Wireless Security Perspectives

Cellular Networking Perspectives

Technical Editor: Les Owens, Managing Editor: David Crowe Vol. 2, No. 9. October, 2000

CryptoNews: Rijndael Selected for AES

On October 2, 2000, the National Institute of Standards and Technology (NIST) completed the DES (Data Encryption Standard) replacement activity with the announcement of their selection of Rijndael as the Advanced Encryption Standard (AES) algorithm. “Reign Dahl”, “Rain Doll” and “Rhine Dahl” are suggested pronunciation alternatives.

The search for a new encryption standard was initiated by demonstrated weaknesses in DES. Stimulated by a competition sponsored by RSA, specialized machines were developed that were eventually able to recover DES keys in a matter of hours.

The Rijndael cipher has a variable block length and key length. The algorithm currently specifies how to use keys with a length of 128, 192, or 256 bits to encrypt blocks with a length of 128, 192 or 256 bits (all nine combinations of key length and block length are possible). Both block length and key length can be extended very easily to multiples of 32 bits. Rijndael can be implemented very efficiently on a wide range of processors and in hardware.

It was back in January 1997 when NIST announced the initiation of the AES development effort and made a formal call for algorithms on September 12, 1997. The call stipulated that the AES would specify an unclassified, publicly disclosed encryption algorithm(s), available royalty-free, worldwide. On August 20, 1998, NIST announced a group of fifteen AES candidate algorithms at the First AES Candidate Conference (AES1). The list was narrowed down to five finalist: MARS, RC6, Rijndael, Serpent, and Twofish.

In the coming weeks, a draft Federal Information Processing Standard (FIPS) for the AES will be published for public review and comment. Following the comment period (of at least three months), the standard will be revised by NIST, as appropriate, in response to those comments. If all steps of the AES development process proceed as planned, it is anticipated that the standard will be completed by the summer of 2001. The Rijndael cipher, developed by Joan Daemen and Vincent Rijmen, like DES, will be written into a Federal Information Processing Standard (FIPS). DES is currently FIPS-46.

For more information visit:

csrc.ncsl.nist.gov/encryption/aes/round2/aesfact.html

A comprehensive report will be available within a few weeks that addresses the analysis and selection process for the cryptographic algorithms, cryptographic strength of Rijndael, and performance / application issues.

For information on Rijndael, you may also contact Jim Foti at NIST:

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology (NIST)
100 Bureau Drive, Mail Stop 8930

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…

November 15th, 2000.

Future Topics

IP security • Public Keys & Wireless • Kerberos PKINIT• Public Key Infrastructure (PKI) • IKE • Wireless Data Security • Abelian Varieties • IETF Security Standards • AES (Rijndael)

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.

Gaithersburg, MD 20899-8930
USA

Phone: +1-301-975-5237

Fax: +1-301-948-1233

Email: jfoti@nist.gov

Qualcomm’s Greg Rose is planning to provide a presentation, titled, ‘Rijndael Revealed’ at the next Telecommunications Industry Association AHAG (*ad hoc* Authentication Group) meeting:

AHAG Meeting
October 24th-25th, 2000
Scottsdale Marriott Suites
7325 East Third Avenue
Scottsdale, AZ 85251

Phone: +1-602-945-1550
Fax: +1-602-945-2005.

Upcoming Security Conferences

E-Finance Conference and Exposition
November 7-8, 2000
San Francisco

www.efinance-expo.com/default.html

27th Computer Security Conference and Exposition
November 13-15, 2000
Chicago

www.gocsi.com/27th_annual/

Cyber Sabotage – Establishing Security Policy and Strategy To Protect Your Enterprise Assets
November 29 - 30, 2000
Alexandria, VA

www.iqpc.com/cgi-bin/templates/
969053496851135253900002/
gen-event.html?event=748&topic=
E-Security Conference and Exposition

November 30 – December 1
Arlington, VA

www.intmedgrp.com/security

AFCEA Canada (Ottawa Chapter)
Information Technology Security
November 30 - 31, 2000
Ottawa, CA

home.istar.ca/~afcea/events/itsec/
index.htm

Another Cryptographic Option for Wireless Communications: “Lattice” Cryptography from NTRU Cryptosystems

This month’s issue of *Wireless Security Perspectives* is the second in a 3-Part series on Public Key Cryptosystems (PKCS) for wireless communications. The first part, by Certicom, explained why Elliptic Curve Cryptography (ECC) is being considered for the wireless industry. This month, Dan Lieman of NTRU Cryptosystems introduces us to another technique that is also being considered for the wireless industry for the cryptographic protection of communications. NTRU is a relatively new public-key cryptosystem (PKCS) that has some significant performance advantages over existing PKCS options.

