Global Information Assurance Certification Paper

Candidate: Carlos da Roza (carlos.009)
Track 1: GIAC Security Essentials (GSEC) v.1.4b
Option 1 – Research on Topics in Information Security
Category: Encryption & VPNs

Mathematical Underpinnings of Asymmetric Cryptography

Abstract

The mathematics of the Diffie-Hellman-Merkle key exchange and of RSA are explored. Constructs and proofs require no more than college-level math and concepts can be grasped in entirety. The intention is to gain an intuitive understanding of the mathematical underpinnings that make these cryptographic processes work and to demonstrate adeptness with the concepts. A short history introduces the paper, and an analysis of the observations of the properties and behaviour of the plaintext-to-ciphertext transformations follow.

Introduction

Mathematics is considered the queen of sciences, and the precursor of analytic techniques applied to the physical sciences. Like any other discipline, most elementary truths have long been discovered during the science's infancy, so much so that current research is highly abstracted, obscure and well beyond the realm of comprehension of the amateur. Along comes a series of startling discoveries which not only has profound impact on the discipline, but also has immediate and wide-ranging effect in the everyday world.

This paper is a personal exploration of the arithmetic and the intuitive concepts behind the discoveries leading to modern "retail" cryptography, that is, cryptography available to the private individual. Because the material and concepts covered are those of Number Theory and Finite Mathematics at the college level, it is quite possible for the author and reader to visualize and grapple with the problem as a whole, and to follow the mathematical constructions and proofs.

I wanted to get a little bit of the sense of "Aha!" that these mathematicians experienced as the revelation of something profound and wonderful dawned upon them. As a candidate submission for GIAC Security Essentials, I hope to demonstrate a sufficiently deep understanding of what is actually happening as these arithmetical transformations are taking place.

Except for the reference to the recent discovery regarding prime determination, much of the reference material refers to events that happened and items that were published more than a decade ago – the topic of the paper demands

sources of that vintage. Only the spreadsheet and observations can be considered non-derivative from other sources.

Use and Mechanization of Ciphers

The advent of mechanical computing set the tone of the development of cryptography and cryptanalysis in the second half of the twentieth century. The brute force capability of these machines made wholesale and routine encryption possible, and provided the means to attack these ciphers effectively. Secret keys became unsecret with prolonged use, either because of cribs developed from operator error, or just by the volume of source information to work on. The history of cryptanalysis during the Second World War is testament to this weakness[1].

All codes and ciphers depend on sender and receiver to possess a shared secret. This shared secret key was used to both encrypt and decrypt a message. The operation was symmetric, hence these keys are referred to as symmetric keys. As secret keys became longer, their lives shorter, and the necessity for use more frequent, distribution of these keys, particularly by electronic means and at a distance, became a greater and greater problem.

Public Key Exchange

It was not until the middle 1970's when Whitfield Diffie, Martin Hellman, and Ralph Merkle devised their key exchange scheme, that the long-held belief that it was not possible to exchange secret keys securely without sender and receiver meeting was disproved. Their system is based on the one-way modulo function and some of their properties. Let's see how this works.

By convention, Alice and Bob are two characters who wish to exchange information securely. Eve is an eavesdropper on Alice and Bob's public exchange. Alice and Bob publicly agree on values of Y and P ($Y<P$) for the one-way function $Y^x \pmod P$. As an example, choose $Y=7$ and $P=11$.

	Alice	**Bob**
Stage 1	Alice chooses a number, say 3, and keeps it secret. We label her number A.	Bob chooses a number, say 6, and keeps it secret. We label his number B.
Stage 2	Alice puts 3 into the one-way function and works out the result of $7^A \pmod{11}$: $7^3 \pmod{11} = 343 \pmod{11} = 2$	Bob puts 6 into the one-way function and works out the result of $7^B \pmod{11}$: $7^6 \pmod{11} = 117{,}649 \pmod{11} = 4$

Stage 3	Alice calls the result of this calculation α, and she sends her result, 2, to Bob.	Bob calls the result of this calculation β, and he sends his result, 4, to Alice.
The Swap	Ordinarily this would be a crucial moment, because Alice and Bob are exchanging information, and therefore this is an opportunity for Eve, a third party, to eavesdrop and find out the details of the information. However, it runs out that Eve can listen in without it affecting the ultimate security of the system. Alice and Bob could use the same line they used to agree on the values for *Y* and *P*, and Eve could intercept the two numbers that are being exchanged, 2 and 4. However, these numbers are not the key, which is why it does not matter if Eve knows them.	
Stage 4	Alice takes Bob's result, and works out the result of β^A(mod 11): 4^3(mod 11) = 64(mod 11) = 9	Bob takes Alice's result, and works out the result of α^B(mod 11) = 2^6(mod 11) = 64(mod 11) = 9
The Key	Miraculously, Alice and Bob have ended up with the same number, 9. This is the key!	

