Authors: R. Moskowitz   S. Card   A. Wiethuechter
         HTT Consulting   AX Enterprize   AX Enterprize

UAS Operator Privacy for RemoteID Messages

Abstract

This document describes a method of providing privacy for UAS
Operator/Pilot information specified in the ASTM UAS Remote ID and
Tracking messages. This is achieved by encrypting, in place, those
fields containing Operator sensitive data using a hybrid ECIES.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a mechanism to provide privacy in the ASTM
Remote ID and Tracking messages [F3411-19] by encrypting, in place,
those fields that contain sensitive UAS Operator/Pilot information.
Encrypting in place means that the ciphertext is exactly the same
length as the cleartext, and directly replaces it.

An example of and an initial application of this mechanism is the 8
bytes of UAS Operator/Pilot (hereafter called simply Operator)
longitude and latitude location in the ASTM System Message (Msg Type
0x4). This meets the Drip Requirements [RFC9153], Priv-01.

It is assumed that the Operator, via the UAS, registers an operation
with its USS. During this operation registration, the UAS and USS
exchange public keys to use in the hybrid ECIES. The USS key may be
long lived, but the UAS key SHOULD be unique to a specific

operation. This provides protection if the ECIES secret is exposed
from prior operations.

The actual Tracking message field encryption MUST be an "encrypt in
place" cipher. There is rarely any room in the tracking messages for
a cipher IV or encryption MAC (AEAD tag). There is rarely any data
in the messages that can be used as an IV. The AES-CFB16 mode of
operation proposed here can encrypt a multiple of 2 bytes.

The System Message is not a simple, one-time, encrypt the PII with
the ECIES derived key. The Operator may move during a operation and
these fields change, correspondingly. Further, not all messages will
be received by the USS, so each message's encryption must stand on
its own and not be at risk of attack by the content of other
messages.

Another candidate message is the optional ASTM Operator ID Message
(Msg Type 0x5) with its 20 character Operator ID field. The Operator
ID does not change during an operation, so this is a one-time
encryption operation for the operation. The same cipher SHOULD be
used for all messages from the UAS and this will influence the
cipher selection.

Future applications of this mechanism may be provided. The content
of the System Message may change to meet CAA requirements, requiring
encrypting a different amount of data. At that time, they will be
added to this document.

Editor note: The Rules for allowing encryption need to be updated to
handle the UA operating in Broadcast Remote ID only mode. That is
conditions where the USS cannot notify the UAS to stop encrypting.

2. Terms and Definitions

2.1. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in
BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

2.2. Definitions

See Section 2.2 of [RFC9153] for common DRIP terms.

ECIES
    Elliptic Curve Integrated Encryption Scheme. A hybrid encryption
    scheme which provides semantic security against an adversary who
    is allowed to use chosen-plaintext and chosen-ciphertext attacks.

Keccak (KECCAK Message Authentication Code):
   The family of all sponge functions with a KECCAK-f permutation as
   the underlying function and multi-rate padding as the padding
   rule. It refers in particular to all the functions referenced
   from [NIST.FIPS.202] and [NIST.SP.800-185].

KMAC (KECCAK Message Authentication Code):
   A PRF and keyed hash function based on KECCAK.

3. The Operator - USS Security Relationship

All CAAs have rules defining which UAS must be registered to operate
in their National Airspace. This includes UAS and Operator
registration in a USS. Further, operator's are expected to report
flight operations to their USS. This operation reporting provides a
mechanism for the USS and operator to establish an operation
security context. Here it will be used to exchange public keys for
use in ECIES.

The operator's ECIES public key SHOULD be unique for each operation.
The USS ECIES public key may be unique for each operator and
operation, but not required. For best post-compromise security
(PCS), the USS ECIES public key should be changed over some
operational window.

The public key algorithm should be Curve25519 [RFC7748].
Correspondingly, the ECIES 128 bit shared secret should be generated
using KMAC [NIST.SP.800-185].

3.1. ECIES Shared Secret Generation

The KMAC function provides a new, more efficient, key derivation
function over HKDF [RFC5869]. This will be referred to as KKDF.

HKDF needs a minimum of 4 hash functions (e.g. SHA256). KKDF does an
equivalent shared secret generation in a single Keccak Sponge
operation.

