Next generation security for wireless: elliptic curve cryptography

Dr S.A. Vanstone

Founder and EVP Strategic Technology, Certicom

Scott Vanstone, from Certicom, polemicizes for elliptic curve cryptography. He advances his company's view that ECC is the next generation of public-key cryptography for wireless.

Recent cryptanalytic advances have caused increased discussion about public key sizes and the security required. One particular advance was put forward this past February by Dr Adi Shamir, the 'S' in RSA, which raises new concerns about the security of 1024-bit RSA. His paper describes a new hardware implementation for factoring that improves the running time of the Number Field Sieve by three to four orders of magnitude over current implementations. With this hardware, Shamir estimates the factoring for 512-bit RSA can be completed in 10 minutes by a $10 000 device and 1024-bit RSA in less than 1 year with a $10 million device.

Public-key schemes are typically used to transport or exchange keys for symmetric-key ciphers. A well-designed symmetric-key algorithm that uses an m-bit key should provide m bits of security. That is, to find the key being used would require the adversary to search exhaustively through the key space that has 2m keys in it.

We know many organizations understand key strength when looking at symmetric ciphers because they are moving from 3DES to AES; even if they only moved from DES a couple of years ago. At the same time, these same organizations implement the 1024-bit RSA for key transport because they feel that it's good enough. Clearly, the 1024-bit RSA does not match the 128-bit security level now used for symmetric ciphers.

Those who are reluctant to migrate from 1024-bit RSA to the larger keys sizes now required defend these inadequate RSA key sizes by arguing that memory limitations and other costs will protect these keys. This is a very dangerous approach because the security of the public key system must be matched with the symmetric cipher used — it's only common sense.

In fact, the US National Institute of Standards and Technology (NIST) notes in *FIPS 140-2: Security Requirements for Cryptographic Modules* that "Compromising the security of the key establishment method (e.g., compromising the security of the algorithm used for key establishment) shall require at least as many operations as determining the value of the cryptographic key being transported or agreed upon." This means that the modulus lengths of the public key must be enforced for use with AES.

One could argue that it's okay to use 1024-bit RSA to deliver a 128-bit AES key if the desired security level is only 80-bits. On the other hand, if one could deliver that same 128-bit key

Table 1

Public-key system	Mathematical Problem	Examples
Integer factorization	Given a number n, find its prime factors	RSA, Rabin-Williams
Discrete logarithm	Given a prime n, and numbers g and h, find x such that $h = g^x \bmod n$	ElGamal, Diffie-Hellman, DSA
Elliptic curve discrete logarithm	Given an elliptic curve E and points P and Q on E, find x such that Q=xP	EC-Diffie-Hellman, ECDSA

Table 2

Public-key system	Best known methods for solving mathematical problem	Running times
Integer factorization	Number field sieve: exp(1.923 (log n)1/3(log log n)2/3)	Sub-exponential
Discrete logarithm	Number field sieve: exp(1.923 (log n)1/3(log log n)2/3)	Sub-exponential
Elliptic curve discrete logarithm	Pollard-rho algorithm: square root of n	Fully exponential

with a public-key mechanism that has a security level equivalent to 128-bits then why not do it? This article will show you that you don't have to compromise the security of the system to address performance issues with Elliptic Curve Cryptography (ECC).

Families of public-key schemes

Today, there are three types of industry-standard public-key cryptographic systems that can be considered secure, efficient, and commercially viable. These systems, classified according to the mathematical problem on which they are based, are: Integer Factorization systems (of which RSA is the best known example), Discrete Logarithm systems (such as the US Government's DSA), and ECC. The two major benchmarks when comparing these systems are security and efficiency (Figure 1).

The security of the system is directly tied to the relative hardness of the underlying mathematical problems (Figure 2). In each case, there are best known methods for solving these three distinct mathematical problems. Because the best known way to solve the elliptic curve discrete logarithm problem (ECDLP) is fully exponential, you can use substantially smaller key sizes to obtain equivalent strengths. Hence ECC provides the most security per bit of any public key scheme known.

The benefits of ECC

The absence of a sub-exponential time algorithm for the ECDLP means that significantly smaller parameters can be used in ECC than with DSA or RSA. The advantages that can be gained from smaller parameters include speed and smaller keys or certificates.

These advantages are especially important in environments where at least one of the following resources is limited:

- Processing power
- Storage space
- Bandwidth
- Power consumption

The result is that ECC is especially well suited for constrained environments such as smart cards, cellular phones, PDAs, digital post marks and other constrained environments.

Relative public-key sizes used in practice

So what does this mean in practice? NIST has recommended that 128-bit protection is necessary to achieve relatively lasting security (to the year 2036 and beyond). This means moving from 3DES to AES.

To avoid compromising the security of the system, NIST's FIPS 140-2 standard, as mentioned at the outset, indicates keys for symmetric ciphers such as AES must be matched in strength by public key algorithms such as RSA and ECC. For example, a 128-bit AES key demands an RSA key size of 3072 bits for equivalent security but only a 256-bit ECC key.

