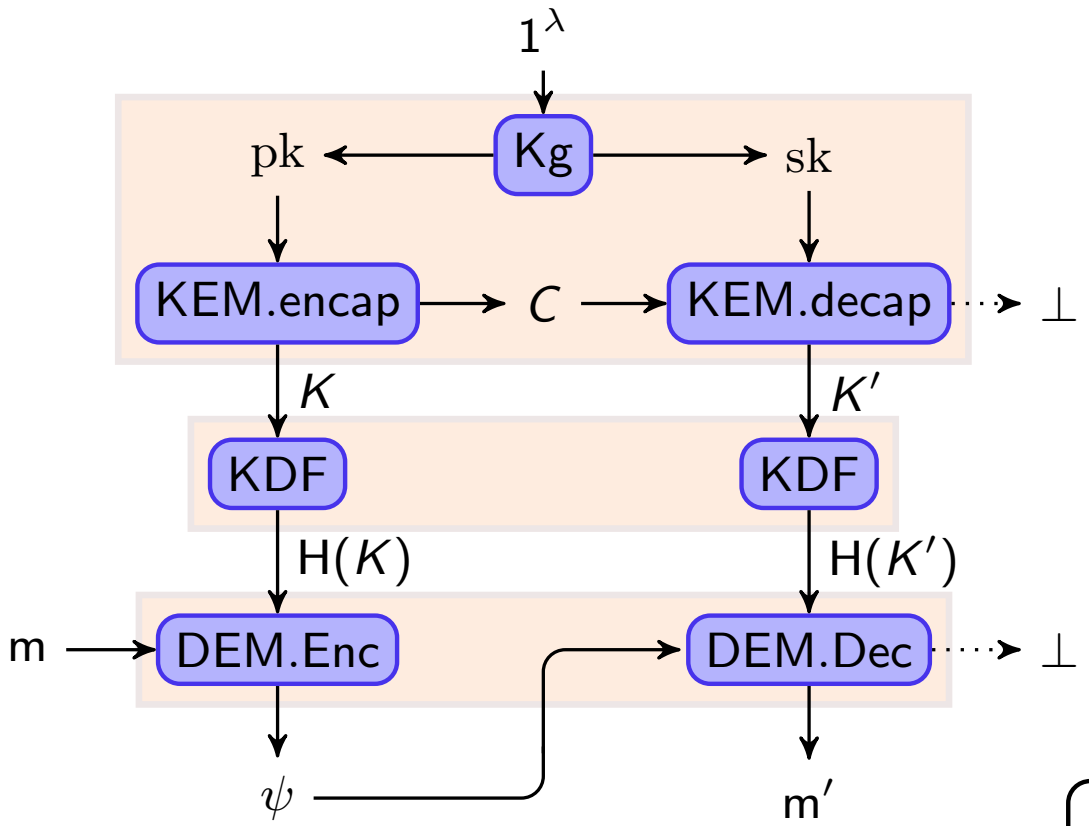
DEM: AE-OT

⇓

PKE: IND-CCA2

Kurosawa and Desmedt Crypto '04; Gennaro and Shoup 2004; Hofheinz and Kiltz Crypto '07;

Hybrid Encryption: KEM-DEM Framework



KEM: ??

+

KDF: Random Oracle

+

DEM: ??

