RSA vs ECC Comparison for Embedded Systems

White Paper

Kerry Maletsky, Senior Product Line Director, Security ICs

Modern cryptographic protocols increasingly use asymmetric algorithms such as RSA and ECC because of their flexibility and enhanced ability to manage keys. RSA, which was developed in the late ‘70s has become the algorithm of choice for internet security. Elliptic Curve Cryptography (ECC), which was first proposed in the ‘80s is becoming more widely used for a number of reasons. There are important differences between the two which warrants a careful comparison.

Security Matters

The level of security in systems is becoming a primary concern as you would expect. Most cryptographic experts recommend that current systems offer at least 128 bits of security, but what does that really mean? Note that this is not the same thing as key length as many may think. Security comes from the *combination* of the specific algorithm *and* its key length. For example, it is generally thought that 128 bits of security can be achieved with 128-bit AES keys, 256-bit Elliptic Curve keys, and 3072-bit RSA keys. If implementation issues are ignored, then these algorithms with those specified key lengths will generally have the same level of security. (Please refer to www.keylength.com[1] for recommendations from various sources.) Typical RSA implementations currently employ 1024 or 2048 bit keys, yet both are less secure than AES-128.

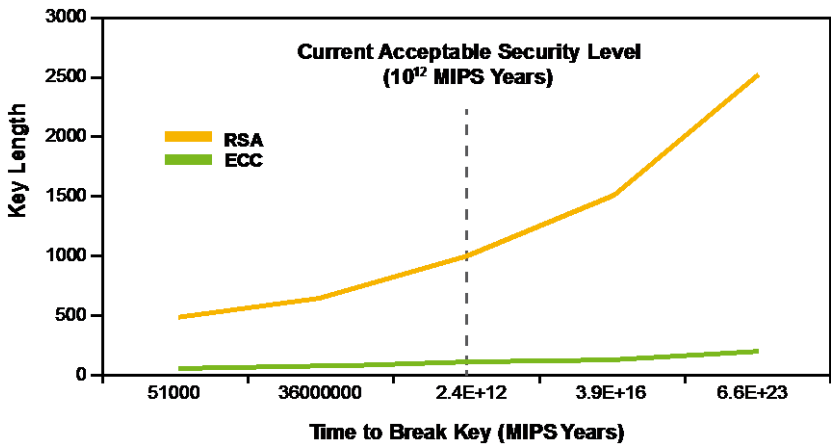
Table 1. Security Comparison for Various Algorithm-key Size Combinations (Source: NSA) [7]

Security Bits	Symmetric Encryption Algorithm	Minimum Size (bits) of Public Keys	
		RSA	ECC
80	Skipjack	1024	160
112	3DES	2048	224
128	AES-128	3072	256
192	AES-192	7680	384
256	AES-256	15360	512

Key lengths generally increase over time as the computation available to attackers continues to increase, which is a manifestation of the cryptographic arms rate. Some experts now suggest that AES-256 be employed for data encryption rather than the prior accepted AES-128 protocol. If you use elliptic curves for the key management (i.e. the encryption/decryption session key) of an AES-256 session, then a 512-bit elliptic curve session key would be required as shown in Table 1 above. To achieve the same level of security with RSA encryption, 15,360 bit keys are required, which is computationally infeasible in embedded systems today. This stark contrast between the feasibility of ECC over RSA for embedded systems indicates that ECC is the algorithm of the future for embedded systems.

Having said that, algorithm security does not actually matter if an attacker can obtain the keys via other methods. This point cannot be emphasized enough. Security starts and ends with how well the keys are protected. In addition to poor key storage, weak or faulty algorithm implementations, bad random number generation, and/or aggressive attacks on end point systems can also degrade security.

Figure 1. RSA and ECC Performance (Source: RSA)[8]



This chart presents another way to look at the performance of RSA and ECC. It compares what key lengths of each algorithm will provide a level of security measured in the time in MIPS-years to break the security. It is clear that ECC is more efficient.

Performance Anxiety

When it comes to performance at 128-bit security levels, RSA is generally reported to be 10-times slower than ECC for private key operations such as signature generation or key management. The performance disparity expands dramatically at 256-bit security levels, where RSA is 50- to 100-times slower. RSA's key generation is also very slow compared to ECC key generation, with the RSA's being 100- to 1000-times slower. However, this may or may not be a significant consideration in systems that generate keys infrequently. It does matter for certain protocols or policies that require more frequent key generation.