As you can see in Table 3, while ECC key sizes scale linearly, RSA does not. The result is that the gap between systems grows as the key sizes increase. This is especially relevant to implementations of AES where at 256-bits you need an RSA key size of 15,360 bits compared to 512 bits for ECC.

This will have a significant impact on systems as the relative computational performance advantage of ECC versus RSA is not indicated by the key sizes but by the cube of the key sizes. The difference becomes even more dramatic as the greater increase in RSA key sizes leads to an even greater increase in computational cost. So going from 1024-bit RSA key to 3072-bit RSA key requires about 27 times (3^3) as much computation while ECC would only increase the computational cost by just over four times (1.6^3).

Dr. Scott A. Vanstone

Founder & EVP Strategic Technology

One of the founders of Certicom, Dr. Vanstone is also a Professor of Mathematics and Computer Science at the University of Waterloo. Dr. Vanstone devotes much of his research to the efficient implementation of the elliptic curve cryptography (ECC) for the provision of information security services in handheld computers, smart cards, wireless devices, and integrated circuits. He is the EVP Strategic Technology for Certicom and oversees all research, which has generated 120 patents and patents pending for the company. He also sits on the Board of Directors for Certicom.

Vanstone has published more than 150 research papers and several books on topics such as cryptography, coding theory, finite fields, finite geometry, and combinatorial designs. He is a co-author of the *Handbook of Applied Cryptography*. Dr. Vanstone also holds the NSERC/Pitney Bowes Senior Chair of Cryptography at the University of Waterloo. Recently, he was elected a Fellow of the Royal Society of Canada, Academy of Sciences.

Scott Vanstone has a Ph.D. in mathematics from the University of Waterloo.

Table 3

Security (Bits)	Symmetric encryption algorithm	Minimum Size (Bits) of Public Keys		
		DSA/DH	RSA	ECC
80		1024	1024	160
112	3DES	2048	2048	224
128	AES-128	3072	3072	256
192	AES-192	7680	7680	384
256	AES-256	15360	15360	512

Table 4

Security (Bits)	Symmetric encryption algorithm	Minimum Size (Bits) of Public Keys	
		RSA	ECC
80		1536	160
112	3DES	4096	224
128	AES-128	6000	256
160	AES-192	10000	320

In fact, the New European Schemes for Signatures, Integrity and Encryption (NESSIE), an EU organization formed as a bridge between the research community and the user community, takes a much more conservative approach and recommends, at 128 bits of security, a 6,000-bit RSA, almost double what NIST recommends compared to the 256-bit ECC key (Table 4).

Standards involving ECC

The main goal of security standards is to facilitate the widespread use of cryptographically sound and well-accepted techniques, promote interoperability and help ensure ongoing detailed analysis by cryptographers through clear, complete, and public specification of baseline techniques.

Over the last nine years, a great deal of work has taken place to ensure that ECC meets these goals and is specified in an ever-increasing number of standards. It started with the IEEE P1363 in 1994 (becoming a standard in 2000), and now includes many accredited standards organizations:

- ISO (in ISO 14888-3: ECDSA and other ECC-based signature schemes).
- IEEE (in IEEE 1363-2000 for public-key cryptography).
- The American National Standards Institute (in ANSI X9: cryptography for financial-services industry).

NIST also specifies ECC in FIPS 186-2: Federal Information Processing Standards ECDSA and SP 800-56: Special Publication on Key management. While in Europe, the BSI: Bundesamt für Sicherheit in der Informationstechnik in Germany also specifies ECC.

Currently, there are ongoing efforts within a number of application standards to include ECC as a required or recommended security mechanism. These efforts include:

- IEEE 802.15 Wireless Personal Area Networks
  - 802.15.3: WPAN, ultra wide band
  - 803.15.4: WPAN, low rate
  - ZigBee
- OMA-Open Mobile Alliance
  - Wireless Application Protocol
  - Wireless TLS
- IETF: Internet Engineering Task Force
  - IPSec, TLS, PKIX, S/MIME
- ATM Forum

Conclusions

Ultimately, the benefits of ECC are many: linear scalability, a small software footprint, low hardware implementation costs, low bandwidth requirements, high device performance. For these reasons, it has gained the support of a number of leading companies —Texas Instruments, Motorola and Pitney Bowes to name a few.

ECC becomes particularly attractive when implementing the stronger symmetric encryption offered by AES. These savings are even more advantageous when computational power, bandwidth, or storage space are limited as is the case in the ever-increasing number of mobile devices that are being used everyday. ECC delivers the highest strength-per-bit of any public-key cryptography system known today.

While the performance advantages are impressive, you need to ensure that the security system has been studied extensively in the public forum and also specified by major standards worldwide. Elliptic Curve Cryptography is here today and is without question the next generation of public-key cryptography.