⇓

PKE: **IND-KDM-CCA[Φ]**

Posed as an open problem by Black, Rogaway & Shrimpton SAC '02

Our Results

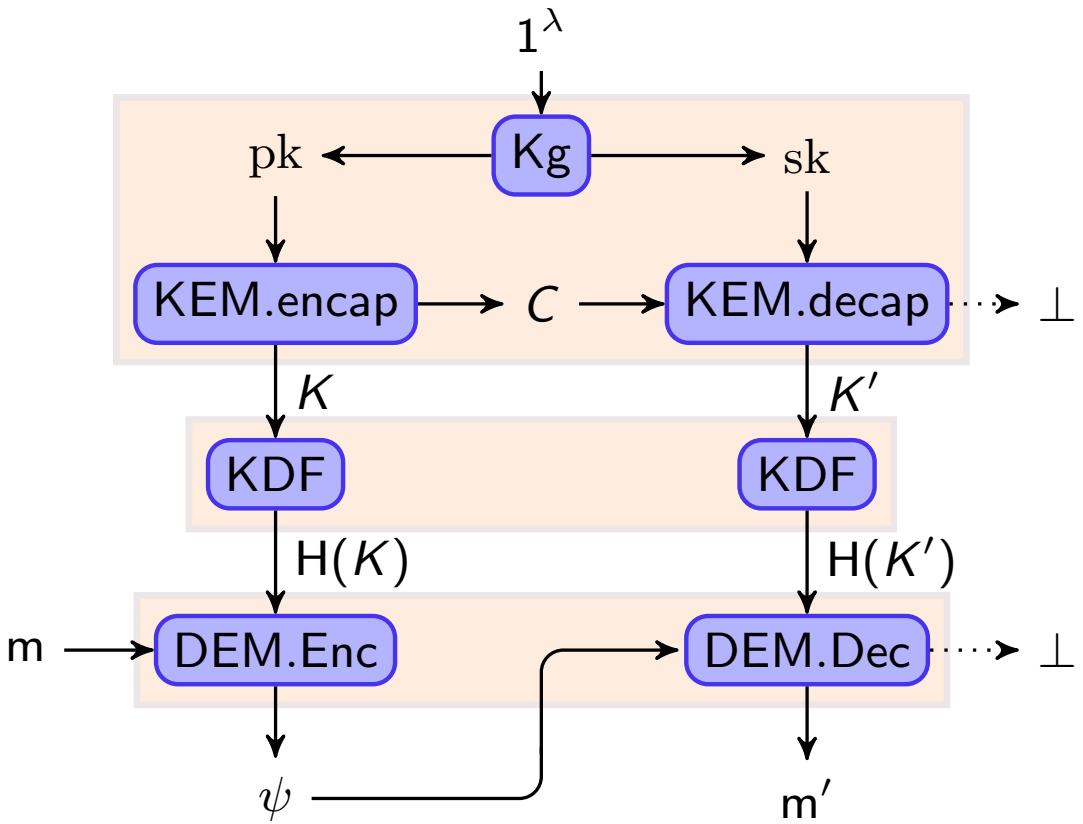
Theorem

If we have a:

- KEM that is μOW-CCA
- KDF that is modelled as a Random Oracle
- DEM that is IND-CCA

then we have a KDM-CCA-**secure Hybrid Encryption** construction for length regular $\varphi \in \mathbf{\Phi}$.

Hybrid KDM



KEM: μ**OW-CCA**

+

KDF: Random Oracle

+

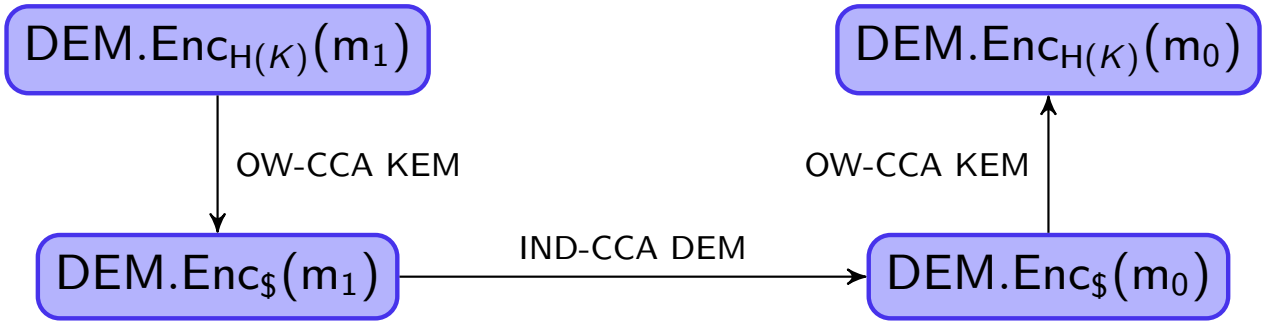
DEM: **IND-CCA**

$\Downarrow$

PKE: **IND-KDM-CCA**[Φ]

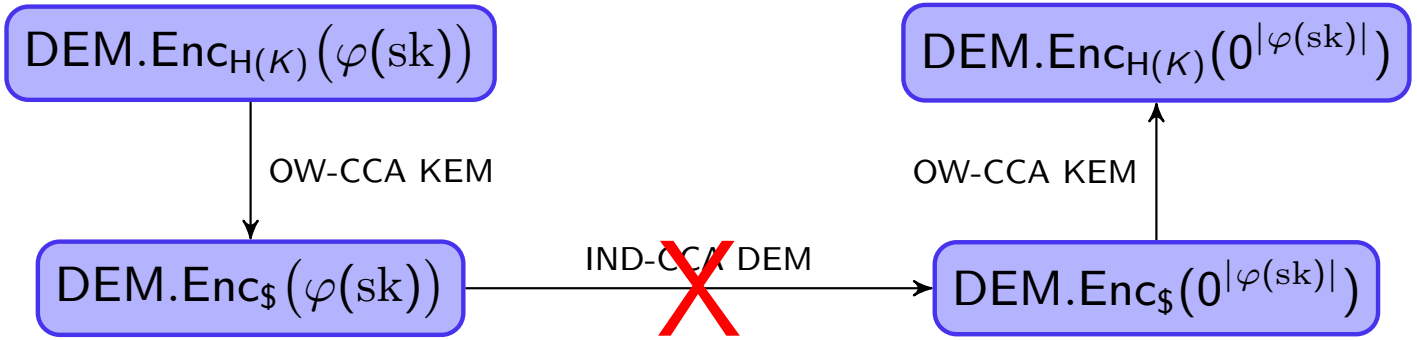
What's different in the KDM setting?