Public-Key Cryptography: What and Why?

Before comparing NTRU to current PKCS, such as RSA and ECC, it is worthwhile to review the fundamentals of public-key cryptography. The basic notion is fairly simple. Each user has a pair of keys – a *public key*, and a *private key*. The public key can be disclosed publicly but the private key is a secret the user must keep protected at all times. Ideally, the secret key would be stored and operated in tamper-resistant hardware, such as a Smart Card. These two keys are related in some mathematical way, which depends on the underlying cryptosystem.[1] One key is essentially used for ‘locking’ – for encrypting and the other for ‘unlocking’ – for decrypting.

As in the last issue of *Wireless Security Perspectives*, we will use our two communicants, Alice and Bob, for a discussion of public key cryptography. If Bob wants to send a secure message to Alice, he encrypts the message using Alice’s public key. To decrypt the message, Alice simply uses her private key and the algorithm. The generation of this key pair may be an inexpensive operation or a costly one. This again depends on the underlying cryptosystem. This simple example of public key cryptography is illustrated in Figure 1.

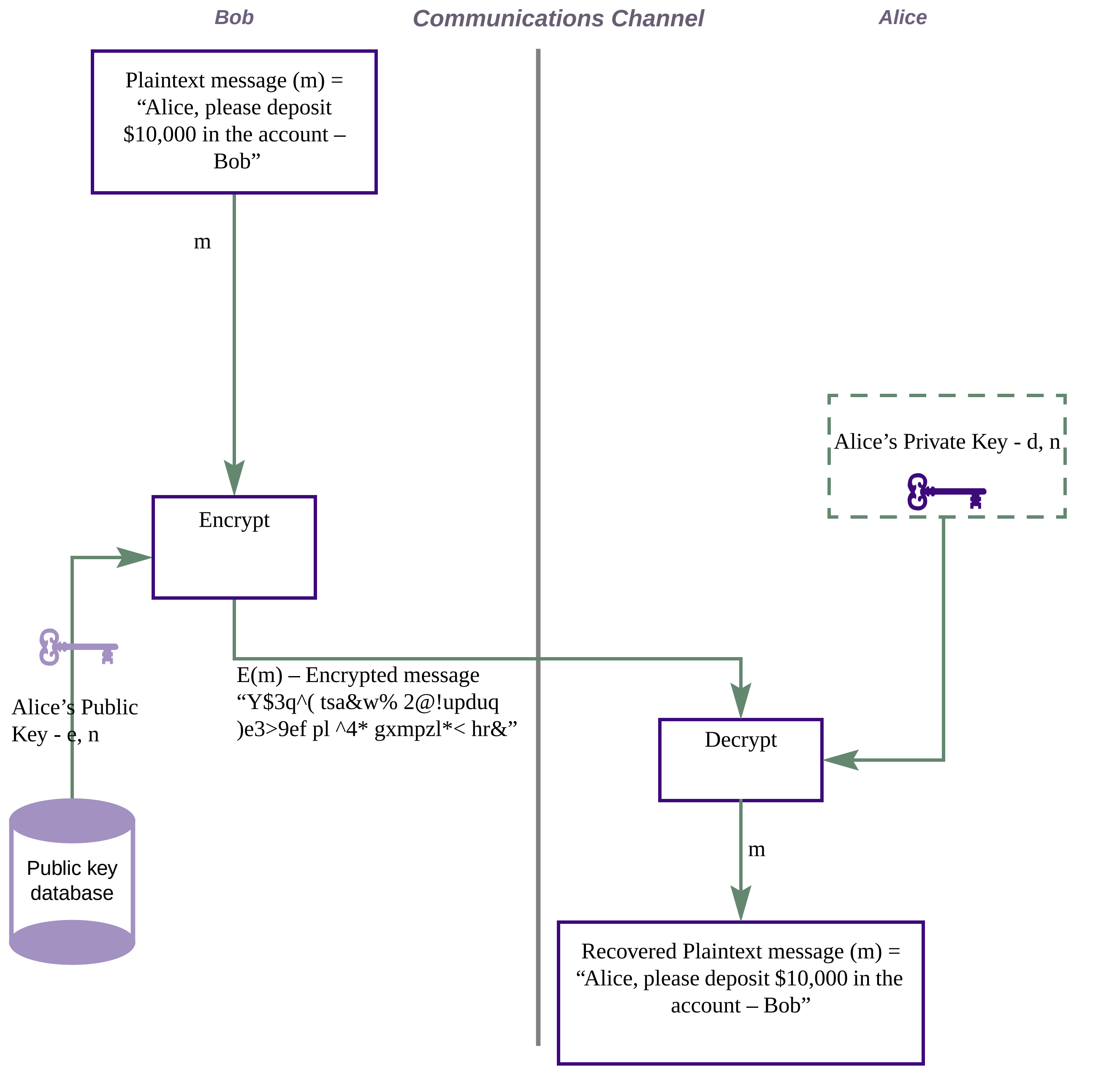
The primary advantage of public-key encryption (we will use ‘encryption’ for simplicity in lieu of the general term cryptography) is that Bob and Alice do not have to meet first to establish a shared secret. Once Alice’s public key is published, anyone can send her a secure message that only she can decrypt. This is ideal in the wireless environment of the future, where ultimately many different operators and application providers may wish to communicate with users as they move geographically and logically through the available wireless offerings.

