The Elliptic Curve Group Over Finite Fields: Applications in Cryptography

by

Jeremy W. Lester

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Mathematics

Program

YOUNGSTOWN STATE UNIVERSITY

August, 2012

The Elliptic Curve Group Over Finite Fields: Applications in Cryptography

Jeremy W. Lester

Signature:

Jeremy W. Lester, Student Date

Approvals:

Dr. Jacek Fabrykowski, Thesis Advisor Date

Dr. Neil Flowers, Committee Member Date

Dr. Thomas Smotzer, Committee Member Date

Peter J. Kasvinsky, Dean of School of Graduate Studies & Research Date

ABSTRACT

It is the intent of this thesis to study the mathematics, and applications behind the elliptic curve group over $\mathbb{F}_p$. Beginning with the definition of the $'+'$ operation, under which the points on the elliptic curves form an abelian group. Then moving to a brief introduction to both public, and private key cryptography. This will lead into an explanation of the discrete logarithm problem along with an implementation using the elliptic curve group over $\mathbb{F}_p$. This thesis will conclude with an exploration Lenstra's factoring algorithm using the elliptic curve group.

I dedicate this paper to Brian Michael Irby, and the entire Irby family, without whom none of this would be possible. Their altruistic gift that provided my life saving transplant, is the reason I am here and I will be forever grateful.

A special thank you to my loving wife, your encouragement has helped me to realize my potential, and reach higher than I imagined possible.

Thank you to my family for all of your love and sacrifice over the years which have helped me to succeed, my brother's dedication to work and family which has served as an exceptional example for me over the years. Finally, for my mother's faith in God that has reminded me time and time again that we are not alone in our endeavors, and that God is always faithful.

Contents

1 Introduction

With the world becoming ever more reliant on technology, the topic of cryptography is becoming increasingly important. This paper provides a look at the elliptic curve group (ECG) over finite prime fields, and the applications of this group in cryptography. The ECG provides strong underlying security of the discrete logarithm problem which is the basis of many modern cryptographic schemes. In the following sections I will describe the ECG over finite prime fields, the discrete logarithm problem, the elliptic curve cryptosystem, and finally elliptic curve factorization.

2 The Elliptic Curve Group Over Finite Fields

Let $\mathbb{F}_p$ be a finite, prime order field, such that the characteristic of $\mathbb{F}_p$ is not 2 or 3, and $E_{(\alpha,\beta)}$ be an elliptic curve of the form $y^2 = x^3 + \alpha x + \beta \in \mathbb{F}_p[x]$. Also, the discriminant $\Delta = 4\alpha^3 + 27\beta^3$ is nonzero, this happens in the case of a repeated root. We do not allow this since these curves are nonsingular. Now we define,

$$E_{(\alpha,\beta)}(\mathbb{F}_p) = \{(x,y) \in \mathbb{F}_p \times \mathbb{F}_p \mid y^2 = x^3 + \alpha x + \beta \ (mod\ p) \text{ where } \alpha, \beta \in \mathbb{F}_p\} \bigcup \{\infty\}$$

as the set of elliptic curves over $\mathbb{F}_p$, together with ∞ the point at infinity. The point at infinity ∞ also denoted as (∞, ∞), is said to exist at the top of the $x-axis$. This point's existence is necessary in the case of a vertical line, which is said to go through this point. Adding together two points on $E_{(\alpha,\beta)}(\mathbb{F}_p)$, can be thought of as drawing a straight line through the two points, this will be better illustrated in Section 2.1. The line will intersect the curve at a third point called $-R$, this point is then reflected about the $x-axis$ to give us our sum, R, in the case of a vertical line, our sum is ∞.

Definition 2.1. *If P,$Q \in E_{(\alpha,\beta)}(\mathbb{F}_p)$ such that $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, we define $P + Q = R$ as follows:*

If $P \neq Q$, and $P \neq \infty$ then,

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \ \ and \ \ R = (x_3, y_3) = (m^2 - x_1 - x_2, m(x_1 - x_3) - y_1)$$

where, m represents the slope of the line between P and Q.

If $P = Q$, and and $P \neq \infty$ then we can think of m as the slope of the tangent line,

$$m = \frac{dy}{dx} = \frac{3x_1^2 + \alpha}{2y_1} \ \ and \ \ R = (x_3, y_3) = (m^2 - 2x_1, m(x_1 - x_3) - y_1)$$

If $Q = \infty$, then $P + Q = P + \infty = P = (x_1, y_1)$.

