A GENERIC CIPHERTEXT FORMAT

draft-sheffer-ietf-ciphertext-format-01

Yaron Sheffer
Gleb Keselman
Yoav Nir

SecDispatch, IETF-110
There are standards for "raw ciphertext", the direct result of encryption

This is not sufficient if you have to manage trillions of ciphertexts, stored at rest

Need (standard) ciphertext metadata

Standard format to denote key identity, key version...

Used for storage: must be efficient

Must be extensible

Allows automated detection and attribution of ciphertext

Supports granular key management: key wrapping and key derivation
DETECTION AND ATTRIBUTION

Big Data is regularly scanned by data classification tools for many reasons

- Classify the data
- Detect misclassified (e.g., too sensitive) data
- Resolve data quality issues
- Detect PII for regulatory compliance

To ensure scans can be acted on, data needs to be attributed back to its producer

Detection: scanners recognize encrypted data with high probability
Attribution: scanners and associated tools can find out who owns the data
A fixed first octet
- Good for ciphertext detection

A format version

A variable, structured header
- As of -01, using CBOR (thanks for the feedback!)
Defined using CDDL

```cddl
var_header = {
    K_KEY_PROVIDER: uint,
    K_KEY_ID: bstr,
    ? K_KEY_VERSION: uint,
    ? K_AUX_DATA: bstr,
    ? K_NONCE : bstr,
    ? K_AUTH_TAG : bstr,
    ? K_AAD : bstr,
    *uint => any ; extensions
}
```

Which key management system owns the key?

Key identity, relative to the Key Provider

Key versioning, a.k.a. rotation

Support key derivation

Support AEAD
IMPLEMENTATIONS

Intuit implements a similar scheme internally, for very large amounts of data.

AWS (Encryption SDK) and Google (in the Tink library) each define a different format for application-level encrypted data.
NEXT STEPS

WG-forming BOF?

Other ideas?