When the USS - UAS Operation Security Context is established, the
UAS provides its UAS ID (null padded to 20 characters per
[F3411-19]) and a 256 bit random nonce to the USS. These are inputs,
along with the ECDH keys to produce the shared secret as follows.

A 64 bit UNIX timestamp for the operation time is also included in
the Operation Security Context. This will be used in the IV
construction.

Per [NIST.SP.800-56Cr1], Section 4.1, Option 3:

```
Shared Secret = KMAC128(salt, IKM, L, S)
```

L is the derived key bit length. Since only a single key is needed,
L=128.

S is the byte string 01001011 || 01000100 || 01000110, which
represents the sequence of characters "K", "D", and "F" in 8-bit
ASCII.

```
salt = Nonce-USS |  Nonce-UAS
```

There are special security considerations for IKM per [RFC7748]. The
IKM as follows:

```
IKM = Diffie-Hellman secret | USS-ID | RID
```

4. System Message Privacy

The System Message contains 8 bytes of Operator specific
information: Longitude and Latitude of the Remote Operator (Pilot in
the field description) of the UA. The GCS MAY encrypt these as
follows.

Editors Note: The next version of [F3411-19], currently in ballot,
is adding a 2 byte Operator Altitude field, thus increasing the
Operator specific information to 10 bytes. This change will be
delineated via Protocol Version field. It is this future shift from
a multiple of 4 bytes to a multiple of 2 bytes that is the reason to
change from CFB32 in earlier drafts to CFB16 used now.

The 8 bytes of Operator information are encrypted, using the ECIES
derived 128 bit shared secret, with one of the cipher's specified
below. The choice of cipher is based on USS policy and is agreed to
as part of the operation registration. AES-CFB16 is the recommended
default cipher.

ASTM Remote ID and Tracking messages [F3411-19] SHOULD be updated to
allow Bit 5 of the Flags byte in the System Message set to "1" to
indicate the Operator information is encrypted.

The USS similarly decrypts these 8 bytes and provides the
information to authorized entities.

4.1. Rules for encrypting System Message content

If the Operator location is encrypted the encrypted bit flag MUST be set to 1.

The Operator MAY be notified by the USS that the operation has entered a location or time where privacy of Operator location is not allowed. In this case the Operator MUST disable this privacy feature and send the location unencrypted or land the UA or route around the restricted area.

If the UAS looses connectivity to the USS, the privacy feature SHOULD be disabled or land the UA.

If the operation is in an area or time with no Internet Connectivity, the privacy feature MUST NOT be used.

4.2. Rules for decrypting System Message content

An Observer receives a System Message with the encrypt bit set to 1. The Observer sends a query to its USS Display Provider containing the UA's ID and the encrypted fields.

The USS Display Provider MAY deny the request if the Observer does not have the proper authorization.

The USS Display Provider MAY reply to the request with the decrypted fields if the Observer has the proper authorization.

The USS Display Provider MAY reply to the request with the decrypting key if the Observer has the proper authorization.

The Observer MAY notify the USS through its USS Display Provider that content privacy for a UAS in this location/time is not allowed. If the Observer has the proper authorization for this action, the USS notifies the Operator to disable this privacy feature.

5. Operator ID Message Privacy

The Operator ID Message contains the 20 byte Operator ID. The GCS MAY encrypt these as follows.

The 20 bytes Operator ID is encrypted, using the ECIES derived 128 bit shared secret, with one of the cipher's specified below. The choice of cipher is based on USS policy and is agreed to as part of the operation registration. AES-CFB16 is the recommended default cipher.

ASTM Remote ID and Tracking messages [F3411-19] SHOULD be updated to allow Operator ID Type in the Operator ID Message set to "1" to indicate the Operator ID is encrypted.

The USS similarly decrypts these 20 bytes and provides the information to authorized entities.

5.1. Rules for encrypting Operator ID Message content

If the Operator ID is encrypted the Operator ID Type field MUST be set to 1.

The Operator MAY be notified by the USS that the operation has entered a location or time where privacy of Operator ID is not allowed. In this case the Operator MUST disable this privacy feature and send the ID unencrypted or land the UA or route around the restricted area.

If the UAS looses connectivity to the USS, the privacy feature SHOULD be disabled or land the UA.

If the operation is in an area or time with no Internet Connectivity, the privacy feature MUST NOT be used.