In the above scenario Bob is certain that only Alice can read the message, assuming that only Alice possesses the private key. However, Alice cannot be certain that the message originated from Bob. This problem is solved with digital signatures, which again follow the public/private model. In this case, Bob can sign the message with his private key (which only he possesses) and Alice can verify the signature using Bob’s public key.

There is also the question of how Bob can be sure he has Alice’s correct public key – this is one of the functions of a Public Key Infrastructure (PKI). See the September, 2000 issue of *Wireless Security Perspectives* for a brief discussion and illustration of digital signatures. PKI will be the subject of a future article in WSP.

[1] For example, in the Rivest-Shamir-Adleman (RSA) scheme, the private key, {d,n}, and public key, {e,n}, are related in the following way: $d = e^{-1} \bmod \Phi(n)$; where Euler’s totient function, $\Phi(n) = (p-1)(q-1)$. Again, $n = p \times q$, and p and q are both prime numbers.

Figure 1: Example of Simple Public Key Cryptosystem (PKCS)


Bob
Communications Channel
Alice
Plaintext message (m) = “Alice, please deposit $10,000 in the account – Bob”
m
Alice’s Private Key - d, n
Encrypt
Alice’s Public Key - e, n
E(m) – Encrypted message “Y$3q^(tsa&w% 2@!upduq)e3>9ef pl ^4* gxmpzl*< hr&”
Decrypt
Public key database
m
Recovered Plaintext message (m) = “Alice, please deposit $10,000 in the account – Bob”

Public Key Encryption for Wireless

There are many different techniques for wireless security including ‘security-through-obscurity’, direct sequence spread spectrum, frequency hopping, and symmetric-key encryption schemes. Most of these only provide security between the operator and the device. Even in the emerging WAP (Wireless Application Protocol) system, cryptographic security is applied from the application server to the gateway, and then from the gateway to the device. The information is actually decrypted and resides in the clear – in plaintext format -- for a while at the gateway. Public-key cryptography allows for end-to-end security between the application server and the device, ensuring that information shared between an application provider and a user is secure. NTRU is the only PKCS that is fast enough to enable end-to-end security with an excellent user experience and without significant deployment costs on the server side.

As another example, a user may wish to wirelessly execute a stock trade, and the transaction may contain items that the trading application should know (e.g. the number of shares, and the price) and information that only the bank should know (e.g. the account from which to withdraw the money and the code for that account). Some of the data should be encrypted with the bank's public key, and some of it, with the stockbroker's public key. In each case, the relevant data should be digitally signed to ensure the trade came from the authorized user. This sort of "field level encryption" is becoming recognized as a requirement for many emerging applications, most notably e-commerce.

