Wireless Security Perspectives

Cellular Networking Perspectives

Technical Editor: Les Owens, Managing Editor: David Crowe Vol. 2, No. 8. September, 2000

Public Key Cryptosystems (PKCS): A Three Part Series

This month's issue of *Wireless Security Perspectives* is the first in a 3-Part series on Public Key Cryptosystems (PKCS) for wireless communications. This first part was written by Jim Semple of Certicom. Editing and figures were provided by Les Owens. It covers Elliptic Curve Cryptography (ECC) and explains why this type of advanced cryptosystem is being considered by the wireless industry for the cryptographic protection of communications. Next month, Boston-based NTRU Cryptosystems will introduce another technique that is also being considered for the wireless industry. In the November issue, we will conclude the 3-Part series with a paper by Toronto-based Karthika Technologies on a very advanced number-theoretic form of PKCS – "Abelian Varieties" . Will this be the Public Key Cryptosystem for the future?

Glossary

For any telecom or security terms you are unfamiliar with, please consult:

www.cnp-wireless.com/glossary.html

PKCS Part I: Is Elliptic Curve Cryptography Ideal for Wireless?

The demand for small wireless devices, enabling mobile commerce, location-based services, and the extension of the corporate intranet, is fueling the deployment of Elliptic Curve Cryptography. This paper, first part in our series, is intended as a brief introduction to ECC – to explain why ECC offers greater efficiencies than other public-key cryptosystems.

Discrete Logarithm Problems – Mathematical Fundamentals

Before discussing ECC for use in wireless, we will provide a brief background on the mathematics on which much many PKCS are based – the discrete logarithm problem, or DLP. This problem is fundamental to the security of many cryptographic protocols, the most famous of which is the Diffie-Hellman Key Exchange. The Diffie-Hellman Key Exchange (DHKE), a method used to securely derive pair-wise cryptographic keys, is illustrated in Figure 1. As shown, the two communicants Alice and Bob, generate random numbers independently, perform a computation, exchange the partial (or public) keys, then perform another computation. The computations that Alice and Bob perform are known as discrete exponentiations – exponentia-

About *Wireless Security Perspectives*

Price

The basic subscription price for *Wireless Security Perspectives* is $250 for one year (12 issues) for delivery by emailed PDF file or by first class mail within the US or Canada. International subscriptions are US$300 per year. The basic subscription allows for distribution to up to 10 people within one organization. Contact us for license fees to allow more readers.

Current subscribers to *Cellular Networking Perspectives* receive a discounted price – only $200 for delivery within the US and Canada or $250 elsewhere.

Back issues are available individually, or in bulk at reduced prices.

Complete pricing information for both publications is available at:

www.cnp-wireless.com/prices.html

To obtain a subscription, please contact us at:

