

What does RPM mean?

RPM™ is the short name for Real Privacy Management™.

What is RPM?

RPM is a secret-key-based security method.

What does RPM do?

RPM provides continuous, mutual authentication for secure electronic communications.

What does “continuous” mean?

When we say “continuous” we mean that RPM authenticates both parties in a communication and encrypts the communication using AES for every transaction in the communications stream. As an example, using RPM is an extension to SSL in our reference implementation of SSLX® we authenticate and encrypt every page transmission between the Web browser and the Web server.

How does RPM encrypt messages?

RPM in and of itself does not encrypt messages. RPM creates the key to be used by a cipher to encrypt the message. This key is called the message key. RPM creates a unique message key for each message and then calls a cipher to encrypt the message using the message key. RPM is “cipher agnostic”, meaning we work with any standard cipher. The most popular cipher is AES.

How is RPM different from other security techniques?

RPM is faster, smaller, more efficient, and more flexible than other security techniques. RPM has been tested by an independent company to be up to 100 times faster than the Diffie-Hellman algorithm. The latest version of RPM is up to 300 times faster. This performance allows RPM to be used where other techniques fail, such as time constrained applications like RF protocols that require fast connection speeds. RPM is fast enough to mutually authenticate and encrypt *every packet* in protocols like TCP, UDP, SIP, IP, etc.

The RPM authentication process is done in 5 microseconds. The speed of RPM is what provides the ability for continuous authentication. RPM is able to authenticate both parties (mutual authentication) for every transaction of any application due to the speed at which the authentication is performed.

RPM is also much smaller than other security methods. Since the RPM libraries are only 10KB in size they can easily be integrated with applications requiring security, without taking up too much space. This size also allows RPM to be placed in embedded systems like RFID, Systems on Chip, Integrated Circuits, ASICs (Application Specific Integrated Circuits) and FPGA (Field Programmable Logic Arrays).

Performing the RPM method has very little overhead too. Measurements have shown that RPM performs at 2 – 5% overhead with TCP/IP, which is within the typical “noise level” of TCP/IP network transmissions. So using RPM does not add additional overhead for the continuous authentication and security that RPM provides.

RPM is also more flexible in its trust model. Applications have different trust models based on the business logic and information processing that the application is representing. RPM offers the flexibility to add authentication and security to any network and communications topology. Most other security methods require third parties to resolve trust using a “root” trust authority, which requires the trust model to be a “tree”. RPM gives the business the flexibility in implementing trust according to their own business model, without requiring third parties.

How does RPM work?

RPM uses a patented algorithm that is based on a mathematical theorem of algebra known as an “underdetermined system of equations.” RPM takes a secret key and using this math generates a new secret key

for authenticating every transmission and a second key that is used as a “message key” with a cipher to encrypt the message. The RPM algorithm is described in more detail in the *RPM Reference Guide*.

Why is RPM so fast and small?

The RPM algorithm only uses simple mathematics known as modular arithmetic. This allows RPM to be implemented using a basic CPU of any computing device. By comparison PKI uses very complex math that requires math coprocessors. Since RPM only uses the basic register-based mathematics of a CPU it is very fast and efficient.

Where does RPM apply?

The short answer is anywhere that electronic communications need security. RPM can be used for various applications and products including:

- Internet – SSL has been extended with RPM in SLLX for continuous mutual authentication and encryption of Web traffic
- VPN – VPN product using SSLX, TCP or IP protocols
- IPsec VPN – using RPM with IPsec to authenticate and encrypt all transmissions in a VPN, not just the session.
- Mobile – securing communications to mobile devices using RPM
- Unified Communications – securing IP telephony and PBX systems for the enterprise
- Information Security – securing enterprise content as it is created, accessed and distributed; data in transit and data at rest

What are we selling when we sell RPM?

We sell the RPM SDKs (Software Developer Kits). The RPM SDKs are the programming libraries that are used to integrate RPM into applications to provide security based on the features of RPM.

How does the customer use RPM?

We will work with customers to help them design RPM into their applications to meet their security requirements. We will assist them in programming the security features into their applications.

How is RPM different than other security?

Most commercial security systems use some variation of public and private keys, most commonly known as PKI (Public Key Infrastructure). The most familiar PKI systems are SSL and IPsec. RPM uses its own patented, secret-key algorithm.

Why not stay with PKI?

PKI is simply not up to the challenge. Today RPM is faster than PKI. If PKI were using key sizes as recommended by security experts PKI would take up to 100 times more time to secure communications, this is why it is not practical to upgrade. Since RPM already meets the secure key size, RPM far surpasses what PKI can do to secure transactions. RPM can do real-time security, with PKI you can watch the paint dry. (See the last section of this FAQ for details.)

Can PKI secure every transaction too?

Yes, according to PKI specifications and theory PKI can also perform mutual authentication and encryption on every transaction. But, this is not done in practice. Due to its size and performance RPM can be used to secure every transaction in a communication process while PKI algorithms are only fast enough in practice to provide session-based security with limited, one-time authentication.

Does PKI secure every transaction?

PKI rarely if ever is used for continuous mutual authentication. This is because PKI uses very complex mathematics and processes for negotiating handshakes between the parties being authenticated. This takes time and is often too costly in terms of performance, overhead and user experience. Most implementations of large PKI have to use hardware accelerators to even make the limited use of PKI acceptable to the user experience. So, even

though PKI can from a theoretical perspective do what RPM does, PKI is not used in this way due to performance constraints.

Does RPM replace PKI?

It is not the intent of RPM to simply replace PKI. RPM can be used in place of PKI, but in most instances RPM will be used to augment the capability of PKI. As an example PKI may be used to establish the session level keys. Once this is done then RPM can use the session level keys to generate transaction level keys for securing the individual transactions within the session.

What is the recommendations for security today?

Currently NIST (National Institute of Standards and Technology) is recommending stronger security key sizes for protecting information. The following Table 1 represents the recommended symmetric key (secret key like used in RPM) size for protecting information.

Table 1: NIST Cryptographic Strength Recommendations

Cryptographic Strength	Protection period	Strength (bits)
	Up to 31 Dec 2010	80
	Up to 31 Dec 2030	112
	Beyond 31 Dec 2030	128

In order for public-key systems to meet these requirements, here is the size keys that they must use:

Table 2: NIST Required Key Sizes

Strength (bits)	ECC Size (bits)	RSA Size (bits)
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

To perform at the 128 bit strength ECC will require around 8 times more processing power than the current 80 bit strength ECC system. The RSA algorithm will require 108 times more processing power for 128 bit strength. The following table represents the processing times required by ECC and RSA to do what RPM does today (authentication and encryption key processing)¹.

Table 3: Key Processing Times

Bit Strength	ECC bit size	ECC Time	RSA bit size	RSA Time
80	160	1	1024	1
112	224	5.4x	2048	24x
128	256	8.2x	3072	108x
192	384	27.6x	7680	3,586x
256	512	65.4x	15360	54,000x

It is very important to note that current public-key systems have a bit strength today of 80 bits and RPM performs at 100 to 300 times faster than today’s public-key systems using the NIST recommended strength of 128 bits.

¹ Times from the report: *Elliptic Curve PKI An exploration of the benefits and challenges of a PKI based on elliptic curve cryptography February 2008*, Entrust