5.2. Rules for decrypting Operator ID Message content

An Observer receives a Operator ID Message with the Operator ID Type field set to 1. The Observer sends a query to its USS Display Provider containing the UA's ID and the encrypted fields.

The USS Display Provider MAY deny the request if the Observer does not have the proper authorization.

The USS Display Provider MAY reply to the request with the decrypted fields if the Observer has the proper authorization.

The USS Display Provider MAY reply to the request with the decrypting key if the Observer has the proper authorization.

The Observer MAY notify the USS through its USS Display Provider that content privacy for a UAS in this location/time is not allowed. If the Observer has the proper authorization for this action, the USS notifies the Operator to disable this privacy feature.

6. Cipher choices for Operator PII encryption

6.1. Using AES-CFB16

CFB16 is defined in [NIST.SP.800-38A], Section 6.3. This is the Cipher Feedback (CFB) mode operating on 16 bits at a time. This

variant of CFB can be used to encrypt any multiple of 2 bytes of cleartext.

The Operator includes a 64 bit UNIX timestamp for the operation time, along with its operation pubic key. The Operator also includes the UA MAC address (or multiple addresses if flying multiple UA).

The 128 bit IV for AES-CFB16 is constructed by the Operator and USS as: SHAKE128(MAC|UTCTime|Message_Type, 128). Inclusion of the ASTM Message_Type ensures a unique IV for each Message type that contains PII to encrypt.

AES-CFB16 would then be used to encrypt the Operator information.

6.2. Using a Feistel scheme

If the encryption speed doesn't matter, we can use the following approach based on the Feistel scheme. This approach is already being used in format-preserving encryption (e.g. credit card numbers). The Feistal scheme is explained in Appendix A.

6.3. Using AES-CTR

If 2 bytes of the Message can be set aside to contain a counter that is incremented each time the Operator information changes, AES-CTR can be used as follows.

The Operator includes a 64 bit UNIX timestamp for the operation time, along with its operation pubic key. The Operator also includes the UA MAC address (or multiple addresses if flying multiple UA).

The high order bits of an AES-CTR counter is constructed by the Operator and USS as: SHAKE128(MAC|UTCTime|Message_Type, 112). Inclusion of the ASTM Message_Type ensures a unique IV for each Message type that contains PII to encrypt.

AES-CTR would then be used to encrypt the Operator information.

7. DRIP Requirements addressed

This document provides solution to PRIV-1 for PII in the ASTM System Message.

8. ASTM Considerations

ASTM will need to make the following changes to the "Flags" in the System Message (Msg Type 0x4):

Bit 5:
   Value 1 for encrypted; 0 for cleartext (see Section 4).

ASTM will need to make the following changes to the "Operator ID Type" in the Operator ID Message (Msg Type 0x5):

Operator ID Type
   Value 1 for encrypted Operator ID (see Section 5).

9. IANA Considerations

TBD

10. Security Considerations

An attacker has no known text after decrypting to determine a successful attack. An attacker can make assumptions about the high order byte values for Operator Longitude and Latitude that may substitute for known cleartext. There is no knowledge of where the operator is in relation to the UA. Only if changing location values "make sense" might an attacker assume to have revealed the operator's location.

10.1. CFB16 Risks

Using the same IV for different Operator information values with CFB16 presents a cyptoanalysis risk. Typically only the low order bits would change as the Operators position changes. The risk is mitigated due to the short-term value of the data. Further analysis is need to properly place risk.

10.2. Crypto Agility

The ASTM Remote ID Messages do not provide any space for a crypto suite indicator or any other method to manage crypto agility.

All crypto agility is left to the USS policy and the relation between the USS and operator/UAS. The selection of the ECIES public key algorithm, the shared secret key derivation function, and the actual symmetric cipher used for on the System Message are set by the USS which informs the operator what to do.

10.3. Key Derivation vulnerabilities

[RFC7748] warns about using Curve25519 and Curve448 in Diffie-Hellman for key derivation:

Designers using these curves should be aware that for each public key, there are several publicly computable public keys that are equivalent to it, i.e., they produce the same shared secrets. Thus using a public key as an identifier and knowledge of a shared secret as proof of ownership (without including the public keys in the key derivation) might lead to subtle vulnerabilities.

This applies here, but may have broader consequences. Thus two
endpoint IDs are included with the Diffie-Hellman secret.