cnpaccts@cnp-wireless.com

Next Issue Due…

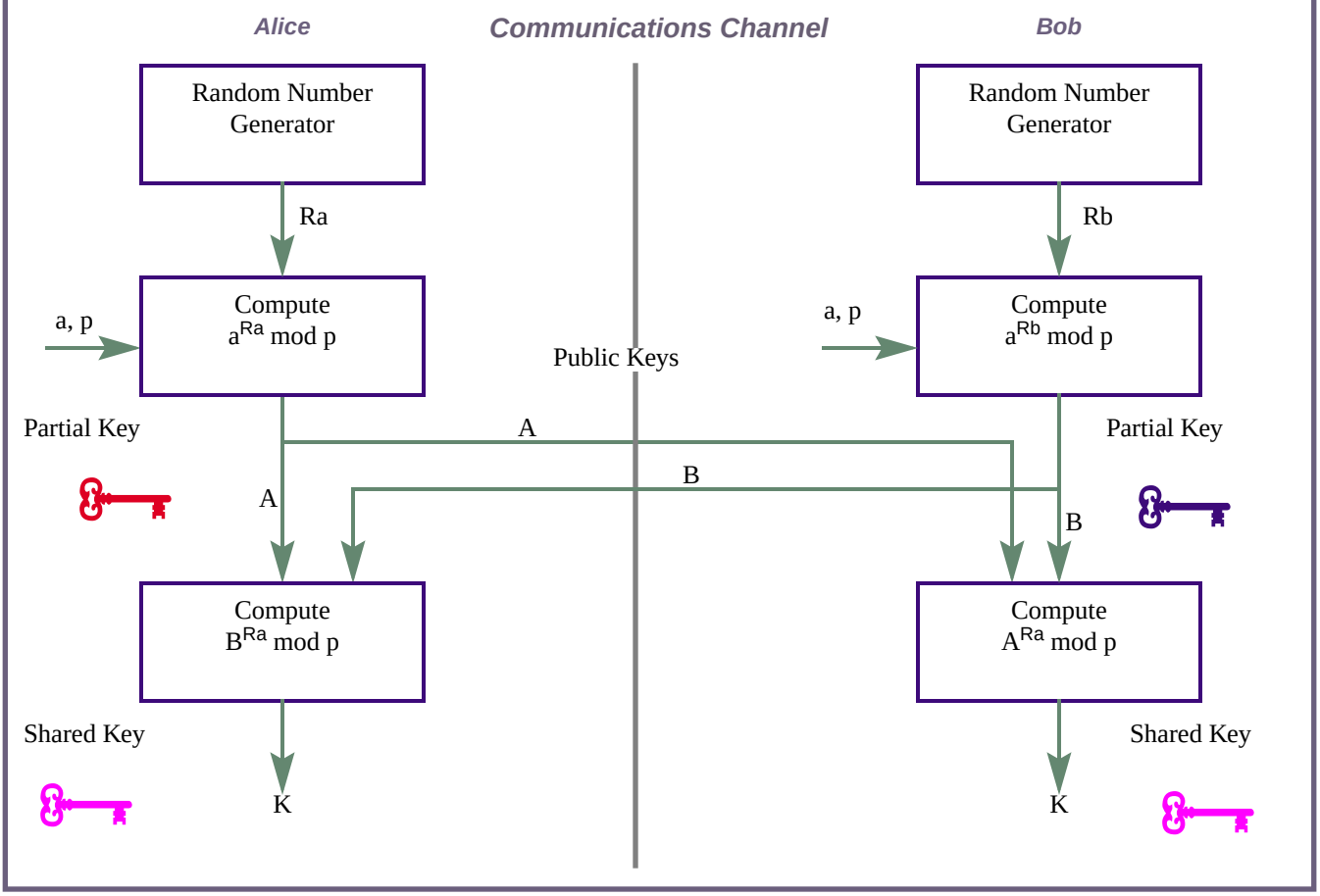
October 16th, 2000.

Future Topics

IP security • Public Keys & Wireless • Kerberos PKINIT• Public Key Infrastructure (PKI) • IKE • Wireless Data Security • Elliptic Curve Cryptography (ECC)• Abelian Varieties • IETF Security Standards

Wireless Security Perspectives (ISSN 1492-806X (print) and 1492-8078 (email)) is published monthly by Cellular Networking Perspectives Ltd., 2636 Toronto Cresc. NW, Calgary, Alberta T2N 3W1, Canada. **Phone:** 1-800-633-5514 (+1-403-274-4749) **Fax:** +1-403-289-6658 **Email:** cnpsales@cnp-wireless.com **Web:** http:/www.cnp-wireless.com/wsp.html. **Subscriptions:** $200 for current *Cellular Networking Perspectives* subscribers for delivery in the USA or Canada, US$250 elsewhere. Non-subscribers pay $250/yr. for delivery in the USA or Canada, US$300/yr. elsewhere. Payment is accepted by cheque, bank transfer, American Express, Diners Club, MasterCard or Visa. **Delivery:** Email or 1st class mail. **Back Issues:** Available individually for $20 in the US and Canada and US$25 elsewhere, or in bulk at reduced rates. **Discounts:** Educational and small business discount: 25% off any order. **Copies:** Each subscriber is licensed to make up to 10 copies of each issue or back issue. Call for rates to allow more copies.

Figure 1: Diffie-Hellman Key Exchange



tion of one number and "modulo reduced" another very large number. The result of this two-stage exponentiation process is a cryptographic key that both Alice and Bob share. No one else knows this shared key despite the exchange and exposure of partial information over the communications channel and what is seemingly a trivial computation.

The simplest instance of this discrete logarithm problem, from our Diffie-Hellman Key Exchange example, is that given a prime number **p** and positive numbers **m** and **n** that are both less than **p**, how easy is it to find a number **r** such that:

$\mathbf{m}^{\mathbf{r}} = \mathbf{n} \bmod \mathbf{p}$ – if such an **r** exists?

