AWS Key Management Service (KMS)

Handling cryptographic bounds for use of AES-GCM

Matthew Campagna
Amazon Web Services

Shay Gueron
Amazon Web Services
University of Haifa

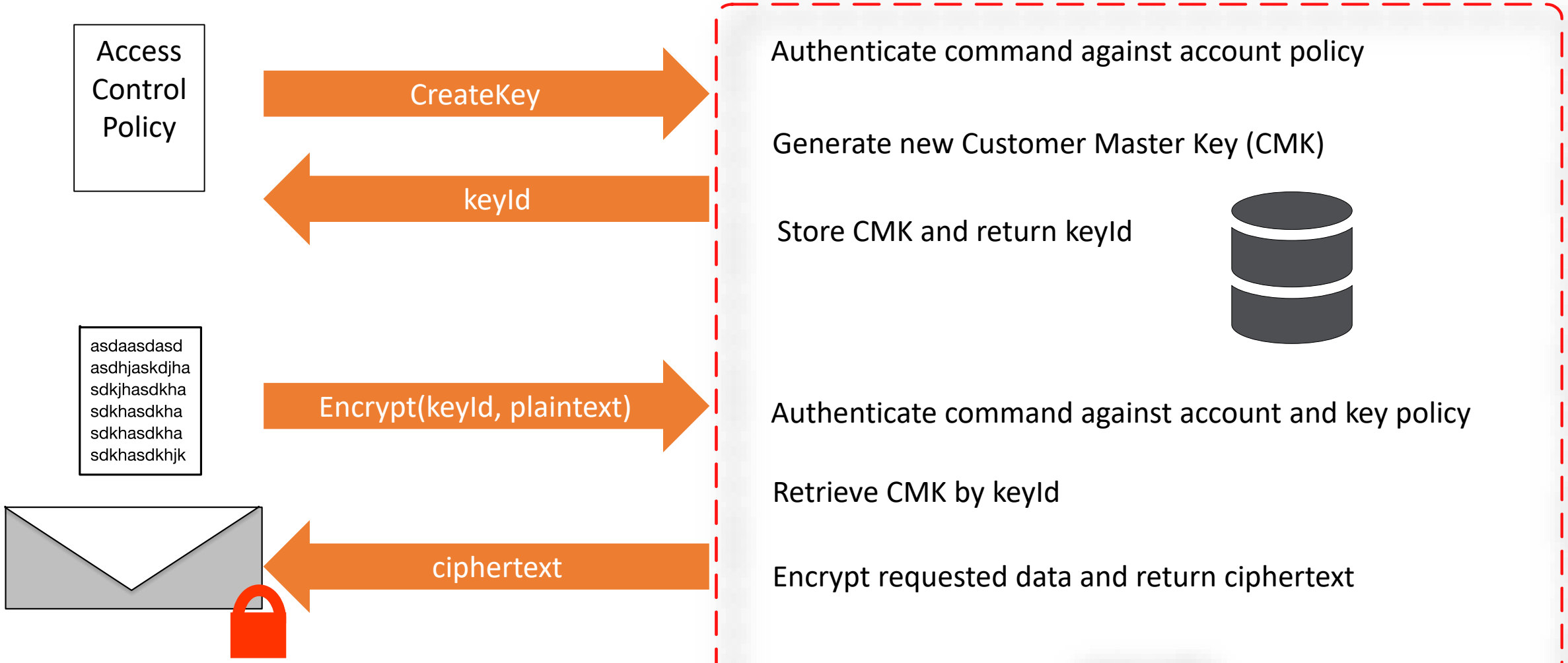
Outline

- The AWS Key Management Service
- Limitations on a naïve use of AES-GCM
- The AWS KMS Encryption Mode: a Derive-Key-AES-GCM instance
- Security bounds and limits of AWS KMS Encryption

What is KMS

- Amazon Web Services' Key Management Service (AWS KMS)
  - Is a web-based service
  - Provides a simple interface to generate, rotate and manage cryptographic keys
  - Operates as a cryptographic service provider for cryptographic keys and encryption of data
  - Configure for use in other AWS services to protect customer data
- Customer master keys are protected in hardware
  - Customer can use a key "implicitly", to encrypt files, and manage decryption requests access to other customers.