Key-independent approach involves two (identical) KEM hops and one DEM hop:



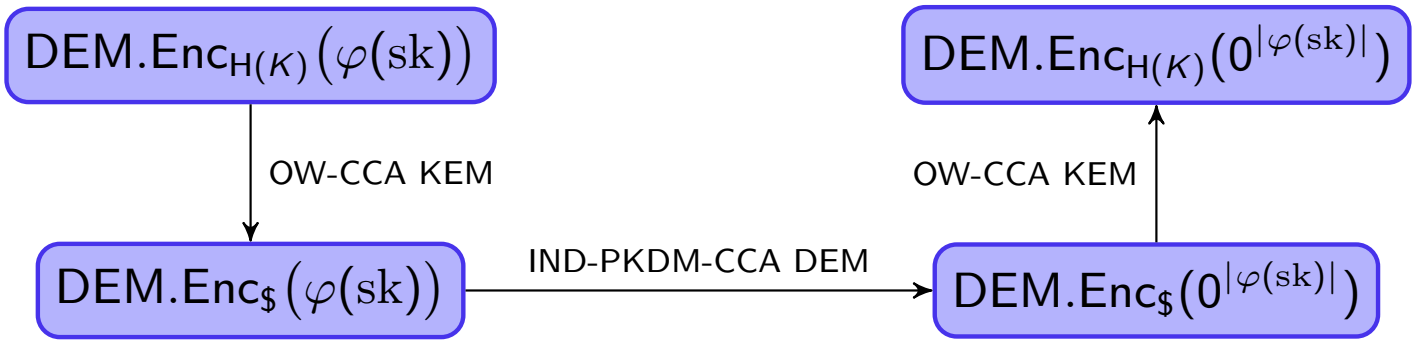
What's different in the KDM setting?

This no longer works, as the encryptions in the DEM hop are not independent.



What's different in the KDM setting?

But it is possible to fix!



Solution: use an equivalent notion, referring to 'prior keys' in system: IND-PKDM-CCA.
KEM hops are not straightforward and require additional tools.

Proof Challenges

- Key-dependency means separation in DEM hop doesn't work.
- No direct reduction to IND-CCA, needed to consider an equivalent notion.
- Needed to include a PRF term in proof to deal simulation issues, though this is not in the construction.

Conclusions

Presented the **first generic composition theorem for adaptively secure key-dependent hybrid encryption**, proven in the random oracle model.

Open problems:

- Tighter reduction to IND-CCA security of the DEM, without using an equivalent notion.
- Standard model composition theorem.

Full version: ePrint 2013/567.

Thanks for your attention!

Questions?

Key Wrapping with the Keccak Permutation

Dmitry Khovratovich

University of Luxembourg

28 February 2014

Key Wrapping

Key wrap problem

Multi-user system (e.g., industrial VPN):

- Many keys in use;
- Need of regular update;
- New session key material (Steve's talk).

Key wrap problem

Multi-user system (e.g., industrial VPN):

- Many keys in use;
- Need of regular update;
- New session key material (Steve's talk).

How to update a key?

$$\text{Encrypt}_{\text{Master key}}(\text{New Key}).$$

Requirements:

- Simple encryption mode;
- Integrity protection;
- Minimum use of extra mechanism (like randomness or nonces).

Encryption Tools

Encryption

Modern encryption:

- Take a block cipher (AES, Present, etc.);
- Plug into a mode of operation (CBC, CTR, etc.);
- Fix a key;
- For each message:
  - Fix IV (random- or nonce-based);
  - Encrypt block by block (pad if necessary).

No integrity protection (yet), only confidentiality — indistinguishability of ciphertexts from random strings.

Authenticated encryption

Authenticated encryption - a single-key construction that achieves both confidentiality and data integrity.

Data integrity/authentication means that a decryptable ciphertext must have been produced with a secret key. Hence most ciphertexts must decrypt to $\perp$.

Authenticated encryption

Authenticated encryption - a single-key construction that achieves both confidentiality and data integrity.

Data integrity/authentication means that a decryptable ciphertext must have been produced with a secret key. Hence most ciphertexts must decrypt to $\perp$.

Several types:

- Modes of operation (OCB, EAX, CCM, GCM);
- Dedicated constructions (Helix/Phelix, Grain128).

They use nonces to achieve confidentiality in the presence of repeated queries or blocks.

Authenticated encryption

Authenticated encryption - a single-key construction that achieves both confidentiality and data integrity.

Data integrity/authentication means that a decryptable ciphertext must have been produced with a secret key. Hence most ciphertexts must decrypt to $\perp$.

