Guidelines for Cryptographic Algorithm Agility

<draft-iab-crypto-alg-agility-00.txt>

Russ Housley

IETF 89 - SAAG Session

Document Abstract

Many IETF protocols may use of cryptographic algorithms to provide confidentiality, integrity, or non-repudiation. Communicating peers must support the same cryptographic algorithm or algorithms for these mechanisms to work properly. This memo provides guidelines for ensuring that such a protocol has the ability to migrate from one algorithm to another over time.

Introduction (1)

- For the mechanisms to work properly, communicating peers must support the same cryptographic algorithms.

- Cryptographic algorithms become weaker with time.
  - New cryptanalysis techniques
  - Computing performance improves
  - As a result, there is a reduction in the work factor to break a particular cryptographic algorithm

Introduction (2)

- For the protocol implementer, this means that implementations should be modular to easily accommodate the insertion of new algorithms.

- For the protocol designer, this means
  - one or more algorithm identifier must be carried
  - the set of mandatory-to-implement algorithms will change over time
  - IANA registry of algorithm identifiers

Algorithm Identifiers

- IETF protocols that make use of cryptographic algorithms MUST carry one or more algorithm identifier
  - IKE carries the algorithm identifiers for ESP and AH
  - This division is completely fine
- Two approaches:
  - Carry one identifier for each algorithm
  - Carry one identifier for a suite of algorithms
- Either approach is acceptable
  - Designers are encouraged to pick one of these approaches and use it consistently
- An IANA registry SHOULD be used for algorithm

Mandatory-to-Implement Algorithms (1)

- For interoperability, the protocol SHOULD specify one or more mandatory-to-implement algorithm

- This is not done for protocols that are embedded in other protocols

  – For example, S/MIME and other protocols makes use of CMS, so S/MIME specifies the mandatory-to-implement algorithms, not CMS

Mandatory-to-Implement Algorithms (2)

- The IETF *must be able to* change the mandatory-to-implement algorithms over time
  - It is highly desirable to make this change without updating the base protocol specification
  - Therefore the base protocol specification SHOULD reference a companion algorithms document, allowing the update of one document without necessarily requiring an update to the other
  - This division also facilitates the advancement of the base protocol specification on the maturity ladder even if the algorithm document changes frequently

Mandatory-to-Implement Algorithms (3)

- Some cryptographic algorithms are inherently tied to a specific key size, but others allows many different key sizes

- When more than one key size is available, the algorithm specification MUST identify the specific sizes that are to be supported
  - Guidance on cryptographic key size for public keys can be found in BCP 86
  - Symmetric keys used for protection of long-term values SHOULD be at least 128 bits

Transition from Weak Algorithms (1)

- It is straightforward to specify an alternative algorithm
- When the alternative algorithm is widely deployed, then the weak algorithm should no longer be used
- Knowledge about the implementation and deployment of the alternative algorithm is imperfect, so one cannot be *completely* assured of interoperability with alternative algorithm

Transition from Weak Algorithms (2)

- In the worst case, the algorithm may be found to be tragically flawed, permitting a casual attacker to download a simple script to break it
  - This has happen when a secure algorithm is used incorrectly or used with poor key management
  - In such situations, the protection offered by the algorithm is severely compromised, perhaps to the point that one wants to refuse to use the weak algorithm well before the alternative algorithm is widely deployed

Transition from Weak Algorithms (3)

- At some point, one refuses to use the weak algorithm
  - This can happen on a flag day, or each installation can select a date on their own

Balance Security Strength

- When selecting a suite of cryptographic algorithms, the strength of each algorithm MUST be considered
- Example from CMS:
  - A previously distributed symmetric key-encryption key can be used to encrypt a content-encryption key, which is in turn used to encrypt the content
  - The key-encryption and content-encryption algorithms are often different
  - Consider:
    - A message content is encrypted with 168-bit Triple-DES key
    - The Triple-DES content-encryption key is wrapped with a 40-bit RC2 key
    - At most 40 bits of protection is provided
    - A trivial search to determine the value of the 40-bit RC2 key will recover Triple-DES key, and then the recovered Triple-DES key can be used to decrypt the content
    - In this situation, the algorithm and key size selections should ensure that the key encryption is at least as strong as the content encryption

Algorithm Agility Considerations

- Some attempts at algorithm agility have not been completely successful
- This document attempts to provide some of the insights based on protocol designs and deployments

Algorithm Identifier Considerations (1)

- The inclusion of an algorithm identifier is a minimal step toward cryptographic algorithm agility
  - If a protocol does not carry an algorithm identifier, then the protocol version number or some other major change is needed to transition from one algorithm to another
- In addition, an IANA registry is needed to pair the identifier with an algorithm specification

Algorithm Identifier Considerations (2)

- Sometimes application layer protocols can make use of transport layer security protocols, such as TLS or DTLS
- This insulates the application layer protocol from the cryptography altogether, but it may still necessary to handle the transition to from unprotected to protected use of the the application layer protocol

Migration Mechanism Considerations

- Protocols need mechanisms to migrate from one algorithm to another over time
  - Eventually any algorithm will become weak
  - A flaw found in the algorithm could greatly shortens its expected life
  - All algorithms age, and the advances in computing power available to the attacker will eventually make them obsolete
- Extra care is needed when one algorithm is used to provide integrity protection for the negotiation of other algorithms
  - A flaw in the negotiation-protection algorithm may allow an attacker to influence the algorithm choices

Key Management Considerations (1)

- Traditionally, protocol designers have avoided a more than one approach to key management because it makes the security analysis of the overall protocol more difficult
- With the increasing deployment of frameworks such as EAP and GSSAPI, the key management is very flexible, often hiding many of the details from the application
- As a result, more and more protocols support multiple key management approaches
- In fact, the key management approach may be negotiable, which creates a design challenge to protect the negotiation of the key management approach before it is used to produce

Key Management Considerations (2)

- Protocols can negotiate a key management approach, derive an initial cryptographic key, and then authenticate the negotiation
  - If the authentication fails, the only recourse is to start the negotiation over from the beginning

- Some environments will restriction the key management approaches by policy
  - Tends to improve interoperability within a particular environment
  - Problems for individuals that need to work in multiple incompatible environments

Next Steps

- Intended Status: BCP
- Security ADs are willing to sponsor this IAB document

- ***Please review and comment!***