AWS Key Management Service (KMS)



Example – simple file encryption

- User A (with GenerateDataKey access to keyId)
  1. (dk, edk) = kms.GenerateDataKey(keyId)
  2. enc_file = encrypt_file(dk, file)
  3. Delete dk, store (edk, enc_file)

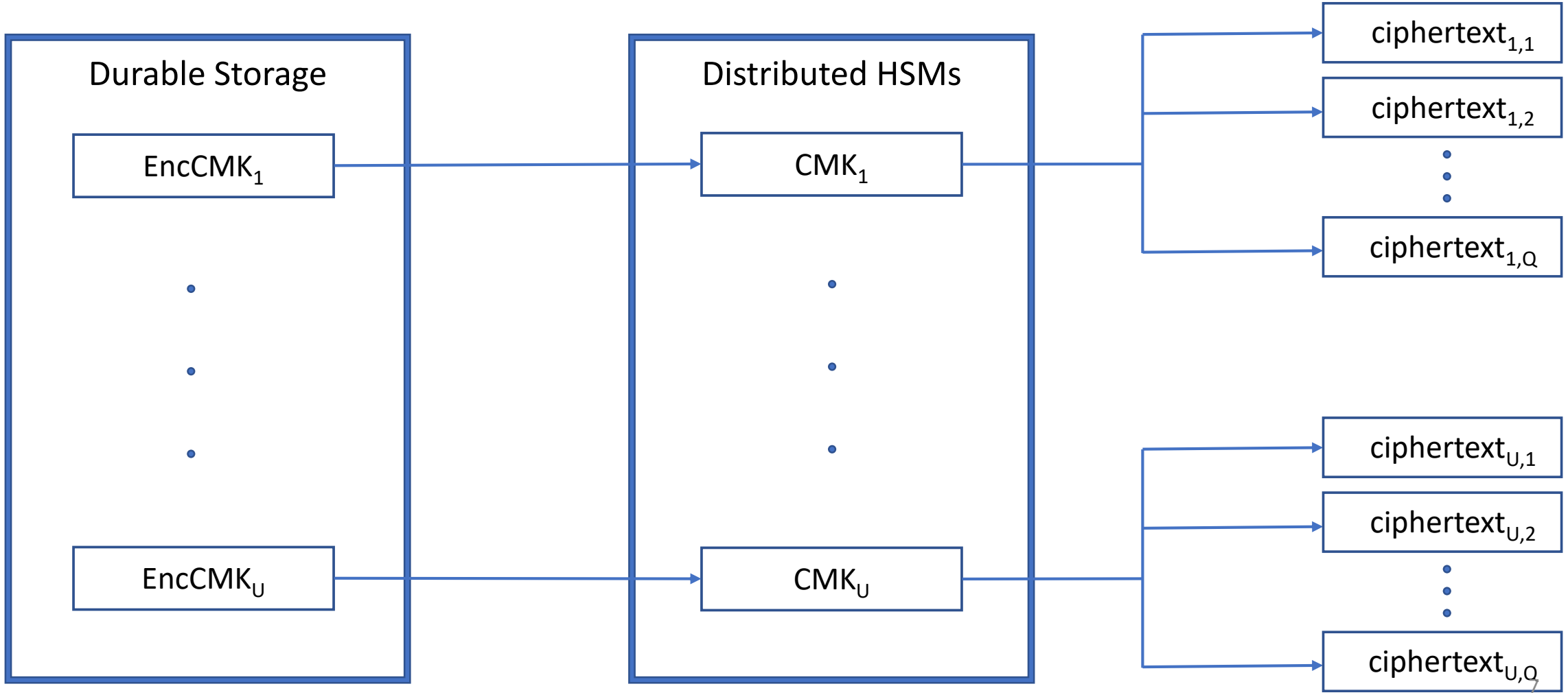
- User B (with Decrypt access to keyId)
  1. Retrive dk = kms.Decrypt(edk)
  2. Decrypt file = decrypt_file(dk, enc_file)
  3. Delete dk

AWS Key Management Service (KMS)