Several types:

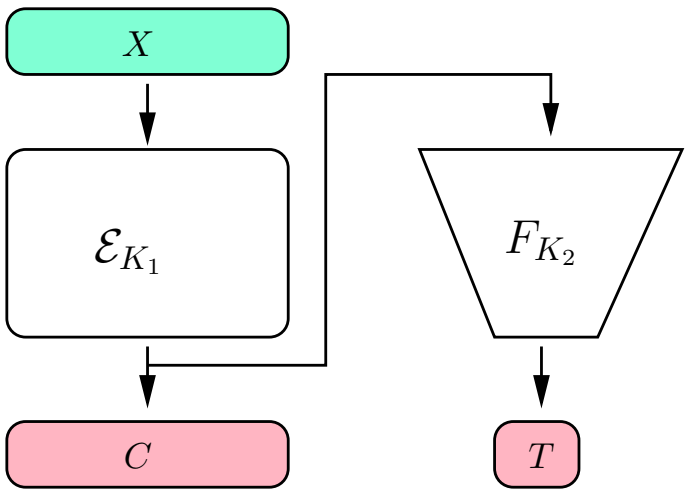
- Modes of operation (OCB, EAX, CCM, GCM);
- Dedicated constructions (Helix/Phelix, Grain128).

They use nonces to achieve confidentiality in the presence of repeated queries or blocks.

Furthermore, some input must be authenticated but not encrypted (e.g., routing information). It is called associated data (AD).

Encrypt then MAC

It is rather easy to [provably secure] add authentication using a second key:



$$C = E_{K_1}(P); \quad T = MAC_{K_2}(C).$$

It is substantially more difficult [to prove it secure] with a single key.

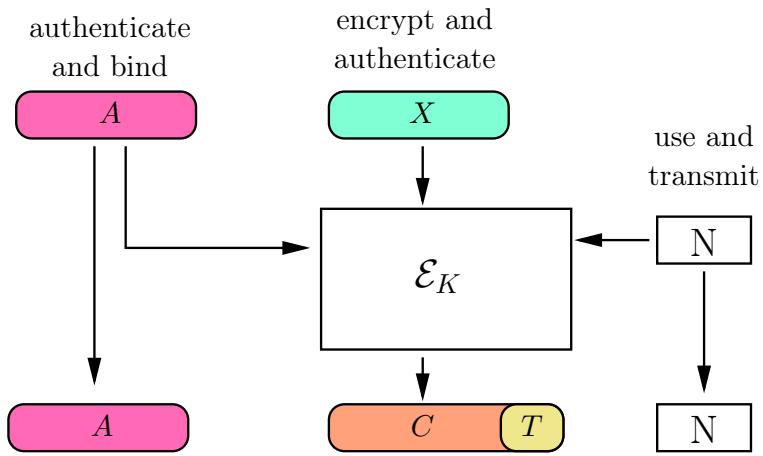
Authenticated encryption with associated data

Encryption:

$$\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{X} \to \mathcal{C}$$

Decryption:

$$\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \to \mathcal{X} \cup \{\perp\}.$$



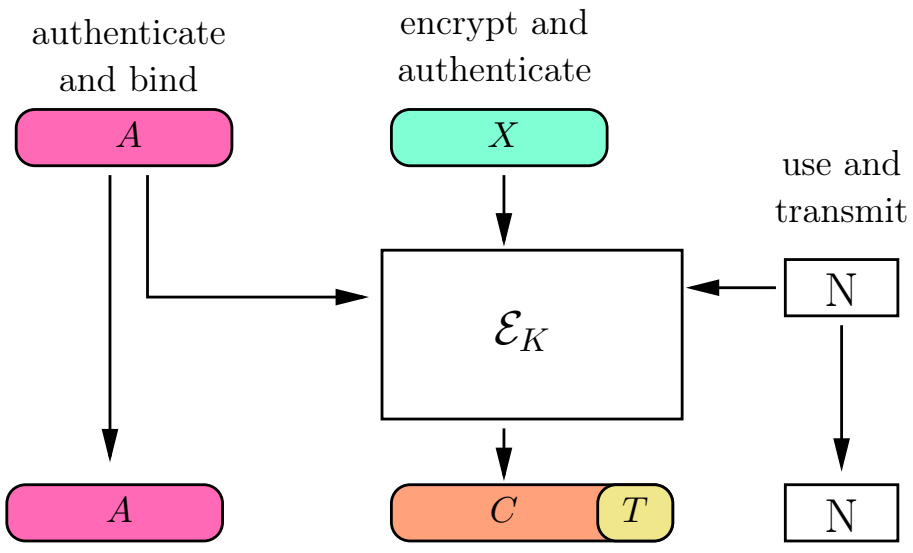
Confidentiality:

- Ciphertexts indistinguishable from random strings;

Data integrity:

- Most of seemingly valid ciphertexts decrypt to $\perp$.

Use for key wrap?