$$(Y^B(\text{mod } P))^A(\text{mod } P) = (Y^A(\text{mod } P))^B(\text{mod } P) = (Y^B)^A(\text{mod } P) = Y^{AB}(\text{mod } P)$$

The problem for Eve is that although it is easy to derive α from A, and β from B, it is extremely difficult to derive A from α or B from β for numbers which are very large[2].

Asymmetric Keys

Although the Diffie-Hellman-Merkle key exchange system established that it is possible to exchange secret information without meeting, it was awkward because of the need for the exchange to be conversational. Whitfield Diffie followed up with a further paper on the concept of asymmetric keys, that is, different keys are used to encipher and decipher a message. So, knowing the encryption key was useless to the act of decryption. This is a significant departure from the then common use of symmetric keys where encryption is reversible and use the same key. The asymmetric encryption key was to be published, and known as the public key; the decryption key was to be kept private.

In 1977, a different set of researchers, Ronald Rivest, Adi Shamir, and Leonard Adleman, working out of MIT, devised a scheme which would meet Whitfield Diffie's criteria. This was to become the first publicly known form of public key cryptography and is known as RSA.

The basis of the scheme is that Bob, who wants to receive secure messages, publishes a public key, part of which is a composite number of which only he knows the prime factors. Using several properties of modulus arithmetic and theorems of number theory, Bob can derive an encryption key as well as a decryption key from these prime factors of which he publishes only the former.

Prime Factorization is an NP problem

Let's digress for a moment regarding factorization of a composite number. The basis of the strength of the public key is that it is considered difficult to find the factors of a composite number. The problem of finding such factors is considered belonging to the NP class of problems[3], also referred to as intractable - that is, there is no known algorithm one can apply to determine such factors in polynomial time. To put it another way, there is no definitive procedure to find a solution to this problem in a number of steps which has an upper bound expressible as a polynomial function of the length of the composite key.

Please note that it is not possible to prove that a problem is NP, just that no solution has been found that is P. Now, mathematics is not static. Consider that in the last decade, Andrew Wiles solved Fermat's Last Theorem[4], a riddle that had stumped the world's mathematicians for centuries. In the last year, a group of mathematicians have proved and developed an algorithm to determine whether a number is prime in polynomial time[5]. Previously thought to be an NP problem, it has now been shown that prime determination is a P class problem. Current thought is that this does not invalidate the strength of keys as they are based on prime factorization, and that is still intact. However, it is not at all certain that no algorithm will be found that will make prime factorization a tractable problem.

Mechanics of Key Generation and Use in Encryption

Let Bob choose two sufficiently large prime numbers, *p* and *q*, and a third number *e* coprime to the product *(p-1)(q-1)* – two numbers are coprime if they share no common factors, i.e. their greatest common divisor is 1. Let *N* be the product of primes *p* and *q*. Bob publishes *e* and *N* in his public key, known to to world (including Alice and Eve). Although the world knows about *N*, it can determine neither *p* nor *q* in a reasonable amount of time for sufficiently large *N* (given that prime factorization is NP).

Alice uses Bob's public key in order to encrypt a message to Bob. Alice uses some numerical representation of her plaintext message, broken up into blocks (say M) whose value cannot exceed *N* (the reason will become apparent later), and encrypts it by raising it to the power *e*. The result is taken modulo *N* to arrive at the ciphertext C.

$$C = M^e \pmod{N}$$

The exponentiation of M is fairly laborious for larger values of M and *e*. However, *e* can be broken down into a sum of powers of 2 and a judicious selection of *e* allows comparatively speedy calculation. That is,

If $e = a_0 2^0 + a_1 2^1 + \ldots a_n 2^n$ for a finite set $(a_0, a_1, \ldots, a_n)$ where a_i is 0 or 1

Then $M^e = M^{(a_0 2^0 + a_1 2^1 + \ldots a_n 2^n)}$

$$= M^{a_0 2^0} * M^{a_1 2^1} * \ldots M^{a_n 2^n}$$

Since $M^{2^{i+1}} = M^{(2)(2^i)} = (M^{2^i})^2$, each term is the square of the previous term. At most, $\sqrt{e}$ squarings must be calculated.