In Figure 1, for example, we know **n** (A or B), **m** (a or b), and **p**; the adversary's job is to solve for **r**. The security of the system relies on the complexity of solving this problem. In order to make the computation difficult, the parameters must be chosen carefully. For example, we prefer that the group formed by the powers of **m** to have a large prime number of elements. The terminology is that we wish m to generate a large prime subgroup of **GF(p)** – the group of invertible elements in the field of integers modulo **p**.[1]

Nevertheless, there are very powerful mathematical techniques to attack the discrete logarithm problem in a finite field, such as the *number field sieve* (NFS). As a result the prime **p** must be very large.

We may pose a discrete logarithm problem in other groups and use them as the basis for our security. Fundamentally, we want computations in one direction to be very efficient, such as exponentiating from **m** to $\mathbf{m}^{\mathbf{r}}$, but we require that the discrete logarithm problem be difficult.[2]

For example, we prefer groups for which index calculus techniques do not apply.[3] There are general approaches to the discrete logarithm problem in any group, such as Pollard's "rho algorithm," but this attack is asymptotically much slower than the NFS.[4] Therefore, we aim to use groups where you cannot do significantly better than Pollard rho: for these groups we may use much shorter key-lengths, which may also bring other efficiencies.

The other popular technique for public key cryptography is, of course, the Rivest-Shamir-Adleman system – or RSA.[5] The RSA scheme is depicted in Figure 2. Here, with our two communi-

1. GF(p) signifies a Gallois Field after the French mathematician, Evariste Gallois (1811-1832) whose work laid the foundation of modern group theory.

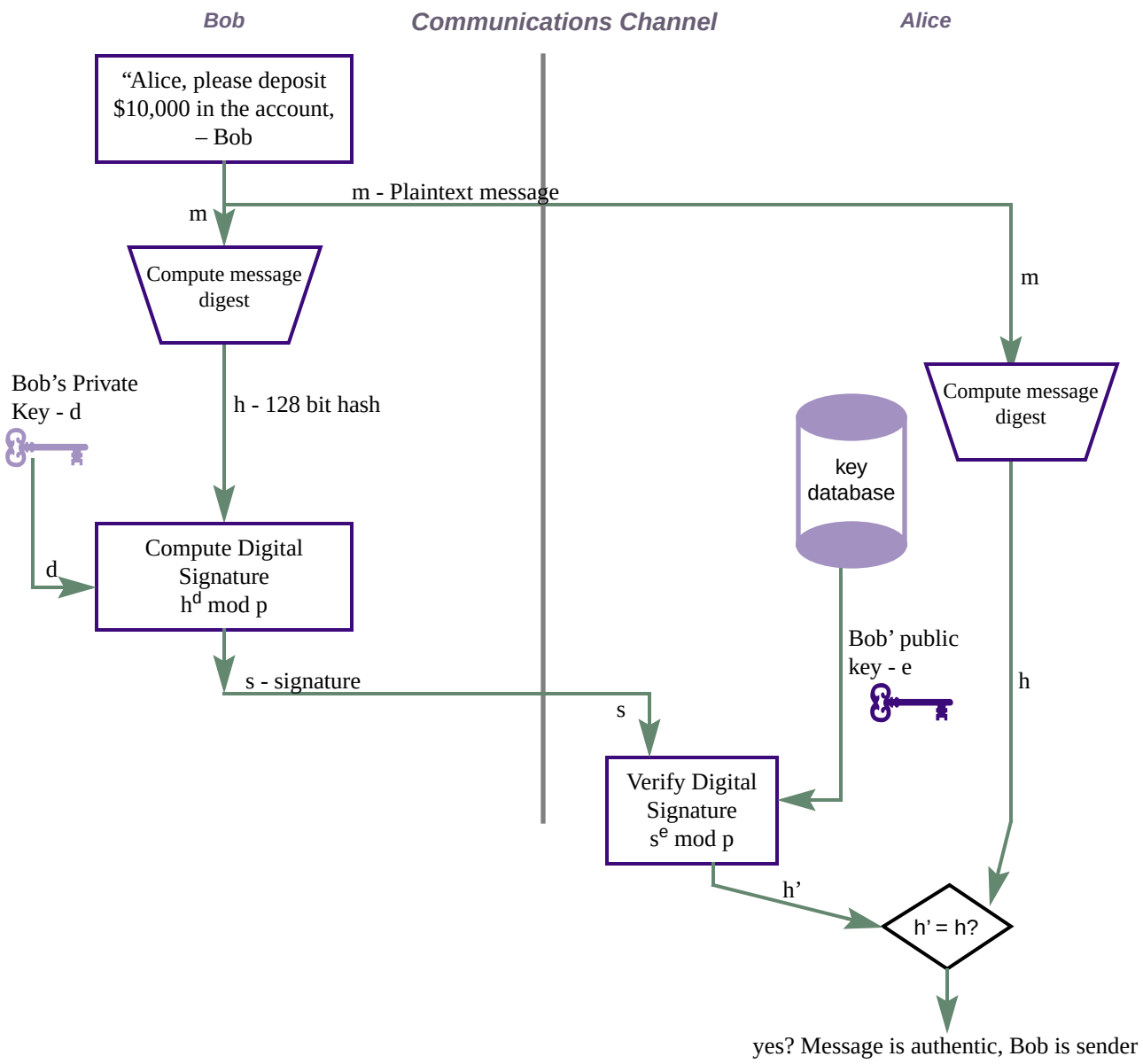
2. In cryptographic terms, this is known as a one-way function – an OWF