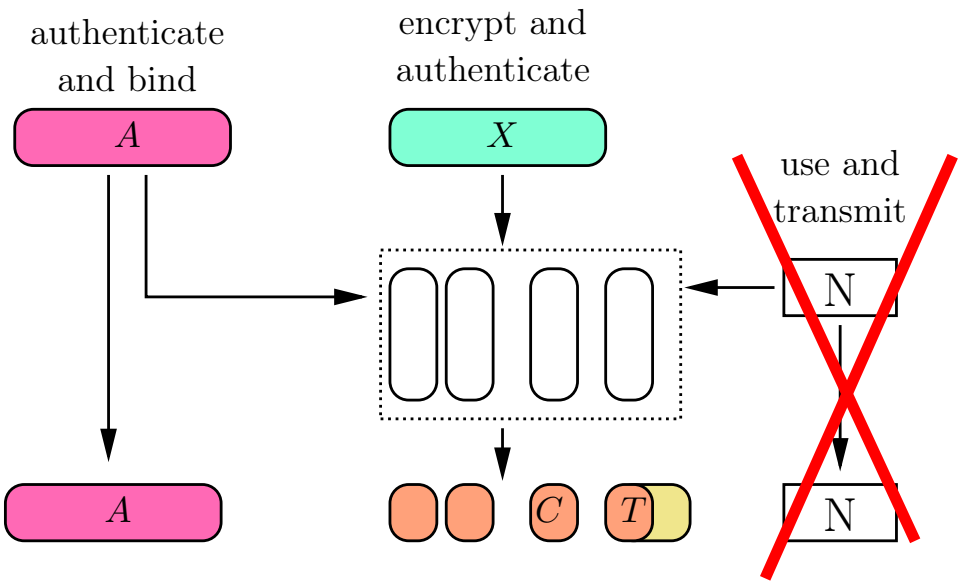
Too much for a key wrap scheme:

- Uses nonces or random IVs.

Also often not misuse-resistant.

No nonce?

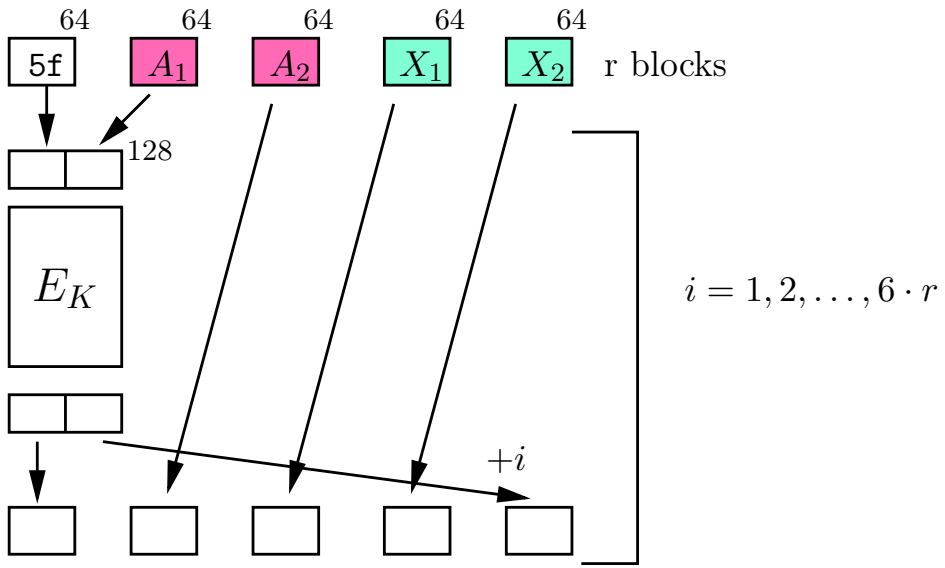
It is difficult to construct a nonce-free AE, and two passes are usually required.



Confidentiality can not be delivered with one pass only — because of the block structure.

Existing solutions

NIST Key Wrap scheme (AES-KW)



- 12× overhead;
- Expansion by the size of AD;
- No provable security (though probably good one);
- No cryptanalysis;
- At least 2^{-64} forgery probability;
- Unparallelizable.

Deterministic Authenticated Encryption

Encryption:

$$\mathcal{E}: \mathcal{K} \times \mathcal{A} \times \mathcal{X} \to \mathcal{C}$$

Decryption:

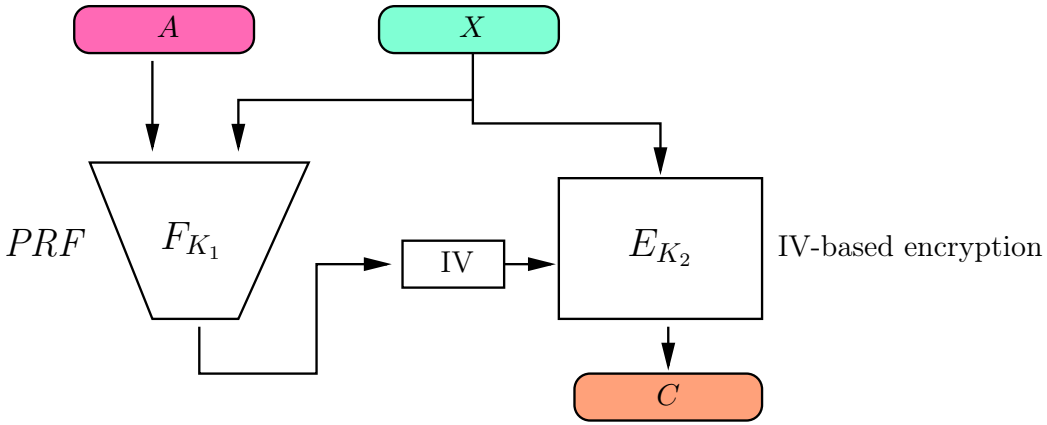
$$\mathcal{D}: \mathcal{K} \times \mathcal{A} \times \mathcal{C} \to \mathcal{X} \cup \{\perp\}.$$

Deterministic Authenticated Encryption (DAE, Rogaway-Shrimpton 2006):

$$(\mathcal{E}(\cdot), \mathcal{D}(\cdot)) \approx (\$(\cdot), \perp(\cdot)); \qquad K \xleftarrow{\$} \mathcal{K}.$$

Indistinguishability from random oracle and "always invalid" oracle.