- CMKs are stored encrypted and only decrypted on service HSMs
- CMKs cannot leave the HSM security boundary
  - By default; there is a “import key” capability and the key can be generated externally
- Access is restricted by a limited set of audited APIs
- Customer plaintext & ciphertexts are not stored or logged by AWS KMS
- Encryption: uses AES-256-GCM with
  - Random 96-bit IV – used because of the distributed nature of the HSMs
  - Maximum plaintext size is 4096 bytes
  - Maximum additional authenticated data is (AAD) 8192 bytes
  - AAD is logged
- CMKs can be configured to rotate yearly
  - Newest key used for all new encryption requests
  - Old keys used for decryption only

A Naïve KMS: encryption by using CMK directly

Each encryption is an invocation of AES256-GCM with a random 96-bit IV



The cloud is still small

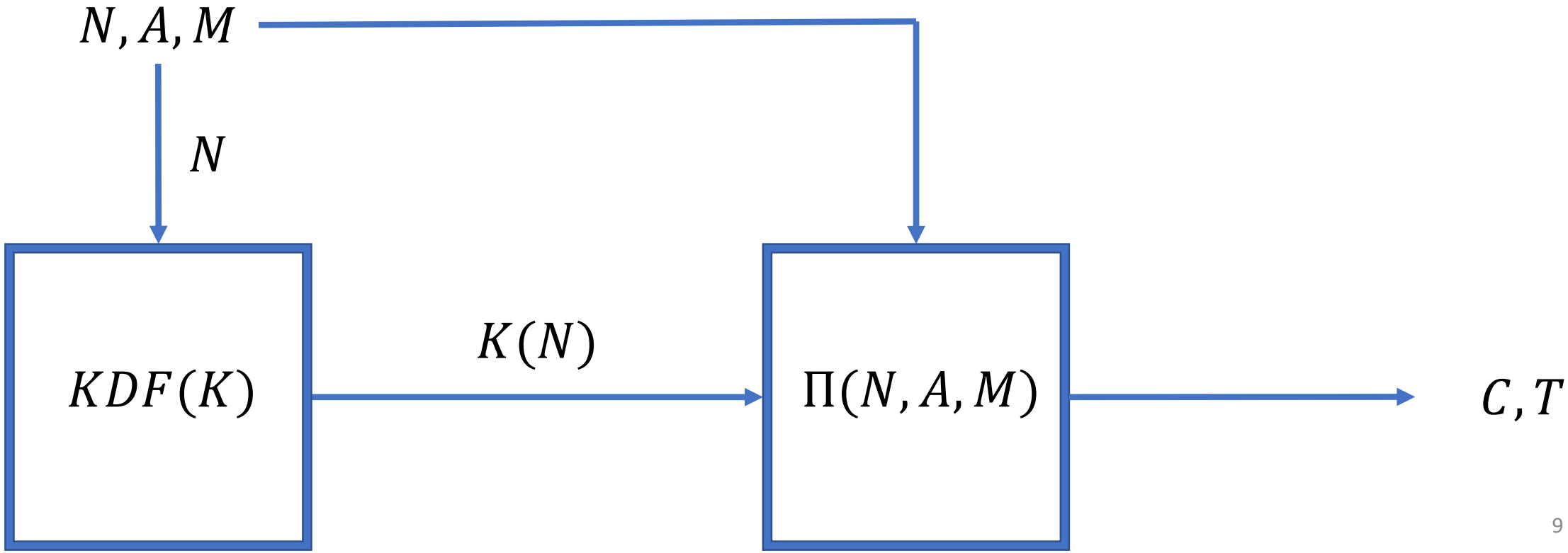
- AES-GCM with a random IV limits the lifetime of a key
  - NIST requirements: never invoke AES-GCM with the same (key, iv) pair with probability > 2^{-32}.
- This restricts the number of encryptions that can be done under a single CMK (without rotation) to 2^{32} encryptions.
  - **4 billion is not a big number in the cloud** - need to protect trillions of objects
- The cloud provider must minimize (CMK, IV) collisions across *U* users.
  - CMK collisions have negligible probability: with U users: $U^2/2^{257}$.