By selecting *e* so that it the power terms of expansion has coefficients mostly zero, i.e. most of a_i is zero, C can be calculated rapidly. Selection of $e = 2^{16}+2^0$ is common[6].

Another shortcut:

We want to calculate $C = M^e \bmod N$.

We note that $(ab \bmod c) = (a \bmod c)(b \bmod c)$

For proof, note that both a and b can be expressed as:

$a = a_1 c + a_0$ for a_0 in [0,c)
$b = b_1 c + b_0$ for b_0 in [0,c)
$ab = (a_1 c + a_0)(b_1 c + b_0)$ for a_0, b_0 in [0,c)
$= (a_1 b_1 c^2 + a_1 b_0 c + a_0 b_1 c + a_0 b_0)$
$= c(a_1 b_1 c + a_1 b_0 + a_0 b_1) + a_0 b_0$

Hence:

$a \bmod c = (a_1 c + a_0) \bmod c$
$= (a_1 c \bmod c) + (a_0 \bmod c)$
$= 0 + (a_0 \bmod c)$ since c divides cX
$= a_0 \bmod c$
$= a_0$ since a_0 is in [0,c)

Similarly:

$b \bmod c = b_0$

and

$ab \bmod c = a_0 b_0$

Resulting in:

(ab mod c) = (a mod c)(b mod c) by substitution

Thus, while calculating (M^e mod N), it is possible to substitute any intermediate result (i.e. each squaring of the power of 2 expansion of M^e) with its modulo N value thus simplifying the calculation and keeping it within the computable size of the host machine.

By publishing the public key, composed of the exponent e, and the large composite number N, Bob now has given Alice and the rest of the world the ability to encrypt a plaintext message M into ciphertext C.

Decryption Phase

Bob now receives ciphertext C from Alice. Using the values e and N in Bob's public key, there is no straightforward method of reversing the encryption other than by brute force methods.

The encryption exponential e cannot be used to recover the plaintext M from the ciphertext C. In order to do so, it is necessary to generate a decryption exponential d. Generation of d from e and N without knowing p or q is akin to the factorization problem, i.e. intractable.

Digression into Euler's Phi Function and Euclid's Algorithm

We're going to walk into a series of assertions in finite mathematics which will result in the generation of the decryption exponential d.

Euler's phi function[7] for a number n, labelled $\Phi(n)$ is the number of integers less than n which are relatively prime to (i.e. have no common factors with) n. By necessity, $\Phi(n) < n$, and $\Phi(n) = n-1$ when n is prime since the definition of a prime number is that it has no factors other than 1 and itself. Thus all numbers less than a prime number n (which number n-1) are relatively prime to n.

Futhermore, Φ is multiplicative, that is, $\Phi(pq) = \Phi(p)\ \Phi(q)$ when p and q are coprime to each other. Obviously, if p and q are prime numbers, then they are coprime to each other. This means that for prime numbers p and q,

$$\Phi(pq) = \Phi(p)\ \Phi(q) = (p-1)(q-1) \text{ when } p \text{ and } q \text{ are prime.}$$

Also, the phi function number $\Phi(n)$ has the property such that for numbers a less than n which are relatively prime with n, $(a^{\Phi(n)} \bmod n) = 1$. By extension, $(a^{\Phi(n)+1} \bmod n) = a$. This is significant because when you raise a number to a given power mod n, it will restore the original number.

Putting this together, we want to find an exponent d, such that

$$C^d \bmod N = M$$

But

$$C^d \bmod N = (M^e)^d \bmod N = M^{ed} \bmod N = M^{\Phi(N)+1} \bmod N = M$$

if we can find d such that

$$ed = \Phi(N)+1$$

Now,

$$\Phi(N) \bmod \Phi(N) = 0 \quad \text{since everything divides itself, thus}$$
$$\Phi(N)+1 \bmod \Phi(N) = (\Phi(N) \bmod \Phi(N)) + 1 \bmod \Phi(N) = 1$$

Hence, we are looking for d such that

$$ed \bmod \Phi(N) = 1$$

that is, d is the modular inverse of e with respect $\Phi(N)$.