Synthetic IV (SIV) scheme (Rogaway-Shrimpton 2006)



- 2× overhead;
- Two keys;
- Combined, not integrated scheme;
- Only encryption parallelizable;
- 64-bit security with AES.

Other schemes

The Key-Wrap concept (Gennaro-Halevi, 2009):

- Random-Plaintext secure (wrapped keys out of attacker's control);
- Similar ciphertext integrity notion;
- Hash-then-CTR and Hash-then-CBC secure schemes, which require both block cipher and a hash function.

More sophisticated schemes (HBS, BTM, etc.).

Hard to deliver the security beyond the birthday bound (64 bits if AES).

Our proposal

Goals

Our goals:

- Design a key-wrapping scheme with provable 128-bit security;
- Handle associated data;
- Make the scheme compact and simple;
- Use well-known wide building blocks of Keccak;
- Shorten the security (cf. the GCM proof bug found after 10 years).

Our restrictions:

- Only short (< 1400 bits) keys are handled;
- Need of the inverse Keccak permutation;
- Ciphertext expansion.

Goals

Our goals:

- Design a key-wrapping scheme with provable 128-bit security;
- Handle associated data;
- Make the scheme compact and simple;
- Use well-known wide building blocks of Keccak;
- Shorten the security (cf. the GCM proof bug found after 10 years).

Our restrictions:

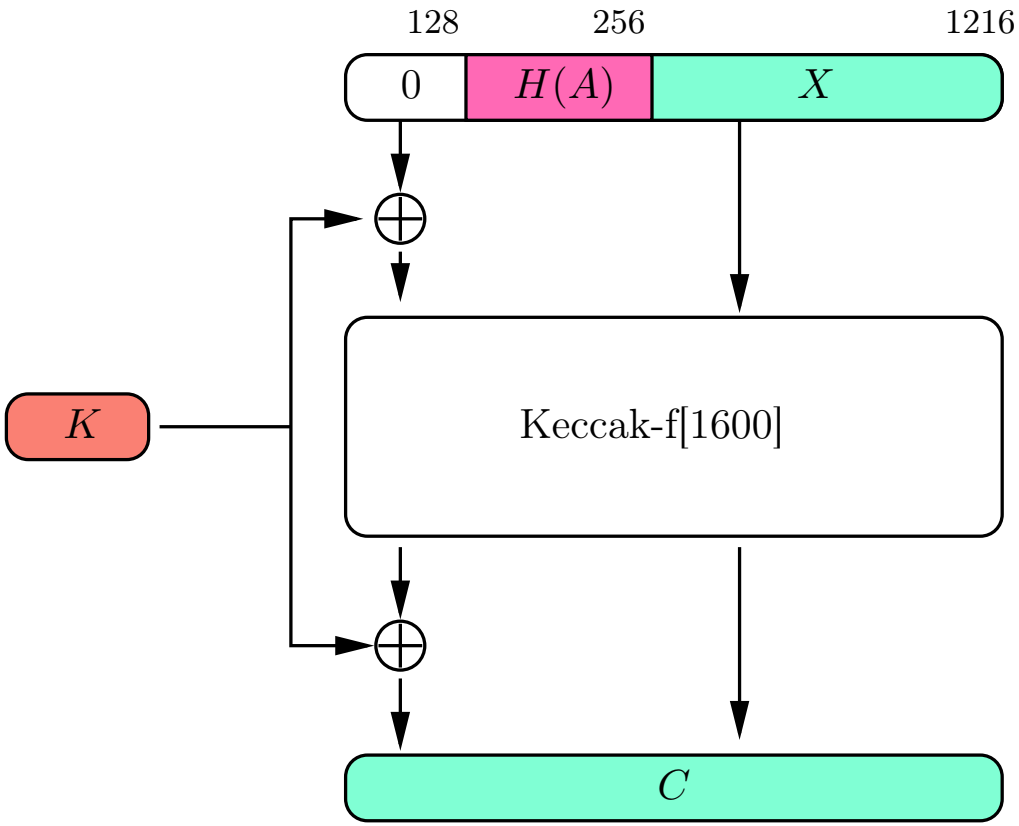
- Only short (< 1400 bits) keys are handled;
- Need of the inverse Keccak permutation;
- Ciphertext expansion.

We found the AES block of 128 bit too short for making a simple scheme.