Finally, if $Q = -P$, where $P = (x_1, y_1)$ and $-P = (x_1, -y_1)$ then,

$$P + Q = P + (-P) = \infty \ since,$$

$$m \neq \frac{y_2 - y_1}{x_2 - x_1} \ is \ \ undefined \ \ and \ \ hence \ \ a \ \ vertical \ \ line, \ \ and \ \ R = \infty.$$

Theorem 2.1. *$E_{(\alpha,\beta)}(\mathbb{F}_p)$ together with the "+" operation defined in Definition 2.1, forms an abelian group provided $\Delta = 4\alpha^3 + 27\beta^3 \neq 0$. That is for all P, Q, and R $\in E_{(\alpha,\beta)}(\mathbb{F}_p)$, the following properties hold:*

1) $P + Q \in E_{(\alpha,\beta)}(\mathbb{F}_p)$ (Closure)

2) $(P + Q) + R = P + (Q + R)$ (Associativity)

3) There exists $\infty \in E_{(\alpha,\beta)}(\mathbb{F}_p)$ such that $P + \infty = \infty + P = P$ (Identity)

4) There exists $-P \in E_{(\alpha,\beta)}(\mathbb{F}_p)$ such that $P + (-P) = \infty$ (Inverse)

5) $P + Q = Q + P$ (Commutativity)

Proof. (Closure) Let $E_{(\alpha,\beta)}(\mathbb{F}_p)$, be an elliptic curve over the field F, and let P Q $\in E_{(\alpha,\beta)}(\mathbb{F}_p)$, the P + Q as so defined in Definition 2.1 is closed.

(Associativity) Let P, Q, R be points on an elliptic curve E. Define the lines

$$l_1 = \overline{P, Q,} \qquad l_2 = \overline{\infty, Q + R,} \qquad l_3 = \overline{R, P + Q}$$

$$m_1 = \overline{Q, R,} \qquad m_2 = \overline{\infty, P + Q,} \qquad m_3 = \overline{P, Q + R}$$

It can be easily veried that these line have the following intersections (where X is unknown).

	l_1	l_2	l_3
m_1	Q	-(Q+R)	R
m_2	-(P+Q)	∞	P+Q
m_3	P	Q+R	X

First we deal with some special cases:

(i) If P, Q or R is ∞ then association is trivial. For example, if $P = \infty$ then, as required

$$(P + Q) + R = (Q) + R = Q + R$$

$$P + (Q + R) = (Q + R) = Q + R$$

(ii) If $P + Q = \infty$ then

$$(P + Q) + R = \infty + R = R$$

(iii) If $Q + R = \infty$ then associativity holds similarly to above.

So now assume that P, Q, R,(P + Q),(Q + R) $\neq \infty$. We must now verify the assumptions of Theorem A.9 for the remaining cases. Now, if two of the points on a line are equal then by definition the line through them will be the tangent line, and will intersect to order 2. If three of the points are equal then it implies that all three are ∞. Earlier we saw that if the tangent line to the curve intersects at ∞ then it will intersect to order 3, so this assumption is satisfied.

Suppose that $l_i = 6m_j$ for all i, j. Then the assumptions of Theorem A.9 [1, p. 114] are all satisfied and so all the points in the table, including X lie on E. Now l_3 will have three points of intersection with E; $R, (P + Q)$ and X. By the definition of elliptic curve addition we have

$$X = -[(P + Q) + R]$$

Similarly m_3 intersects E in three places; $P, (Q + R)$ and X so

$$X = -[P + (Q + R)]$$

So we see that, $(P + Q) + R = P + (Q + R)$ as desired.

Our final task will be to consider what happens if some line l_i equals some line m_j. First observe the following three results:

(i) If P, Q, R are collinear then

$$(P+Q)+R = (-R)+R = \infty \text{ and } P+(Q+R) = P+(-P) = \infty$$

So associativity holds.

(ii) If P,Q, and (Q+R) are collinear then,

$$P+(Q+R) = -Q.$$
$$Also, P+Q = -(Q+R)$$

so

$$(P+Q)+R = -(Q+R)+R = -Q$$

where the second equality is proved by Lemma 2 below.

(iii) If $Q, R, (P+Q)$ are collinear then associativity holds as above.

Lemma 2.1. *Let P_1, P_2 be points on an elliptic curve. Then*

$$(P_1+P_2)-P_2 = P_1 \text{ } and \text{ } -(P_1+P_2)+P_2 = -P_1$$

Proof. The first equation is the reection of the second so we just prove the second. The line, L, through P_1 and P_2 intersects the elliptic curve again at $-(P_1+P_2)$. So to calculate $-(P_1+P_2)+P_2$ we would draw the line between them which is L. This cuts again at P_1 so its reection is $-P_1$. □

Now suppose $l_i = m_j$ for some i, j. We can assume the all the points of intersection except ∞ and possibly X are finite. Consider the various cases

(i) $l_1 = m_1$: Then P, Q, and R are on the same line. This means they are collinear and so associativity follows.