Public key signature validation is generally faster with RSA compared to ECC, which can provide a benefit.

Bandwidth

When it comes to network bandwidth, the number one concern relates to the symmetric algorithm used for message encryption and Message Authentication Coding (MAC) for integrity checking (this is unrelated to the choice of RSA versus ECC). Smaller embedded systems may start sessions more frequently, or the asymmetric authentication may be a larger percentage of the overall traffic and the size of the keys and signatures can make a difference. At the 128-bit security level, public keys and signatures are six-times larger for RSA than for ECC. Private keys are 12-times larger for RSA compared to ECC at the 128-bit security level. The key size generally has no impact on performance, but size matters when it comes to the cost of secure storage of the keys.

Government and Industry Standard Recommendations

Based upon the trade-offs noted earlier, there is an almost endless list of new standards that are mandating the use of ECC. A small selection is noted below:

- **Zigbee Networking Standards**

  This standard includes the use of asymmetric algorithms for authentication and key management and specifies ECDSA and ECDH as the algorithm of choice. Refer to:

  "Securing Ad Hoc Embedded Wireless Networks with Public-key Cryptography" [2]

- **Security Module PP Standards**

  The German BSI agency has published a set of standards for energy metering gateway security which specifies elliptic curves as the authentication algorithm. Refer to:

  "Protection Profile for the Security Module of a Smart Meter Gateway (Security Module PP)" [3]

- **CPA Security Characteristic for Smart Metering Standards**

  This UK energy standards for smart metering also specifies elliptic curves. Refer to:

  "Smart Metering – Han Connected Auxiliary Load Control Switch" [4]

- **Intelligent Transportation Systems (ITS) Standards**

  This automotive industry documents the choice of the automotive standards industry for elliptic curves as the algorithm of choice for Vehicle-Vehicle communication security.

  "*Low-Latency ECDSA Signature Verification – A Road Towards Safer Traffic*" [5]

- **Suite B Cryptography Standards**

  The US government publishes a set of standard algorithms approved for use in non-defense applications called Suite B Cryptography. Refer to: "*Suite B Cryptography*" [6]

  Currently, this standard includes only ECC for authentication and key management. RSA has been completely removed. The rationale for this decision is noted at:

  "*The Case for Elliptic Curve Cryptography*" [7]

Conclusion

Due to the security issues, most new cryptographic protocols are moving away from RSA to elliptic curves. That transition is happening even faster in the embedded space where the ECC cost/performance benefits quickly become significant.

References

1. BlueKrypt. "*Cryptographic Key Length Recommendation*," www.keylength.com, 2015.
2. Mitch Blaser. "*Securing Ad Hoc Embedded Wireless Networks with Public-key Cryptography*," www.embedded.com/design/other/4025638/Securing-ad-hoc-embedded-wireless-networks-with-public-key-cryptography, 2006.
3. Bundesamt für Sicherheit in der Informationstechnik. "*Protection Profile for the Security Module of a Smart Meter Gateway (Security Module PP),*" www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/ReportePP/pp0077V2b_pdf.pdf?__blob=publicationFile, Version 1.03, 2014.
4. SECAS and CESG. "*Smart Metering – Han Connected Auxiliary Load Control Switch*," www.cesg.gov.uk/publications/Documents/SmartMeteringHANConnectedALCS.PDF, Version 1.0, 2014.
5. Miroslav Knežević, Ventzislav Nikov, and Peter Rombouts. "*Low-Latency ECDSA Signature Verification – A Road Towards Safer Traffic*," https://eprint.iacr.org/2014/862.pdf, 2014.
6. National Security Agency. "Suite B Cryptography," www.nsa.gov/ia/programs/suiteb_cryptography/, 2014.
7. National Security Agency. "*The Case for Elliptic Curve Cryptography*," www.nsa.gov/business/programs/elliptic_curve.shtml. 2009
8. M. Alimohammadi, and A. A. Pouyan. "Performance Analysis of Cryptography Methods for Secure Message Exchanging in VANET," www.ijser.org/paper/Performance-Analysis-of-Cryptography-Methods-for-Secure.html. 2014.