3. The index calculus algorithm is the most powerful method for computing discrete logarithms.

4. A storage-efficient algorithm developed by J.M Pollard. See reference 1.

5. Patent (4,405,829), issued 9/20/83, for the RSA PKCS expires this month (September, 2000).

Figure 2: Simplified Example of RSA Digital Signature Process



cants, a message is being digitally signed by Bob and sent to Alice. As shown, Bob first applies his original plaintext message to a message digesting (or hash) algorithm to produce a compressed "fingerprint" of the plaintext. The hash is then applied to the RSA algorithm with Bob's secret key, d. Bob then transmits the plaintext message and the signature to Alice. In order for Alice to verify that the message did not change in transit (message integrity) and that it did, in fact, originate from Bob, she performs the same hashing of the message and compares the real-time hash with the decrypted value of the signature (using Bob's public key). If the two hashes are equal, then Alice can be sure that the message is authentic, or unaltered, and that it came from Bob.

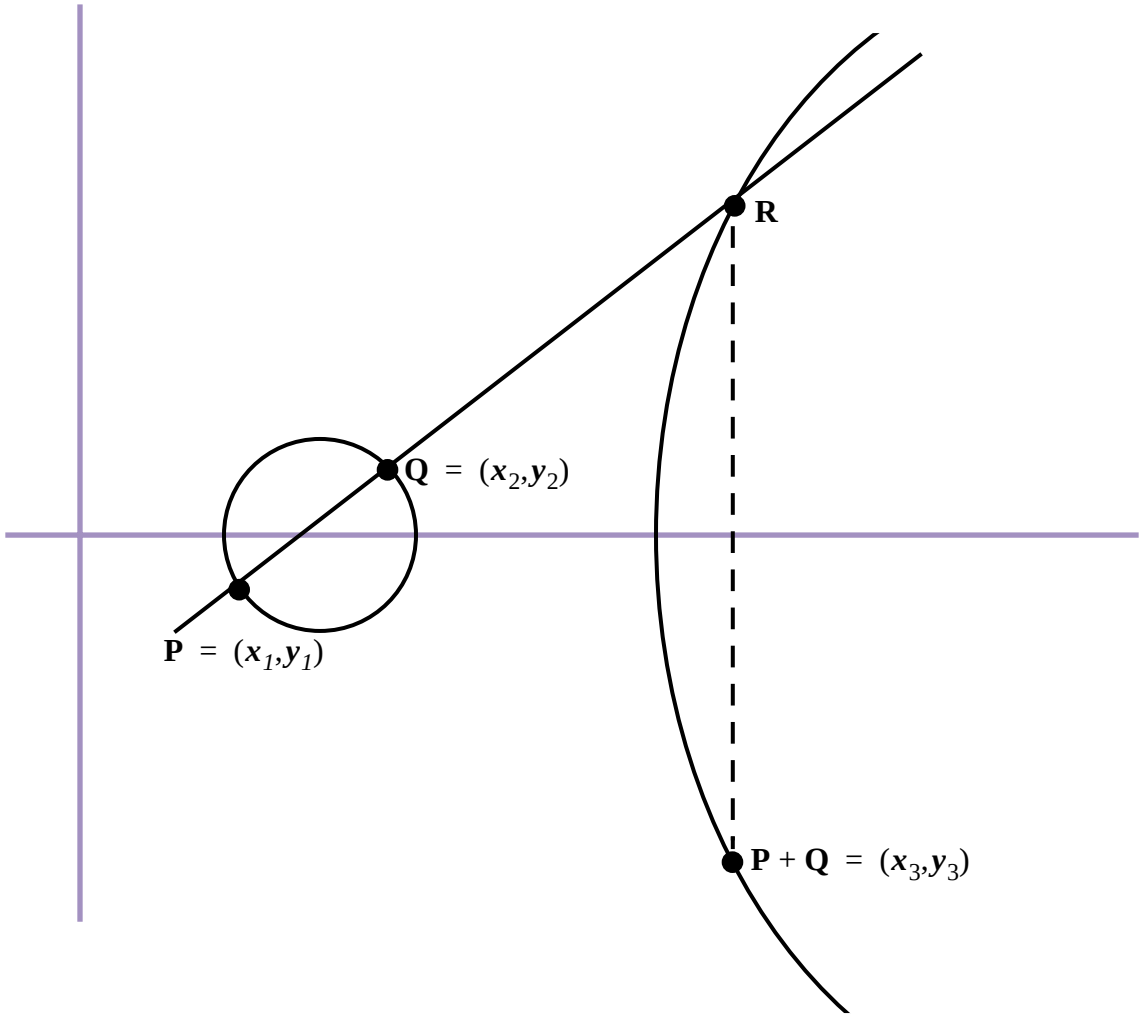
In the RSA PKCS, as in the example, security is based on the difficulty of factoring a large number into its prime factors. Fast factoring techniques like the NFS can be used against the RSA scheme as well. Hence, the parameter sizes must again be chosen to be very large to achieve the required level of security.