(ii) $l_1 = m_2 : \overline{\infty, P+Q}$ is a verticle line so $\overline{PQ}$ is too. Therefore $P + Q = \infty$, and by the earlier argument associativity follows.

(iii) $l_2 = m_1$: In this case its $Q + R = \infty$ so associativity holds similarly.

(iv) $l_1 = m_3$: Then P, Q and $(Q + R)$ are collinear, so associativity holds.

(v) $l_3 = m_1$: Then Q, R and $(P + Q)$ are collinear, so associativity holds.

(vi) $l_2 = m_2$: So we know that $(P + Q), (Q + R)$ and are on this line. So $P + Q = (Q + R)$. If $P + Q = Q + R$ then by Lemma 2

$$P = (P + Q) - Q = (Q + R) - Q = R$$

Therefore

$(P + Q) + R = R + (P + Q) = P + (P + Q) = P + (R + Q) = P + (Q + R)$ as required. If $P + Q = -(Q + R)$, then

$$(P + Q) + R = -(Q + R) + R = -Q$$
$$P + (Q + R) = P - (P + Q) = -Q$$

So associativity holds.

(vii)) $l_2 = m_3$: We have a line with $P, (Q+R), \infty$ on it meaning $P = -(Q+R)$. Since Q, R and $-(Q+R)$ are collinear by denition we have that Q and R are on this line as well. So P, Q, and R are collinear and associativity holds.

(viii) $l_3 = m_2$: We have a line with $R, (P+Q)$, and ∞ on it so associativity holds similarly to the previous case.

(ix) $l_3 = m_3$: So $P, R, (Q+R)$ and $(P+Q)$ lie on the same line, but this line cannot intersect in 4 points, so either $P = R, P = P+Q$ or $Q+R = P+Q$ (other combinations would imply was on the line. If P = R then we are in the case $l_2 = m_2$. If $P = P+Q$ then

$$\begin{aligned} P - P &= (P+Q) - P \\ \infty &= Q \end{aligned}$$

and so associativity follows. If $Q+R = P+Q$ then similarly adding $-Q$, gives $P = R$ which we have already treated. So this completes the proof of associativity for all possible cases. [1]

(Identity) Consider the point at infinity ∞. Then if $P \in E_{(\alpha,\beta)}(\mathbb{F}_p)$ we see that $P + \infty = P = \infty + P$ from Definition 2.1. Hence ∞ serves as the identity for $E_{(\alpha,\beta)}(\mathbb{F}_p)$.

(Inverse) Let $P \in E_{(\alpha,\beta)}(\mathbb{F}_p)$ such that $P = (x_1, y_1)$. Then consider the point $-P = (x_1, -y_1)$, where $-y_1$ is the inverse of $y_1 \in \mathbb{F}_p$. Now $-P$ is an element of $\mathbb{F}_p$, and $P + -P = \infty$ as in Definition 2.1, hence the existence of the inverse.

(Commutativity) Commutativity follows immediately from Definition 2.1, since

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_1 - y_2}{x_1 - x_2}.$$

Which is the result of the commutativity of $\mathbb{F}_p$. □

2.1 The Elliptic Curve Group over $\mathbb{R}$

The elliptic curve group over $\mathbb{R}$ is perhaps one of the more intuitive examples, defined as

$$E_{(\alpha,\beta)}(\mathbb{R}) = \{(x, y) \in \mathbb{R} \times \mathbb{R} \mid y^2 = x^3 + \alpha x + \beta \text{ where } \alpha, \beta \in \mathbb{R}\} \bigcup \{\infty\}.$$

Then as long as $\Delta = 4\alpha^3 + 27\beta^3 \neq 0$, we see the following three curves:

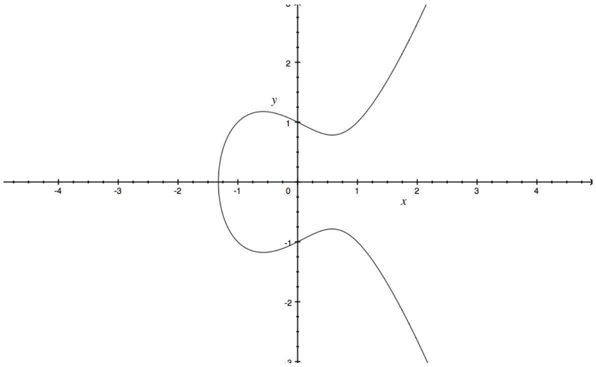

Figure 1: $y^2 = x^3 - x + 4$

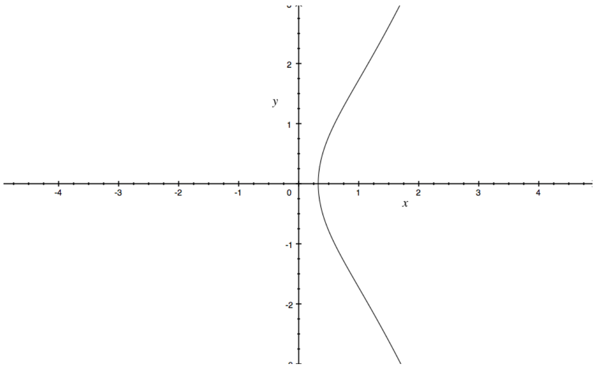

Figure 2: $y^2 = x^3 + x - .5$

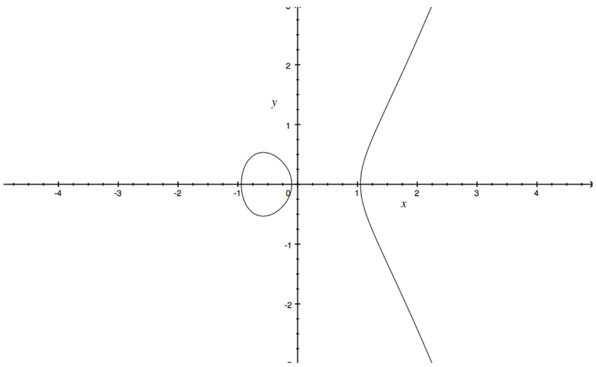

Figure 3: $y^2 = x^3 - x - .1$

In the case where $\Delta = 4\alpha^3 + 27\beta^3 = 0$, we see in Figure 4 that the curve contains a double root and hence is singular. We see also that this curve violates Definition 2.1, observe that if you draw a horizontal line along the $x - axis$ you will see that the line does not intersect the curve at a third point. It is this reason that we must exclude these graphs from our definition.

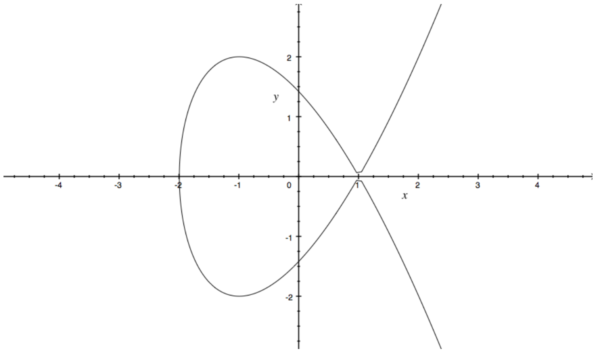

Figure 4: $y^2 = x^3 - \sqrt[3]{27}x + 2$

Let us consider the following curve, it can be easily verified that the points $P = (-1.54, 1.38)$, and $Q = (0.0, 2.0)$ lie on this curve. Then by the Definition 2.1, since $P \neq Q \neq \infty, m = .62/1.54 =\sim .4026$, and $R = (1.71, -2.69)$ as illustrated in Figure 5.

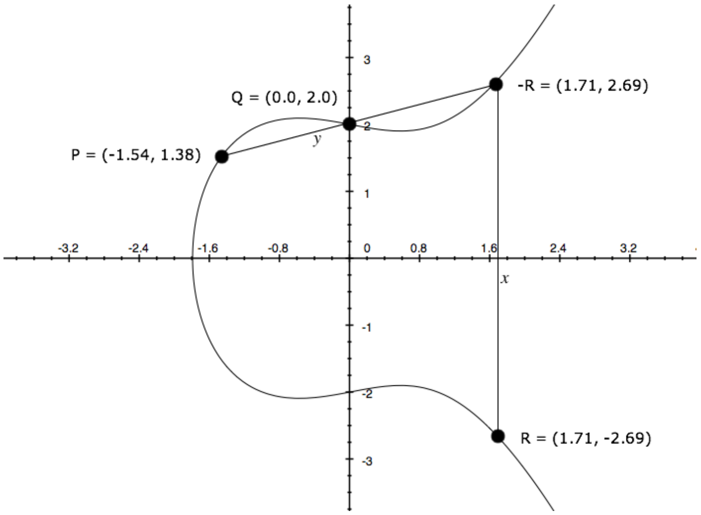

Figure 5: $y^2 = x^3 - x + 4$

Now consider the same curve as in the previous example, and let $P = (-0.847, 2.06)$. Now $P+P = (-0.847, 2.06)+(-0.847, 2.06)$, with $'+'$ as in Definition 2.1. Then $m = -0.7651$ and $-R = (1.77, 2.79)$ which implies $R = (1.77, -2.79)$ which is illustrated in Figure 6