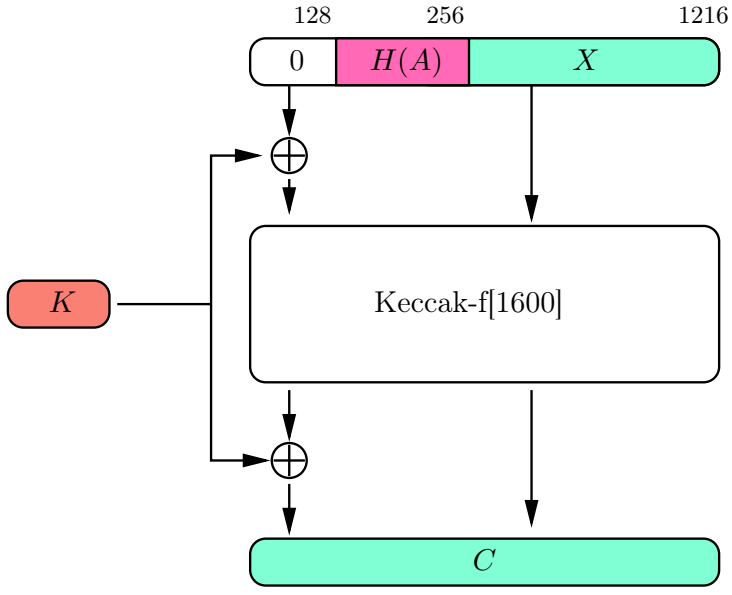
Key wrap scheme I

Encryption (X — plaintext for wrapping):

- Compute hash of associated data with [collision-resistant] Keccak-256 — $H(A)$;
- Apply Keccak-f[1600] to $K||H(A)||X$, where K — master key.
- XOR the master key K to the output.



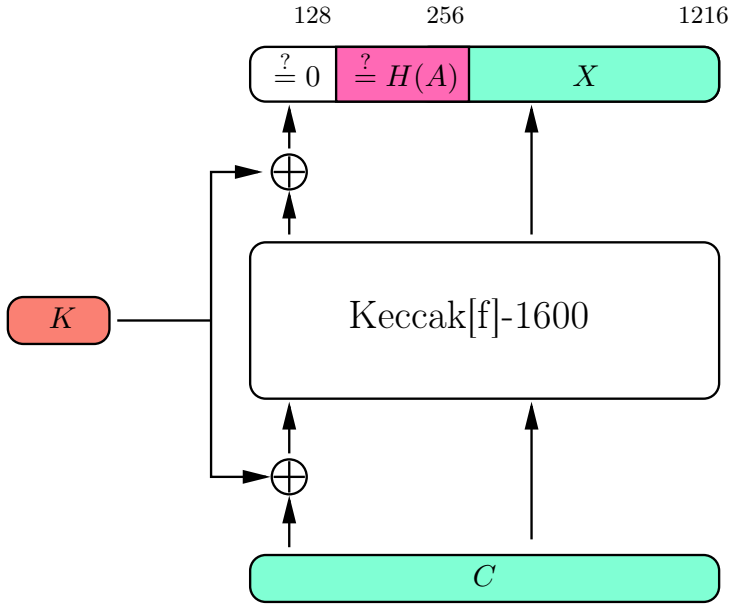
Confidentiality



Confidentiality (Left-or-Right) for random permutation (proof intuition):

- Submit two plaintexts on your own;
- Unable to figure out inputs and outputs of the permutation unless the key is guessed;
- Two ciphertexts become indistinguishable.

Ciphertext integrity

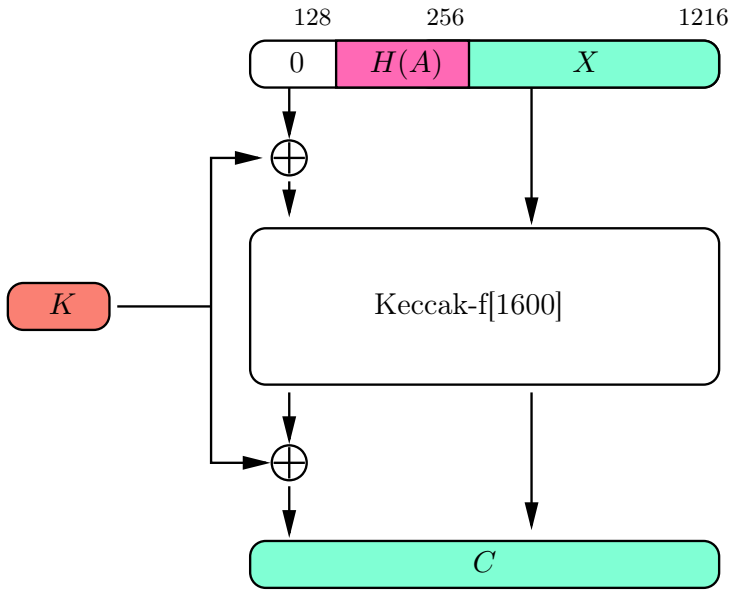


Ciphertext integrity for random permutation (proof intuition):

- Request to decrypt fresh pairs (A, C);
- Ciphertext must be fresh, otherwise there is mismatch in $H(A)$ due to collision resistance;
- If ciphertext is fresh, then it is a new query to π^{-1}, and $H(A)$ is obtained with prob. $\approx 2^{-256}$.