NTRU and Characteristics of Public-Key Encryption Systems

One may ask how to select the best public-key cryptosystem. The primary characteristics of a system are its speed and its size (the software code footprint in the device). Other important features include the length of the keys required and the ease of scalability on the server side of the deployment.

NTRU is a new PKCS that encrypts and decrypts 50 to 500 times faster than other PKCS solutions. It is also smaller, so it fits more easily on constrained devices (e.g. in a Subscriber Identity Module 'Smart Card'), draws less battery power, and does not require a coprocessor.

The crucial element that provides these benefits is in NTRU's mathematical structure. The underlying cryptographic algorithms do not involve complex mathematical operations. This is unlike the operations required by ECC or the manipulation of large numbers required by RSA. NTRU only requires many simple operations on small numbers, which can be done easily and quickly on any 8-bit microprocessor, negating the need for a cryptographic coprocessor. The coprocessor that is often required by other PKCS systems results in increased deployment costs, more rapid depletion of battery power and additional memory.

Equally important, NTRU enables rapid key generation. Key pairs can be computed hundreds of times faster than in other techniques. They can be generated by the terminal device and changed quickly, enabling the most sophisticated security paradigms that cannot be achieved using competing technologies. For example, keys with a shorter cryptoperiod are one way to defeat many common cryptographic attacks including timing attacks, which have achieved success against older, computationally intensive cryptographic technologies. For operators, this speed allows solutions that can be deployed with fewer servers and without costly accelerator hardware.

NTRU and the Other PKCS

There are three main PKCS solutions offered in the marketplace today.

1. The Rivest-Shamir-Adleman scheme, or RSA, is the name of a patented PKCS whose primary advocate is RSA Security.
2. Elliptic Curve Cryptography, or ECC, is a technology marketed primarily by Certicom.
3. NTRU, refers to both the patented algorithm and the company, NTRU Cryptosystems, which licenses the NTRU PKCS technology.[1]

To compare the computational complexity of NTRU to RSA and ECC, it is important to understand the fundamental differences between these three systems.

RSA

RSA is the oldest, best-known and the *de facto* standard PKCS. It has received the largest amount of public analysis and scrutiny, and has been deployed in enterprise environments for many years. It is based on the integer factorization problem discussed last month. Its "hard problem" for the cryptanalyst, is factoring a product, N = p q, into its two factors, p and q (See [1] and [2]). An RSA encryption, decryption, signature or verification requires repeated multiplication operations involving very large numbers, roughly 1000 bits long. Because of this, RSA requires a coprocessor to run efficiently in the constrained environments discussed previously. As a result, RSA is inefficient for use in wireless applications.

The RSA patent expired in September, 2000. See the September, 2000 issue of *Wireless Security Perspectives* for more details.

Elliptic Curve Cryptography – ECC

The ECC public key system is younger than RSA, but has received intense independent scrutiny. ECC is based on a hard problem from an area of mathematics called "elliptic curves", also sometimes called ECDLP, or Elliptic Curve Discrete Logarithm Problem. The fundamental operations in ECC involve a complicated formula (multiplication and division of polynomials) that requires multiplications, additions, and divisions on numbers smaller than those used in RSA. These smaller numbers are still fairly large at roughly 170 bits. Because of the complexity of the ECC operations, and the size of the numbers involved, it may also require a coprocessor and has been only sporadically deployed in wireless environments.

NTRU

NTRU, although the youngest of these three technologies, has also been widely studied by independent mathematicians and cryptographers [3]. NTRU is based on a hard problem in an area of mathematics called "lattice theory". This hard problem is completely unrelated to those used in RSA or ECC, and leads to revolutionary speed and efficiency improvements (See [8]). The fundamental operation of NTRU involves additions of many small numbers. Because of NTRU's manipulation of 8-bit numbers, the scheme is capable of encryptions and decryptions that are 50-500 times faster than RSA and ECC (See [4]). These lightweight processing requirements alleviate the need for a coprocessor. Moreover, NTRU key generation of independent keys can be accomplished several hundred times faster than RSA or ECC.

[1] NTRU Cryptosystems is the assignee for US patent 6,081,597, entitled 'Public key cryptosystem method and apparatus', that was issued on June 27th, 2000. Jeffrey Hoffstein, Jill Pipher, and Joseph Silverman are listed as co-inventors.