10.4. KMAC Security as a KDF

Section 4.1 of NIST SP 800-185 [NIST.SP.800-185] states:

"The KECCAK Message Authentication Code (KMAC) algorithm is a PRF
and keyed hash function based on KECCAK . It provides variable-
length output"

That is, the output of KMAC is indistinguishable from a random
string, regardless of the length of the output. As such, the output
of KMAC can be divided into multiple substrings, each with the
strength of the function (KMAC128 or KMAC256) and provided that a
long enough key is used, as discussed in Sec. 8.4.1 of SP 800-185.

For example KMAC128(K, X, 512, S), where K is at least 128 bits, can
produce 4 128 bit keys each with a strength of 128 bits. That is a
single sponge operation is replacing perhaps 5 HMAC-SHA256
operations (each 2 SHA256 operations) in HKDF.

11. Normative References

[NIST.FIPS.202] Dworkin, M., "SHA-3 Standard: Permutation-Based Hash
           and Extendable-Output Functions", National Institute of
           Standards and Technology report, DOI 10.6028/nist.fips.
           202, July 2015, <https://doi.org/10.6028/nist.fips.202>.

[NIST.SP.800-185] Kelsey, J., Change, S., and R. Perlner, "SHA-3
           derived functions: cSHAKE, KMAC, TupleHash and
           ParallelHash", National Institute of Standards and
           Technology report, DOI 10.6028/nist.sp.800-185, December
           2016, <https://doi.org/10.6028/nist.sp.800-185>.

[NIST.SP.800-38A]
           Dworkin, M., "Recommendation for block cipher modes of
           operation :: methods and techniques", National Institute
           of Standards and Technology report, DOI 10.6028/nist.sp.
           800-38a, 2001, <https://doi.org/10.6028/nist.sp.800-38a>.

[NIST.SP.800-56Cr1] Barker, E., Chen, L., and R. Davis,
           "Recommendation for key-derivation methods in key-
           establishment schemes", National Institute of Standards
           and Technology report, DOI 10.6028/nist.sp.800-56cr1,
           April 2018, <https://doi.org/10.6028/nist.sp.800-56cr1>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/

RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

12. Informative References

[F3411-19] ASTM International, "Standard Specification for Remote ID and Tracking", February 2020, <http://www.astm.org/cgi-bin/resolver.cgi?F3411>.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

[RFC9153] Card, S., Ed., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements and Terminology", RFC 9153, DOI 10.17487/RFC9153, February 2022, <https://www.rfc-editor.org/info/rfc9153>.

Appendix A. Feistel Scheme

This approach is already being used in format-preserving encryption.

According to the theory, to provide CCA security guarantees (CCA = Chosen Ciphertext Attacks) for m-bit encryption X |-> Y, we should choose d >= 6. It seems very ineffective that when shortening the block length, we have to use 6 times more block encryptions. On the other hand, we preserve both the block cipher interface and security guarantees in a simple way.

```
How to encrypt an m-bit plaintext X using an n-bit block cipher
    E = {E_K} for n > m?

  Enc(X, K):
    1. Y <- X.
    2. Split Y into 2 equal parts: Y = Y1 || Y2
    (let us assume for simplicity that m is even).
    3. For i = 1, 2, ..., d do:
       Y <- Y2 || (Y1 ^ first_m/2_bits(E_K(Y2 || Ci)),
    where Ci is a (n - m/2)-bit round constant.
    4. Y <- Y2 || Y1.
    5. Return Y.

  Dec(Y, K):
    1. X <- Y.
    2. Split X into 2 equal parts: X = X1 || X2.
    3. For i = d, ..., 2, 1 do:
       X <- X2 || (X1 ^ first_m/2_bits(E_K(X2 || Ci)).
    4. X <- X2 || X1.
    5. Return X.
```

Acknowledgments

Authors' Addresses

Robert Moskowitz
HTT Consulting
Oak Park, MI 48237
United States of America

Email: rgm@labs.htt-consult.com

Stuart W. Card
AX Enterprize
4947 Commercial Drive
Yorkville, NY 13495
United States of America

Email: stu.card@axenterprize.com

Adam Wiethuechter
AX Enterprize
4947 Commercial Drive

Yorkville, NY 13495
United States of America

Email: adam.wiethuechter@axenterprize.com