Elliptic Curves – A Better Mousetrap?

In 1985, Victor Miller and Neal Koblitz proposed using the group of points of an elliptic curve as basis for the discrete logarithm problem. Elliptic curves have been studied classically and are at the heart of modern number theory. Rather than elements of **GF(p)**, we consider points on a curve: pairs of elements **(x,y)** from **GF(p)** which satisfy some fixed equation such as:

$$\mathbf{y}^2 = \mathbf{x}^3 + \mathbf{ax} + \mathbf{b}$$

Figure 3: Example of Elliptic Curve



This set of points comes with a group structure we require to pose our discrete logarithm problem. Figure 3 shows how we can take any two points on the curve and define their sum: given two points $\mathbf{P} = (\mathbf{x}_1,\mathbf{y}_1)$ and $\mathbf{Q} = (\mathbf{x}_2,\mathbf{y}_2)$ on the curve, we form their sum **P**+**Q** by drawing the line joining **P** and **Q**, finding the third point **R** where that line intersects the curve. Then we reflect **R** onto the x-axis. The result $(\mathbf{x}_3,\mathbf{y}_3)$ is the sum. The discrete logarithm problem in this context then becomes:

> Given points **P** and **Q** on an elliptic curve, find a number **x** such that if we add **P** to itself **x** times we arrive at **Q**.

Again we choose our parameters carefully. For example, we must ensure that the point **P** generates a prime number of points. Despite fifteen years of vigorous research there are no better attacks on this discrete logarithm problem, in general, than versions of Pollard's rho algorithm. As a result, for a given level of security we may use elliptic curve key sizes that are considerably smaller than the corresponding RSA key, as is described in Table 1.

Table 1: Key Size Comparison for Classical and Public Systems

Symmetric key size	ECC key size	RSA key size	Approximate MIPS years to break
80	160	1024	10^{12}
112	224	2048	$5 * 10^{21}$
128	256	3072	$3 * 10^{26}$

A MIPS (Million Instructions Per Second) year represents the total number of instructions that can be performed in one year on a machine capable of performing one MIPS. For example, 50,000 MIPS years means that 50 computers operating one MIPS would take 1,000 years to perform the factorization. 50,000 computers operating at this speed would take only one year. Therefore, the difference in comparable key sizes becomes even more dramatic as the required level of security increases.

Implementation of Elliptic Curve Cryptography

In comparing ECC with RSA, the real efficiencies come from this difference in key lengths for a given level of security; given an RSA modulus **N**, the equivalent elliptic curve group should have a size of about

$$\mathbf{N}^{(1/3)}(\log \mathbf{N})^{(2/3)}$$

ignoring several constants to try to keep this simple!