NTRU – Under the Hood

The basic NTRU data structure is a polynomial with coefficients that are typically 8-bit numbers. For some operations, the coefficients are only 1-bit and 2-bit numbers. The basic NTRU operation is a multiplication of such a polynomial by another which has coefficients that are just zeroes and ones. These binary coefficients provide the basis for NTRU's addition-only mathematics. This operation, called a "convolution product", is then reduced by combining terms in a mathematical process that is easy and fast – but difficult to invert. This one-way function is the critical component of the system security. The steps required to implement NTRU for our two communicants, Alice and Bob, are listed below.

To encrypt a message "m" in the NTRU system, Alice must:

1. Pick a random polynomial "r" comprising a series of zeros and ones.
2. Compute r × h, where h is Bob's public key, and the multiplication is a convolution product.
3. Compute a = rh + m, which is a polynomial addition.
4. Transmit a to Bob.

To decrypt the encrypted message "a" in the NTRU system, Bob must:

1. Compute f × a, where f is Bob's private key, and the multiplication is a convolution product.
2. Reduce the product modulo a prime number.
3. Multiply by the inverse of f.
4. Reduce modulo a second prime number.

As shown, NTRU encryption requires a simple polynomial multiplication and an addition. The NTRU decryption process requires only two polynomial multiplications. In practice, if the private key "f" is chosen from a special class of polynomials, this second convolution product can be ignored, further improving efficiency. It is important to note that, in practice, additional technical steps are performed to protect against standard cryptanalytic attacks. These same steps – called enveloping and padding – are implemented in every secure public-key cryptosystem.

On PKCS Performance

One of the most important comparisons between any two secure cryptosystems is computational performance. At comparable security levels, NTRU is much more efficient than RSA or ECC. Table 1 illustrates the performance of NTRU versus RSA and ECC for a Palm personal digital assistant device.

Table 1: Comparison of PKCS Performance

Operation	NTRU	RSA	Certicom
Encrypt one block	66 ms	1,770 ms	2,000 ms
Decrypt one block	125 ms	25,970 ms	750 ms

As shown, NTRU is significantly more than an order of magnitude faster than RSA or ECC for the encrypt operation. For the decryption function, NTRU also significantly outperforms both RSA and ECC. Equally important to NTRU's performance in client devices, resulting in smaller and less-expensive devices, is performance on the client side. A desktop workstation can perform in software 100 times more NTRU operations per second than the fastest hardware RSA accelerators can perform in the same amount of time at equivalent security levels.

Regarding Large Transfers and Hybrid Cryptosystems

One of the disadvantages of public key cryptosystems is that they are typically slower than so-called symmetric-key (private key or classical) cryptosystems. In a symmetric-key cryptosystem, the same key is used for both encryption and decryption. Obviously, this key must be a shared secret between the encrypting and decrypting parties – Alice and Bob. For large data transfers, for example a large file – be it media or data – or a WAP session, a hybrid cryptographic system model is typically employed. The hybrid system includes both an asymmetric (public) and a symmetric cryptographic algorithm. In this hybrid model, the benefits of both public-key encryption (no prior shared secret need be established) and symmetric-key encryption (speed) can be realized.

In a typical hybrid cryptosystem, for our two communicants Alice and Bob, the following steps must occur.

