More About

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

Raising the Standard for RSA Signatures: RSA-PSS

Burt Kaliski, RSA Laboratories
February 26, 2003

Executive Summary

RSA-PSS is a new signature scheme that is based on the RSA cryptosystem and provides incre assurance. It was added in version 2.1 of PKCS #1.

While the traditional and widely deployed PKCS #1 v1.5 signature scheme is still appropriate to Laboratories encourages a gradual transition to RSA-PSS as new applications are developed.

The "PSS" refers to the original Probabilistic Signature Scheme by Mihir Bellare and Phillip Rog which RSA-PSS is based. Bellare and Rogaway's work raised the bar in the research communit secure signature schemes based on the RSA cryptosystem. RSA-PSS is the adaptation of their industry standards.

RSA-PSS has recently been added to RSA Security's RSA BSAFE Crypto-C and Crypto-J toolk

How RSA-PSS Works

RSA-PSS, like most digital signature schemes, follows the “hash-then-sign” paradigm. Let *M* be be signed. A signature is computed on the message *M* in three steps:

1. Apply a one-way hash function to the message *M* to produce a hash value *mHash*.
2. Transform the hash value *mHash* into an encoded message *EM*.
3. Apply a signature primitive to the encoded message *EM* using the private key to produ signature *S*.

This can be expressed in equation form as

$$S = \text{SigPrim}(\text{private key}, \text{Transform}(\text{Hash}(M)))$$

Here, SigPrim denotes the signature primitive. With the RSA cryptosystem, this is the classic fo

$$S = EM^d \bmod n$$

where (n, d) is the private key, and *EM* and *S* are considered as integers.

Assuming that the encoded message can be recovered from the signature, which is the case fo described here, the signature is verified in three steps:

1. Apply a one-way hash function to the message to produce a hash value *mHash*.
2. Apply a verification primitive to the signature *S* to recover the encoded message *EM*.
3. Determine whether the encoded message *EM* is a valid transform of the hash value *m* (If there is only one valid transform of each hash value, then one can just transform *mHa* again and compare to *EM*; but if more than one, further processing is needed.)

In the PKCS #1 v1.5 signature scheme, the Transform operation consists of fixed padding; the h simply prepended with a header string of the form 00 01 ff ff … ff ff 00 (in hexadecimal) followed that identifies the hash function. In RSA-PSS, the operation is much more “random.” Instead of the scheme generates a random “salt” value then applies a hash function and a mask generatio

the salt and the hash value to produce the encoded message. The transformation, illustrated in
consists of the following steps:

1. Generate a random salt value *salt*.
2. Concatenate fixed padding, the hash value *mHash*, and *salt* to form a string *M'*.
3. Apply the hash function to the string *M'* to compute a hash value *H*.
4. Concatenate fixed padding and the salt value to form a data block *DB*.
5. Apply the mask generation function to the string *M'* to compute a mask value *dbMask*.
6. Exclusive-or the mask value *dbMask* with the data block *DB* to compute a string *mask*
7. Concatenate *maskedDB*, the hash value *H*, and fixed padding to compute the encode
message *EM*.

To determine whether an encoded message *EM* is a valid transformation of a given hash value
simply reverses steps 7 to 4 to recover the salt value and original hash value *H*, then repeats st
see if the hash value is correct.

Because of the hash function and the mask generation function, the encoded message *EM* is al
"random" as there is no almost special structure that distinguishes it from a random string of the
assuming that the two functions are considered as "black boxes" (aka "random oracles"). The or
random parts are the fixed padding bc at the end (introduced for compatibility with other standar
and potentially a few leading 0 bits at the beginning (when *EM* is considered as an integer modu
addition, the transformation is *randomized* because of the random salt value: there are many pc
encoded messages and hence many possible signatures for a given message *M*. This helps wit
analysis as described next.