Better scheme?

Some redundancy:



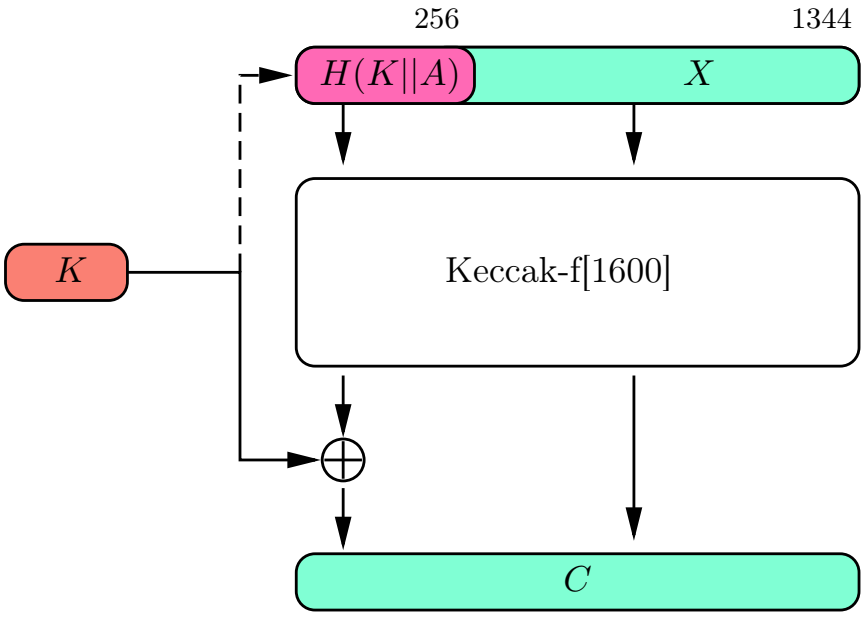
- 0 for confidentiality;
- $H(A)$ for integrity.

Combine?

Key wrap scheme II

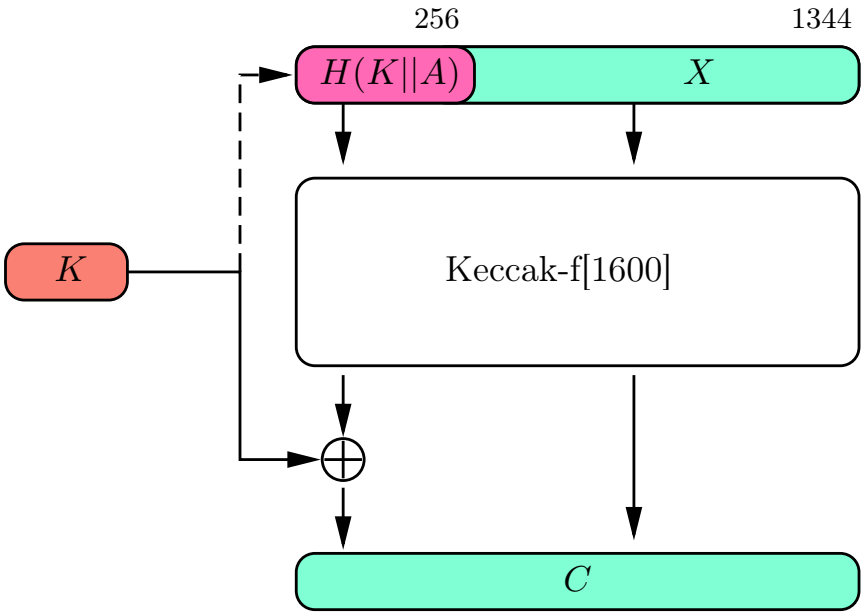
Encryption:

- Compute MAC of associated data with Keccak-256 — $H(K||A)$;
- Apply Keccak-f[1600] to $H(K||A)||X$;
- XOR the master key K to the output.



$H(K||A)$ supposed to be unpredictable, collision-resistant, and infeasible to match.

Key wrap scheme II



- Higher rate;
- Proof seems to be more difficult.

Summary

Assume other schemes use AES (as usually specified):

	Scheme 1	Scheme 2	AES-KW	SIV	HtCTR
Message length	1216	1344	Arbitrary		
Overhead	(1.3)	(1.2)	12	2	2
Expansion	≥ 384	≥ 256	$\lvert A\rvert + 64$	128	128
Parallelizable	-	-	No	Partly	Partly
Security proof	Working out DAE		No	DAE	KW
Block cipher	No	No	Yes	Yes	Yes
Hash function	Yes	Yes	No	Yes	Yes
Precompute AD	Yes	Yes	No	Yes	Yes
128-bit security	Yes	Yes	No	Not with AES	