A (key, iv)-collision results in loss of authentication of all ciphertexts encrypted with that key, and exposes plaintexts of corresponding ciphertexts with the (key, iv)-collision

Derive Key Mode: extending the lifetime of a key

Gueron Lindell CCS 2017

Context: a nonce based encryption scheme $\Pi(N, A, M)$
Setup key: K Input: N, A, M
Step 1: Apply a KDF to derive a new (per-nonce) key $k_N = KDF_K(N)$
Step 2: Use k_N with $\Pi(N, A, M)$



Derive Key Mode: security bounds

Gueron Lindell CCS 2017

- The advantage of adversary $\mathcal{A}$ with N different nonces is the sum of the advantage of $\mathcal{A}$
  1. for N key derivations
  2. in the multi-instance experiment with N ciphers
  3. with the original scheme when a new truly random function is used instead of each derived block cipher

#1 depends on how the per-nonce keys are derived
#2 depends on what we are willing to assume on the block cipher
#3 depends on the scheme

Example: CTR mode with unique nonces

Gueron Lindell CCS 2017

- AES-CTR mode with unique 96-bit nonces
- $\boldsymbol{B}$ = total # of blocks encrypted; $\boldsymbol{B}\mathbf{max}$ = maximum # blocks in a msg

CTR: advantage: $\frac{\boldsymbol{B}^{\mathbf{2}}}{\mathbf{2}^{\mathbf{129}}}$ Derive-Key CTR: advantage dominated by $\frac{\boldsymbol{N}\cdot\boldsymbol{B}\mathbf{max}^{\mathbf{2}}}{\mathbf{2}^{\mathbf{128}}}$

- Example: 2^{48} encryptions of length 2^{16}
  - CTR is **broken** with probability ½
  - Derive-Key CTR: advantage is **just** 2^{-46}
- Derive-Key CTR: can encrypt even 2^{64} plaintexts of length 2^{16} blocks with advantage of 2^{-32}. This is **way beyond** the birthday barrier

The KMS Encryption Mode

Input:

Random 16-byte nonce N and 12-byte IV,

$A = a_1, a_2, \ldots, a_d$, additional authenticated data, and

$M = m_1, m_2, \ldots, m_p$, plaintext (blocks)

Keys:

CMK (master key)

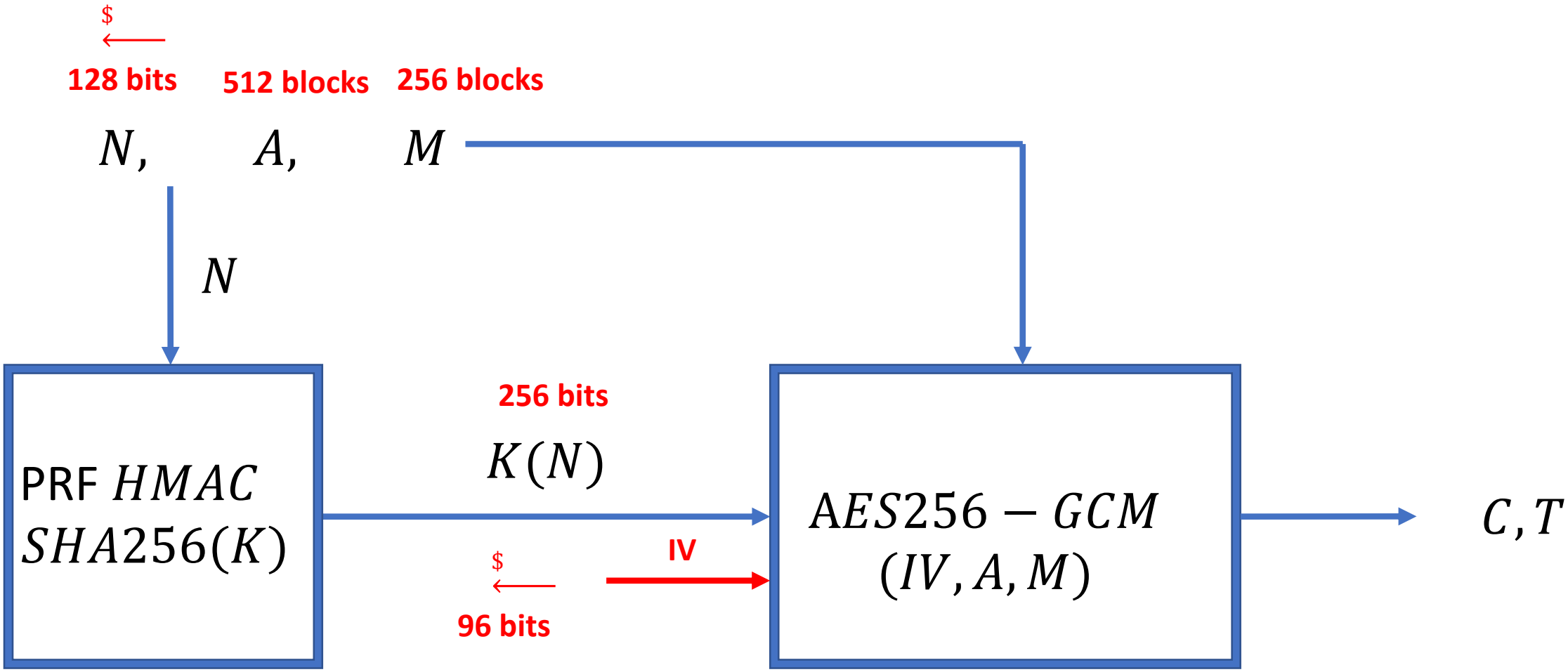
Steps:

1. Select uniform random $N[16]$ and $IV[12]$
2. Derive a 32-byte Wrapping Key $WK = KDF(CMK, N)$
3. Then $(C, T) = AES\_256\_GCM(WK, IV, A, M)$

$$|AAD| = d \leq 512$$
$$|M| = p \leq 256$$

KDF: NIST SP800-108 KDF (in Counter Mode with PRF HMAC_SHA256)

The KMS Encryption Mode



Two perspectives to consider

- A customer is concerned about the protection of his master key, wrapping keys, and encrypted plaintexts
  - User perspective ➔ multi-key scenario (induced by the Derive key mode)
- Cloud provider is concerned about the protection of all the customers' master keys, wrapping keys and encrypted plaintexts
  - Cloud perspective ➔ multi-user & multi-key scenario

Customer’s perspective

- Within the key-space of a Customer Master Key we have multiple wrapping keys, derived from the master key.
  - What is the probability of a (key, iv) reuse?
  - What advantage does an adversary has, in distinguishing the use of AES, (assumed to be a good approximation of a pseudorandom-permutation), from a pseudorandom-function?
  - What is the protection against a forgery attack on the authentication?
  - What is the probability of recovering one of the wrapping keys?

User perspective: (derived-key, iv) collision

- Out of Q key derivations from 128-bit nonces:
  - Prob (at least two nonces collide) $\leq Q^2/2^{129}$
  - Prob (at least one case of 3 or more such collisions) $\leq Q^3/(6 * 2^{256})$
    - Negligible for $Q \leq 2^{64}$
- (lemma) Prob (10 keys get repeated) < 2^{-32}
  - With probability higher than (1-2^{-32}), at most $(Q - 20)$ unique keys were used for encrypting a single message (256B plaintext + 512B AAD blocks)
    - No IV collision on such keys
  - At most 10 keys were used for encrypting 2 messages
  - Prob (IV collision on the same derived-key) $1 - (1 - 2^2/2^{97})^{10} \approx 1/2^{91}$

Q can be as large as 2^{64}, before remotely approaching NIST probability requirement on (derived-key, IV) collisions

User perspective: PRP-PRF advantage

- Longest message has 256 blocks ($256 + 1 = 257$ encrypted plaintext blocks)

$$Adv \leq 257^2/2^{129} \leq \frac{1}{2^{113}}$$

- When two derived-keys collide: maximum # of blocks is 514. Thus

$$Adv \leq 514^2/2^{129} < 1/2^{111}$$

- User with $Q \leq 2^{64}$encryptions: high chance there are at most 10 key collisions

$$Adv \leq 10 * \frac{1}{2^{111}} + (Q - 20) * \frac{1}{2^{113}} = 5 * \frac{1}{2^{111}} + \frac{Q}{2^{113}}$$