Remember that $N = pq$, a product of primes, so:

$$\begin{aligned}\Phi(N) &= \Phi(pq) \\ &= \Phi(p)\Phi(q) \quad \text{since } \Phi \text{ is multiplicative} \\ &= (p-1)(q-1) \quad \text{since } p \text{ and } q \text{ are primes}\end{aligned}$$

So, to restate, we are searching for d such that

$$ed = 1 \bmod (p-1)(q-1)$$

Here, we use an extension of Euclid's algorithm[8] to determine d. Euclid's algorithm is a method of finding the greatest common divisor (gcd), of two positive integers, that is, the largest number which will divide these integers evenly.

One of the results based on Euclid's algorithm is that given two numbers a and b, one can derive two coefficients a' and b' such that

$$a'a + b'b = \gcd(a,b)$$

One of the earlier conditions in selecting e, p and q was to make e and $(p-1)(q-1)$ relatively prime. That would make their greatest common divisor one. So, if we make

$a = (p\text{-}1)(q\text{-}1)$
$b = e$

then *a'* and *b'* can be calculated so that

$a'(p\text{-}1)(q\text{-}1) + b'e = 1$ since e and $(p\text{-}1)(q\text{-}1)$ are relatively prime

If we take the equality and apply modulus arithmetic on it,

$a'(p\text{-}1)(q\text{-}1) + b'e = 1 \bmod ((p\text{-}1)(q\text{-}1))$
$b'e = 1 \bmod ((p\text{-}1)(q\text{-}1))$ since $((p\text{-}1)(q\text{-}1))$ divides $(a'(p\text{-}1)(q\text{-}1))$

which is the value *d* (i.e. b') we were looking for. We will do an exercise of the extension to Euclid's algorithm will real numbers to establish the method. Again, we are looking at

$a'a + b'b = \gcd(a,b) = 1$ $a = (p\text{-}1)(q\text{-}1)$, $b = e$

We are searching for *b'*.

If *b'* is negative,

$b'b = 1 - a'a$
$b'b = 1 \bmod a$ since a divides $a'a$
$b'b + ab = 1 \bmod a$ since a divides ab
$(b'+a)b = 1 \bmod a$

Hence, if *b'* is negative, just add *a* to it. In fact, all *b'* mod *a* works[9].

Assuming $a > b$, express

$a = p*b + r$ where p is the greatest integer which leaves r positive
$r = a - p*b$ r is the remainder or $r = a \bmod b$

Express

$a = 1*a + 0*b$ trivial,
$b = 0*a + 1*b$ trivial.

Multiplying the second equation by p and subtracting it from the first gives

$a = 1*a + 0*b$
$b = 0*a + 1*b$
$r = 1*a + (-p)$

Repeat the operation with the second pair of equations and continue as such until we cannot procede further. With real numbers, let *a* = 975 and *b* = 616

left side	*a	*b	p
975	1	0	
616	0	1	1
359	1	-1	1
257	-1	2	1
102	2	-3	2
53	-5	8	1
49	7	-11	1
4	-12	19	12
1	151	-239	2
0			

Thus 151*975 + (-239*616) = 1

b' = -239
b' = -239 mod 975
= 975 – 239
= 736

151*975 + 736*616 = 1 mod 975

Spreadsheet Example

Included below is a spreadsheet which illustrates the encryption and decryption process. When valid values of *p*, *q*, *e* and *d* are inserted into the spreadsheet, plaintext values M (in red) are mapped onto ciphertext values C (in yellow), then recovered into plaintext M' (in green). Spreadsheet limitations constrain *d*'s value to 254.

p	q	N	(p-1)(q-1)		e	d	e*d	e*d(mod p-1 q-1)				Open	M		C		M'				
11	17	187		160	13	37	481	1				Field	plaintext		ciphertext		recovered				
Exponent		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	151	2	4	8	16	32	64	128	69	138	89	178	169	151	115	43	86	172	157	127	67
2	2	151	174	94	169	87	47	178	137	117	89	162	152	138	81	76	69	134	38	128	67
3	148	3	9	27	81	56	168	130	16	48	144	58	174	148	70	23	69	20	60	180	166

Observations

While experimenting with small values of *p*, *q*, *e* and *d*, some properties of the encryption and decryption process show up.