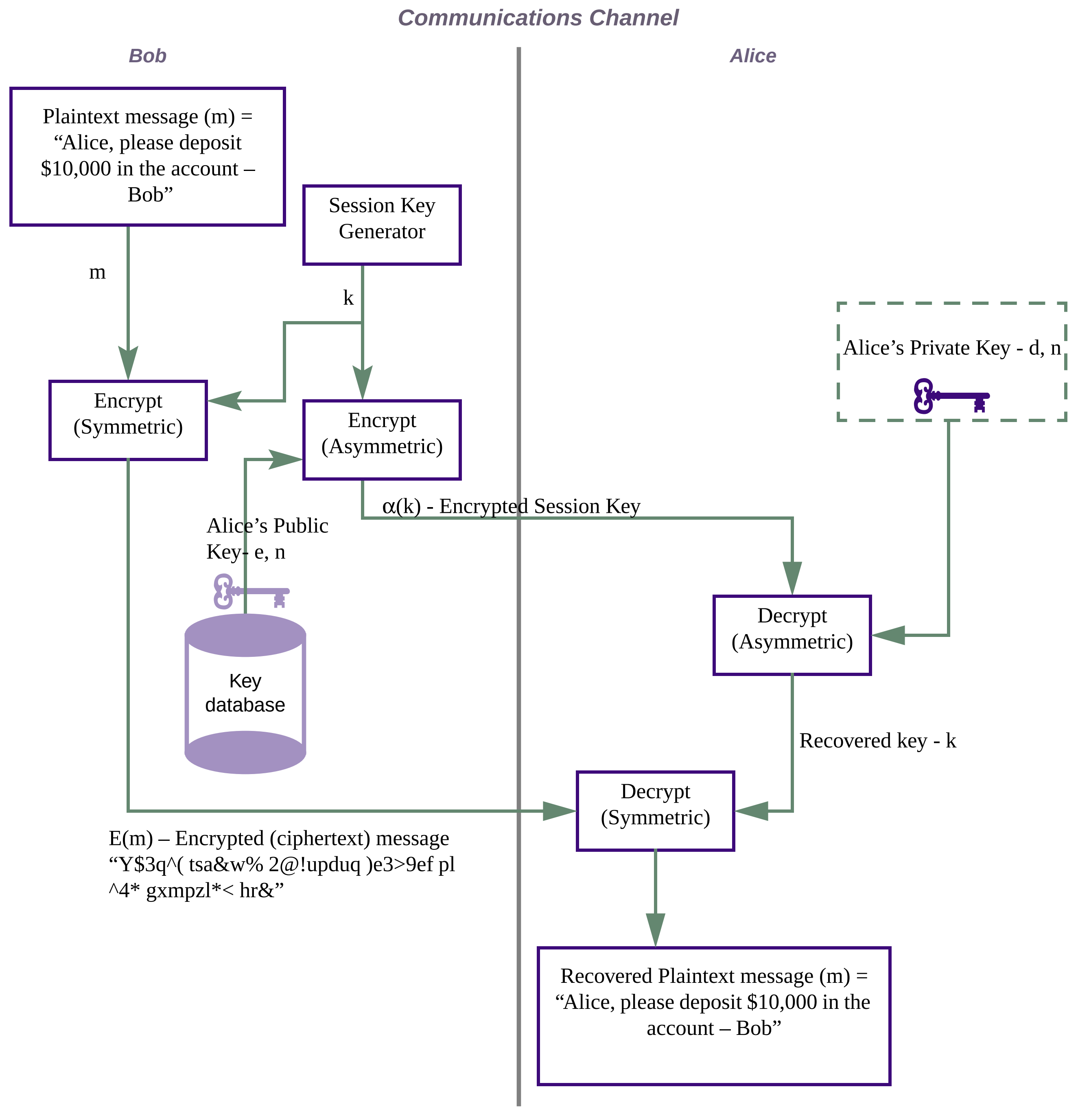
To encrypt a message, Bob must:

1. Generate a random symmetric key.
2. Encrypt the symmetric key.
3. Send this encrypted key to Alice.
4. Encrypt the file and using the symmetric algorithm and the randomly generated symmetric key.
5. Send the encrypted file to Alice.

To decrypt, Alice first decrypts the symmetric key using her private key and the public-key decryption algorithm and then decrypts the file using symmetric encryption. A hybrid cryptosystem, as just described, is depicted in Figure 2.

From Figure 2 one can readily note that the hybrid cryptosystem is more complex and less optimal than the simple non-hybrid cryptosystem of Figure 1. The hybrid system with its complexity and increased code space requirements is required, however, to capitalize on the best characteristics of the symmetric and asymmetric cryptosystems: key management and speed. That is, when using RSA or ECC PKCS, an implementor must, in most instances, take the hybrid approach to gain the benefits. However, remarkably, NTRU, in some instances,

Figure 2: Hybrid Public-Key Cryptosystem


Communications Channel
Bob
Alice
Plaintext message (m) = "Alice, please deposit $10,000 in the account – Bob"
Session Key Generator
m
k
Encrypt (Symmetric)
Encrypt (Asymmetric)
Alice's Private Key - d, n
α(k) - Encrypted Session Key
Alice's Public Key- e, n
Key database
Decrypt (Asymmetric)
Recovered key - k
Decrypt (Symmetric)
E(m) – Encrypted (ciphertext) message
"Y$3q^(tsa&w% 2@!upduq)e3>9ef pl ^4* gxmpzl*< hr&"
Recovered Plaintext message (m) = "Alice, please deposit $10,000 in the account – Bob"

allows implementors to meet the two primary objectives with the simplicity of the Figure 1 model. This is a true paradigm shift in the use of public-key cryptography.