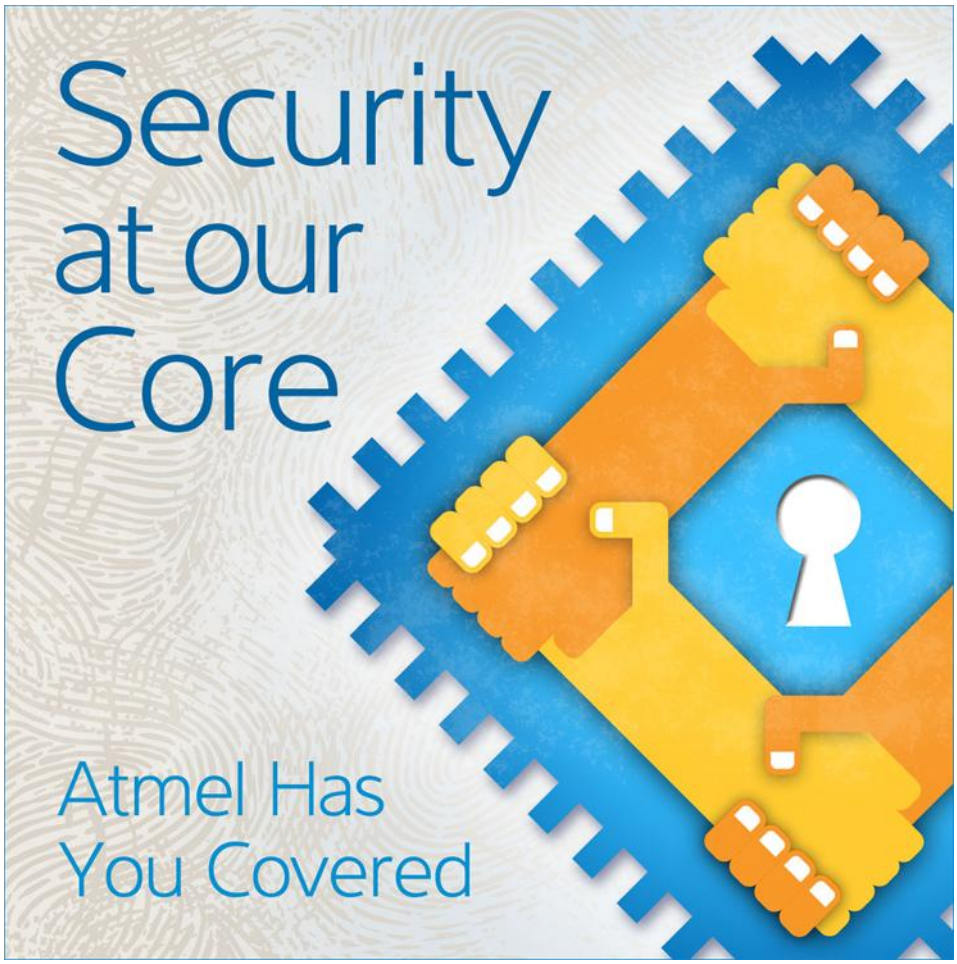
About Atmel Corporation

Atmel Corporation (NASDAQ: ATML) is a worldwide leader in the design and manufacture of microcontrollers, capacitive touch solutions, advanced logic, mixed-signal, security, nonvolatile memory and radio frequency (RF) components. Leveraging one of the industry's broadest intellectual property (IP) technology portfolios, Atmel® provides the electronics industry with complete system solutions focused on industrial, consumer, security, communications, computing and automotive markets.

Today, microcontrollers are just about everywhere, powering an expansive array of digital devices. Many are calling this the era of The Internet of Things, a highly intelligent, connected world where Internet-enabled devices will outnumber people. Atmel is pleased to be at the heart of this movement, developing innovative technologies that fuel machine-to-machine (M2M) communication and the "industrial Internet."

Further information can be obtained from the Atmel website at www.atmel.com.

Author: Kerry Maletsky, Senior Product Line Director, Security ICs
1150 E Cheyenne Mountain Blvd
Colorado Springs, CO 80906
United States
T: (+1)(719) 540-1848
Kerry.Maletsky@atmel.com



Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**