First, the values that M can take on must be in [0,*N*). Once M reaches *N*, C wraps. That is, M=0 and M=*N* wrap to the same value C_0, M=1 and M=*N*+1 wrap to C_1, etc. Now, although *N* is normally large, so should the potential

values of M. Because asymmetric encryption is considerably more compute intensive than symmetric encryption, a typical security application would do bulk encryption with a symmetric key, then encrypt the symmetric key with an asymmetric public/private key pair[10] (depending on whether authentication of source, or message security is desired). Longer symmetric keys make for better security.

The values M_i in [0,*N*) maps one-to-one onto values C_i also in [0,*N*), so the values of C are just a permutation of M. C must also map one-on-one to M'. When *e* and (*p*-1)(*q*-1) are not coprime, M_i does not map one-to-one onto C_i. Mapping one-on-one seems to be a consequence of *e* being coprime to (*p*-1)(*q*-1), or $\Phi(pq)$. By this reasoning, if *e* is coprime to (*p*-1)(*q*-1), then so must *d*.

If you chase a train of transformations, that is, starting with M_1, determining C_1, taking $M_2=C_1$ and determining C_2, and so on, these trains form closed loops (e.g., with {*p*,*q*,*e*,*d*}={11,17,3,27}, 2⇒128⇒138⇒162⇒2. These trains are of length 1 (M=0, M=1), 2 or 4. A line of inquiry would be to see if these train lengths are powers of 2, and why.

M=0 and M=1 are the only values that map onto themselves for all values of *e*.

Aftermath

Since the time of the development of RSA, the personal computer revolution has taken off. With it, personal information security has become big business. Phil Zimmerman brought it to the masses with his distribution of PGP (Pretty Good Privacy), incorporating RSA and IDEA, PGP's asymmetric and symmetric aspects respectively; this despite patent protection and government prosecution[11]. Generally, the availability and popularity of personal computing spawned rapid development in cryptology along these lines, with fast factorization methods and more defensible primes being developed. The tension between cryptology and cryptanalysis continues as it has for hundreds of years. Sometimes the codemakers pull ahead, sometimes the codebreakers.

Conclusions

Being at the cusp of change for which you are the agent of change is something few of us have the opportunity to experience. One can only remotely and vicariously imagine the fever that drove Whitfield Diffie that afternoon in 1975, and Ronald Rivest during his 1977 overnight travail as they saw their goals come within their grasp. Personally, the walk through the arithmetical constructions and proofs was illuminating as the mathematical truths that serve as the underpinnings of modern cryptology emerged. The fact that it all makes sense and the arguments can be followed is not a surprise. Nevertheless, the exercise is useful in removing the magic that surrounds this process which is

not immediately intuitive.

References

(1) "Codebreaking and Secret Weapons in World War II", URL: http://home.earthlink.net/~nbrass1/enigma.htm (September 25, 2002)

(2) Singh, Simon. The Code Book, The Science of Secrecy from Ancient Egypt to Quantum Cryptography, Toronto: Random House of Canada ©1999, 265

(3) "Wikipedia's Integer factorization", Wikipedia, the free encyclopedia, URL: http://www.wikipedia.com/wiki/Integer_factorization (September 25, 2002)

(4) Singh, Simon. Fermat's Enigma, The Epic Quest to Solve the World's Greatest Mathematical Problem, Toronto: Penguin Books Canada, Ltd. ©1997

(5) "PRIMES is in P", Indian Institute of Technology Kanpur, Department of Computer Science & Engineering, URL: http://www.cse.iitk.ac.in/news/primality.html (September 25, 2002)

(6) Ronan Killeen, "Possible Attacks on RSA", RSA: Hacking and Cracking, URL: http://members.tripod.com/irish_ronan/rsa/attacks.html (September 25, 2002)

(7) "Euler's phi function", Wikipedia, the Free Encyclopedia, URL: http://www.wikipedia.com/wiki/Eulers_phi_function (September 25, 2002)

(8) "Extension of Euclid's Algorithm", URL: http://www.cut-the-knot.org/blue/extension.shtml (September 25, 2002)

(9) "Example of RSA key generation ", GT Information Security, URL: http://www.security.gatech.edu/protection/rsa/key.html (September 25, 2002)

(10) "How PGP works", The International PGP Home Page, URL: http://www.pgpi.org/doc/pgpintro/p10 (September 25, 2002)

(11) "The Feasibility of Breaking PGP", The PGP Attack FAQ URL: http://www.stack.nl/~galactus/remailers/attack-faq.html (September 25, 2002)