Up to $Q < 2^{64}$ the indistinguishability advantage is less than $1/2^{32}$

User perspective: forgery protection

- Forgery success probability with $|A| + |M| + 1 = 769$ blocks is at most

$$\frac{769}{2^{128}}$$

- With Q_D forgery attempts, this is no more than

$$Q_D\left(\frac{769}{2^{128}}\right)$$

- Decryption is limited to 1200 transactions per second (tps) via an authenticated API under the access control policy.

Forgery is not a concern

User perspective: key recovery

- Multi-key scenario: if the same block is encrypted X times under different keys then the probability to recover one of the keys is

$$\frac{X}{2^{keylength}}$$

- In the KMS system, $keylength = 256$
- Prob (random 96-bit IV repeats 5+ times across 2^{64} users) $\leq \frac{2^{320}}{5! * 2^{384}}$
  - Remains negligible ➔ we can assume $X \leq 5$

Up to $Q < 2^{64}$ the key recovery probability (multi key scenario) is negligible

Cloud-provider perspective

- What is the probability of a (derived-key, iv)-collision across all users?
- What is the advantage of an adversary to distinguish between AES, as a pseudorandom-permutations, from a pseudorandom-function across all user's (derived-key, iv)-pairs?
- Forgeries are limited to decryptions under a specific master key, and is no different than the customer perspective.
- What is the probability of recovering one of the wrapping keys?

Cloud perspective: (derived-key, iv) collision

- Probability of customer master key collision among U users is at most
$$U^2/2^{257}$$
- Probability for a user's (derived-key, iv) collision ($Q < 2^{64}$) is at most
$$1 - (1 - 2^2/2^{97})^{10} \approx 1/2^{91}$$

- So the probability of a (derived-key, iv) collision for at least one of U users is at most
$$1 - (1 - 1/2^{91})^U \approx U/2^{91}$$

Cloud perspective: derived-key collision

- What is the probability that a derived key across U users doing Q encryptions collide?
- Two possibilities
  - random[16], CMK[32] collide on the KDF input, probability $\approx (U * Q)^2/2^{385}$
  - Collide on KDF output, probability $\approx (U * Q)^2/2^{257}$

- Even for large Q and U (say 2^{64} and 2^{48} resp.), this probability is

$$\approx (2^{64} * 2^{48})^2/2^{257} = (2^{112})^2/2^{257} = 1/2^{33}$$

Cloud perspective: PRP-PRF advantage

- The longest message is 256 in blocks ($256 + 1 = 257$ blocks encrypted with AES-GCM)

$$Adv \leq 257^2/2^{129} \leq \frac{1}{2^{111}}$$

- For U users doing Q encryptions this becomes

$$Adv \leq \frac{U * Q}{2^{111}}$$

- Even when U and Q are large, 2^{40} users doing 2^{50} encryptions this advantage remains small $< 1/2^{20}$

Cloud perspective: key recovery

- Block cipher in a multi key scenario: if one block is encrypted X times under different keys, then the probability to recover one of the keys is

$$\frac{X}{2^{keylength}}$$

  - Recall: in KMS $keylength = 256$
- How large can X be?
  - Prob random 96-bit IV repeats more than X (=16) times across U users, each encrypting Q files) $\leq \frac{(UQ)^{16}}{16! * 2^{1440}} = \frac{1}{16!}$
  - $U = 2^{40}, Q = 2^{50}, X = 16$ ➔ $\leq 2^{-44}$

Multi user & multi-key: key recovery is not a concern

Summary

- AWS KMS is a secure cloud-scale implementation that can support
  - up to 2^{40} master keys, and
  - each master key can be used to perform 2^{50} encryptions.
- Earth‘s population (end of 2017): 7.2 billion ~$2^{32.7}$
- Additional details can be found here:
  - https://acmccs.github.io/papers/p1019-gueronA.pdf
  - https://d0.awsstatic.com/whitepapers/KMS-Cryptographic-Details.pdf