STANDARDS AND SECURITY OF PKCS SYSTEMS

The most important attribute of any PKCS is, obviously, its security. The variety of attacks is bewildering – from simple "brute-force" attacks[1] to sophisticated mathematical attacks involving the hidden structures underlying the mathematics of each PKCS. The most important way to verify that a PKCS is secure is independent scrutiny by the cryptographic community. Although NTRU was invented only 5 years ago, it has rapidly achieved a high level of scrutiny and deployments because of its compelling advantages. Researchers in Europe, the Middle East, Australia, Asia and the US have published papers on NTRU, each reaffirming NTRU's security (See [3], [4], [5], [6] and [7]). Additionally, the IEEE P1363, (Standard Specifications for Public Key Cryptography) committee is beginning the process of standardizing NTRU. This is a powerful endorsement of the rigor to which NTRU has been subjected.

Selected References

[1]. Schneier, Bruce. Applied Cryptography: Protocols, Algorithms and Source Code in C, 2nd Edition. New York: John Wiley and Sons, Inc., 1996.

[2]. Stallings, William. Cryptography and Network Security: Principles and Practice, 2nd Edition. Upper Saddle River, NJ: Prentice Hall, 1998.

[3]. D. Coppersmith, A. Shamir, "Lattice Attacks on NTRU", Presented at Eurocrypt, 1997.

[4]. J. Hoffstein, J. Pipher, J. Silverman, "NTRU: A ring-based public key cryptosystem", Proceedings of ANTS III, Portland (1998), Springer Verlag Lecture Notes in Computer Science.

[5]. A.K. Lenstra, H.W. Lenstra, L. Lovasz, "Factoring polynomials with polynomial coefficients", Math. Annalen 261 (1982) 515-534.

[6]. C.P. Schnorr, M. Euchner, Proc. Fundamentals of computation theory, LNCS 529, pages 68-85, 1991.

[7]. C.P. Schnorr, H.H. Hoerner, "Attacking the Chor-Rivest crypto-system by improved lattice reduction", Proc. Eurocrypt 1995, LNCS 921, 1-12, 1995.

[8]. P. Nguyen, J. Stern, Lattice Reduction in Cryptology: An Update, in Algorithmic Number Theory – proceedings of ANTS-IV (July, 2000).

About the Author

Daniel Lieman, Ph.D., works in NTRU's Burlington, Massachusetts office as Chief Evangelist. Dan co-invented some NTRU technology and represents them on a number of standards bodies, including IEEE P1363 and the WAP Forum. He has over 15 years of experience in systems development and R&D. Dan holds a position at the University of Georgia and has held positions at the Mathematical Sciences Research Institute and Columbia University. He has published over 25 research articles on topics including number theory and cryptography. Dan received numerous teaching awards, and the National Science Foundations CAREER grant. He holds a B.A. in Mathematics from the University of California, Berkeley, and a Ph.D. from Brown University. Dan can be reached at dlieman@ntru.com.

To Probe Further

White papers and other information on NTRU and "lattice" cryptography may be found at www.ntru.com. Interested persons may also contact Steve Sampson, NTRU's Vice-President of Wireless Business Development at:

ssampson@ntru.com

For details of the NTRU patent, visit www.uspto.gov or obtain a copy of patent number 6,081,597 from the US Patent and Trademark Office at the address or telephone numbers below:

General Information
Services Division
U.S. Patent and Trademark Office
Crystal Plaza 3, Room 2C02
Washington, DC 20231
1-800-786-9199 or +1-703-308-4357

For more information on the IEEE's P1363 cryptographic standardization efforts, visit:

grouper.ieee.org/groups/1363

The latest drafts are available at:

stds-bbs.ieee.org/groups/1363/
index.html.

[1] This is an attack during which an adversary tries every possible key. This approach, sometimes referred to as an exhaustive key attack, can easily be thwarted by ensuring that the number of possible keys – the keyspace – is extremely large.