More About
- Bulletins
- Challenges
- Crypto FAQ
- CryptoBytes
- RSA Algorithm
- PKCS
- Advanced Encryption Standard
- Tech Notes
- Staff & Associates
- Standards

RSA Security Response to Weaknesses in Key Scheduling Algorithm of RC4

Ron Rivest

Recently, Scott Fluhrer, Itsik Mantin and Adi Shamir published a report [FMS01] that describes several weaknesses in the key scheduling algorithm of RC4 and proposes attacks for exploiting those weaknesses.

In Appendix A to their report, they describe how the WEP (Wired Equivalent Privacy) protocol, intended to provide confidentiality on 802.11 wireless networks, is vulnerable to their attacks. Based on this work, Stubblefield, Ioannidis and Rubin [SIR01] actually implement one of the attacks and demonstate that WEP is very vulnerable "in practice" and not just "in theory". WEP is one of many protocols based on the RC4 algorithm. The attacks are specific to protocols like WEP. As noted below, other RC4-based applications are not necessarily affected.

What are the implications of these developments for those who are either currently using RC4 or considering RC4 for a new application?

(1) Those who are using the RC4-based WEP or WEP2 protocols to provide confidentiality of their 802.11 communications should consider these protocols to be "broken", and to plan remedial actions as necessary to mitigate the attendant risks. Actions to be considered should include using encryption at higher protocol layers and upgrading to improved 802.11 standards when these become available.

In protocols such as WEP, it is often necessary to generate different RC4 keys from different messages (or packets) from a common base key. A method frequently suggested to obtain the keys is to add or concatenate a counter to the base key. The key-scheduling algorithm of RC4 has been widely recognized to be rather lightweight for this purpose, particularly when the initial few bytes of plaintext are easily predictable.

RSA Security has discouraged such key derivation methods, recommending instead that users consider strengthening the key scheduling algorithm by pre-processing the base key and any counter or initialization vector by passing them through a hash function such as MD5. Alternatively, weaknesses in the key scheduling algorithm can be prevented by discarding the first 256 output bytes of the pseudo-random generator before beginning encryption. Either or both of these techniques suffice to defeat the new attacks on WEP and WEP2.

(2) RC4 is most commonly used to protect Internet traffic using the SSL (Secure Sockets Layer) protocol. Indeed, this use of RC4 may make RC4 the most widely-used stream cipher in the world.

There are two reasons why the new attacks do not apply to RC4-based SSL. First, SSL generates the encryption keys it uses for RC4 by hashing (using both MD5 and SHA1), so that different sessions have unrelated keys. Second, SSL does not re-key RC4 for each packet, but uses the RC4 algorithm state from the end of one packet to begin encryption with the next packet. The recent techniques of Fluhrer, Mantis and Shamir thus do not apply to SSL.

As can be seen from these two examples, the applicability of the new attacks to existing applications utilizing RC4-based encryption schemes depends on the details. Applications which pre-process the encryption key and IV by using hashing and/or which discard the first 256 bytes of pseudo-random output should be considered secure from the proposed attacks.

(3) Looking ahead to future applications, the following points seem relevant:

- The "heart" of RC4 is its exceptionally simple and extremely efficient pseudo-random generator. The recent attacks relate only to the key-scheduling algorithm, not to the generator. There are at present no known practical attacks against this generator when initialized with a randomly-chosen initial state.

  For this reason, RC4 is likely to remain the algorithm of choice for many applications and embedded systems.

  (Of course, strong block ciphers like AES or RC6 should also routinely be considered as candidates for any new application, particularly when authentication is also required, since block ciphers can utilize modes of operation, such as Rogaway's OCB mode [R00], that efficiently provide both confidentiality and integrity.)

- The initial key scheduling component of RC4 should for now be routinely amended for new applications to include hashing and/or discarding the first 256 bytes of pseudo-random output. (This has in any case been RSA's routine recommendation.)

- There are clearly many possible approaches for improved RC4 key-generation; further study will certainly produce some that are simpler than hashing and/or discarding output, yet still secure.

In Summary
Current applications should be reviewed to determine whether or not they follow recommended practice for key generation, and designers of RC4-based applications should not be concerned about the new attacks, as long as they follow recommended key-generation practice.

References.
[FMS01] Fluhrer, Scott, Itsik Mantin, and Adi Shamir.
Weaknesses in the Key Scheduling Algorithm of RC4.
(undated, published in August 2001), 23pp.
To be presented at the Eighth Annual Workshop on Selected Areas in Cryptography (August 2001).

[R00] Rogaway, Phil.
Comments to NIST concerning AES Modes of Operation: OCB Mode: Parallelizable Authenticated Encryption.
(Preliminary draft, October 16, 2000), 7pp.
Available at: http://www.cs.ucdavis.edu/~rogaway/papers/ocb.pdf.

[SIR01] Stubblefield, Adam, John Ioannidis, and Aviel D. Rubin.
Using the Fluhrer, Mantin, and Shamir Attack to Break WEP.
AT&T Labs Technical Report TD-4ZCPZZ (August 6, 2001), 8pp.

United States: 1-877-RSA-4900 or 781 515 5000, Europe, Middle East, Africa: +44 (0)1344 781000,
Asia/Pacific: +65 733 5400, Japan: +81 3 5222 5200
Home | Contact Us | Search | Terms of Use and Privacy Statement