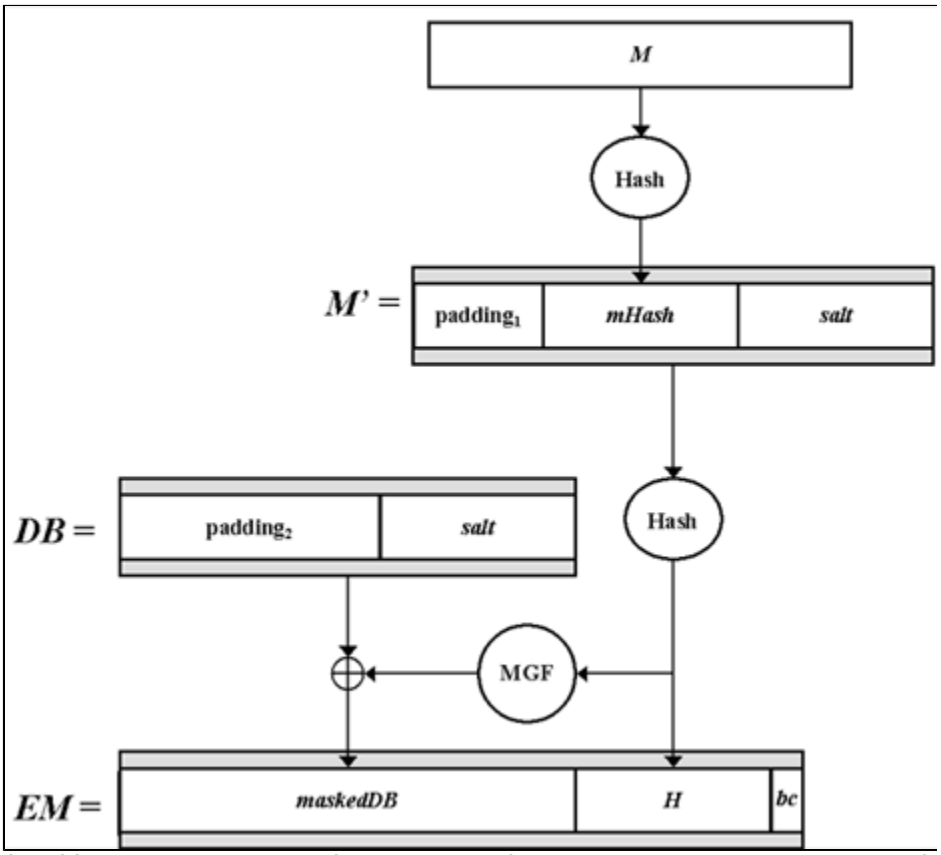

Figure 1: EMSA-PSS encoding operation. Verification operation follows reverse steps to recover *salt*, then fo
recompute and compare *H*. (Source: PKCS #1 v2.1: RSA Cryptography Standard, June 2002.)

Advantages of RSA-PSS

The primary advantage of RSA-PSS over the traditional PKCS #1 v1.5 signature scheme is that
methods of security analysis can relate its security directly to that of the RSA problem. While no

known on the traditional scheme, and while solving the underlying RSA problem (e.g., factoring is the best method known for forging a signature, the connection of PKCS #1 v1.5 signatures to problem has never been proved. RSA-PSS, in contrast, has such a proof if one models its hash "random oracles" as is commonly done.

In recent years there has been a trend toward so-called "provably secure" cryptographic technic more directly connected to underlying hard problems. If a signature scheme does not have a se is theoretically possible that signatures could be easy to forge, yet the underlying problem still b solve. Ideally, one would like some assurance that the problems take about the same amount o Although the state of complexity theory does not let us prove that an underlying problem, e.g., F definitely hard to solve, we will still have the assurance that if the problem is indeed hard to solv are just as hard to forge.

RSA-PSS offers the long-term benefit of higher assurance by narrowing the gap between the wi assumption that the RSA problem is hard to solve, and the claim that signatures are hard to forg RSA-PSS has one of the smallest such gaps among current "provably secure" techniques; in th parlance, the proof of security for RSA-PSS is very "tight." The randomization in the signature s an important role in achieving tightness and is one of the key contributions from Bellare and Rog scheme.

Another advantage of RSA-PSS is that, due to the randomization, an attacker does not know in the encoded message EM will be. This makes "fault analysis" attacks of the sort proposed by B years ago more difficult to mount (see the RSA Laboratories' Bulletin No. 5).

Standards Work

RSA-PSS was added in version 2.1 of PKCS #1, which was published by RSA Laboratories in J The document was recently republished as IETF RFC 3447.

The signature scheme has been recommended by the European NESSIE project, and has also positive evaluations by from Japan's CRYPTREC project. RSA-PSS is also in the (nearly final) d amendment IEEE P1363a. A companion scheme that also provides "message recovery" is inclu international standard ISO/IEC 9796-2:2002.

RSA Laboratories is encouraging a gradual transition to RSA-PSS as standards bodies upgrade techniques in other areas, such as SHA-256 and AES. As one example of the ongoing work, a p specifying RSA-PSS within IETF PKIX certificates has been published as an Internet-Draft. RSA welcomes suggestions for other venues in which to promote the new scheme.

Further Reading

- Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign w Rabin. In U. Maurer, editor, *Advances in Cryptology - EUROCRYPT 96*, volume 1070 of *Notes in Computer Science*, Springer-Verlag, 1996, pages 399-416. Full version availabl http://www.cs.ucdavis.edu/~rogaway/.

- Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002*, vol. 2332 of *Lecture Not Computer Science*, Springer, 2002, pages 272-287. Available via http://www.gemplus.com/smart/r_d/publi_crypto/.

- Jakob Jonsson. Security proofs for the RSA-PSS signature scheme and its variants. Pres Second NESSIE Workshop, September 12-13, 2001. Full version available as IACR ePri http://eprint.iacr.org/2001/053/.

- B.S. Kaliski Jr. RSA digital signatures. *Dr. Dobb's Journal*, May 2001. Available at http://www.ddj.com/articles/2001/0105/.

- NESSIE Security Report. Version 1.0, October 21, 2002. Available at https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D20-v1.pdf.

United States: 1-877-RSA-4900 or 781 515 5000, Europe, Middle East, Africa: +44 (0)1344 781000,
Asia/Pacific: + 61 2 9463 8400, Japan: +81 3 5222 5200

© Copyright 2003 RSA Security Inc. - all rights reserved.
Reproduction of this Web Site, in whole or in part, in any form or medium
without express written permission from RSA Security is prohibited.
RSA, Keon, SecurID, ClearTrust and BSAFE are either registered trademarks
or trademarks of RSA Security Inc. in the United States and/or other countries.
All other products and services mentioned are trademarks of their respective companies.