As a result, as **N** increases we find the times for key generation or performing and verifying digital signatures becomes significantly faster using elliptic curve groups than for the equivalent task using RSA. This is particularly important for constrained devices, such as cellular phones, satellite networks, pagers and PDAs. In those devices computational power (microprocessor speeds), bandwidth (spectrum), and battery resources are limited. Moreover, smaller certificate sizes mean that elliptic curve certificates lend themselves to more efficient delivery via SMS, storage on smart cards, etc.[1] Elliptic curve security is now being adopted by such terminal manufacturers as Motorola, Neopoint, Palm, RIM and Sony. Application service providers (ASP) such as 724 Solutions, AvantGo, and Aether have also embraced the technology.

Table 2 compares the performance of ECC and RSA on a Palm Pilot (a 68000 processor running at 16MHz). The ECC timings were produced using Certicom's cryptographic toolkit for embedded devices, and the RSA timings were produced using Ian Goldberg's port of SSLeay based on the work of Dan Boneh and Neil Daswani of Stanford.

1. A certificate (public key) is a data structure that securely binds an entity to a public key. A certificate is digitally signed by a presumably trustworthy entity known as a certificate authority (CA).

Table 2: Performance of public key algorithms on the Palm Pilot

Algorithm	Key generation	Sign	Verify
163-bit ECC	600 ms	750 ms	2,000 ms
1024-bit RSA	1,260,000 ms	25,970 ms	1,770 ms

As shown, the digital signature process for an RSA implementation takes almost 26 seconds. In contrast, the signature process for ECC requires less than a second. In general, ECC key generation and signing are faster than the equivalent RSA at 1024 bits, while verification too is faster at 2048 bits.

ECC in Standards

Standardization plays an essential role in the security industry. Over the years, countless cryptographic schemes have been proposed, with the majority broken as attackers use increasingly sophisticated techniques. The only way we gain confidence in the security of a particular system is by opening the proposal to years of scrutiny by the world's cryptographic community. This is where standardization comes in. Standardization acts as a seal of approval, indicating that the security of a system has withstood years of intense scrutiny.

Roughly speaking, there are two types of cryptographic standards: core standards, which specify the cryptographic schemes themselves, and application standards, which describe how to use the schemes in a particular application. Core ECC standards providing encryption, key agreement, and signatures have now been adopted by ANSI, IEEE, NIST, and the SECG. These standards have, in turn, been used in a wide variety of application standards, including IPSec for remote corporate intranet access, S/MIME for secure email, and WAP for wireless internet browsing.

To Probe Further

White papers and further examples of applied Elliptic Curve Cryptography may be found at:

www.certicom.com.

The Standards for Efficient Cryptography is a group dedicated to choosing common standards for inter-operability and easy deployment of Elliptic Curve Cryptography:

www.secg.org

Also, a conference dedicated to Elliptic Curve Cryptography is taking place at the University of Essen, Germany, 4-6 October 2000:

www.cacr.math.uwaterloo.ca/conferences/2000/ecc2000/announcement.html

Selected References

[1]. I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, London Mathematical Society Lecture Note Series 265.

[2]. D. Boneh and N. Daswani, *Experimenting with electronic commerce on the Palm Pilot*, www.stanford.edu/~dabo.

[3]. A.W. Knapp, *Elliptic Curves*, Princeton University Press, 1992.

[4]. A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

About the Authors

Jim Semple works in Certicom's London, UK office as a Business Development Manager in the European wireless industry. Jim completed his doctorate in mathematics at Oxford in 1992 and has since been working in cryptography and GSM security. He can be reached at:

jsemple@certicom.com

Les Owens is the technical editor of *Wireless Security Perspectives*. He is also an internationally recognized industry expert in telecommunications security with multiple patents on methods for fraud control, cryptography and network security. He can be reached at:

owensville@worldnet.att.net

Upcoming Security Conferences

The Future Of Information Security
October 10-12, 2000
San Francisco, CA

www.dci.com/events/security

ITTA InfoSec Summit
October 16-17, 2000
International Trade Center
Washington, DC

www.itaa.org/infosec/summit.htm

23rd National Information Systems Security Conference
October 16-19, 2000
Baltimore Convention Center

csrc.nist.gov/nissc/overview.htm

IPv6 2000
Internet Protocol version 6
October 19-20, 2000
Washington, DC

www.xiwt.org/XIWT-IPv6/meetingsite.html