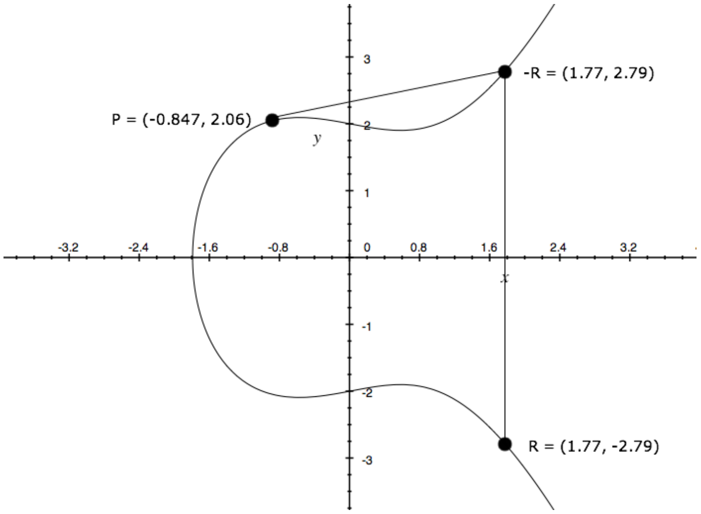

Figure 6: $y^2 = x^3 - x + 4$

For a final example on the same curve, consider $P + -P$ again with $'+'$ as in Definition 2.1. This time let $P = (-0.77, 2.08)$, then by our definition of the inverse, $-P$ is reflected about the $X - axis$ and we get $-P = (-.077, -2.08)$. Now the slope for $P + -P$ is undefined since $x_2 - x_1 = 0$, hence we have a vertical line, which by definition intersects ∞.

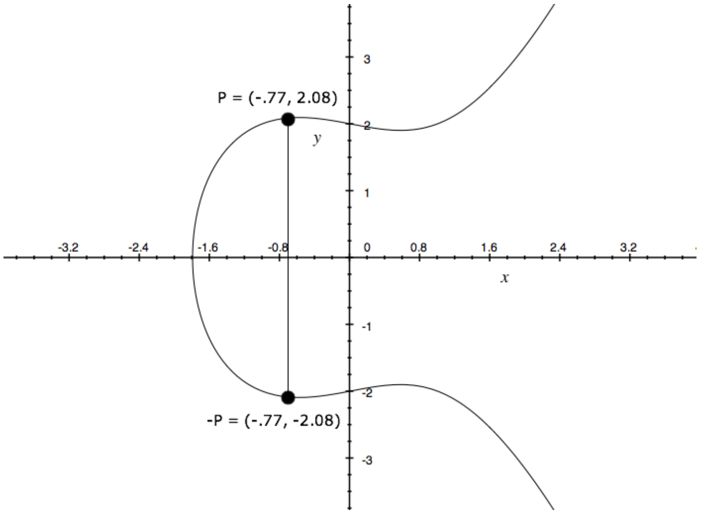

Figure 7: $y^2 = x^3 - x + 4$

3 Applications of the Elliptic Curve Group

In the following section we will see some of the applications of the ECG as it pertains to cryptography. As it turns out the ECG's group structure is such that the discrete logarithm problem is extremely hard given a large enough order. The following sections will describe current public and private cryptography schemes, which will lead to how the ECG will apply in these situations.

3.1 Cryptography

From warfare to commerce, cryptography has played an important part in human history. Julius Caesar famously encrypted his messages by shifting the letters of the alphabet by three places. This method of shifting the letters of the alphabet is now known as a "Caesar Cipher", and in general enciphering plaintext α in this manner can be described as $E(\alpha) = \alpha + k \ (mod\ n)$ where n is the number of characters in the alphabet used, and k is the key. Similarly, deciphering $E(\alpha)$, is described as $D(E(\alpha)) = E(\alpha) - k \ (mod\ n)$.

The many advances in technology that have taken place in the years since have rendered this, as well as many other algorithms useless in hiding information. Computers that are capable of executing billions of operations per second can easily exhaust all possible keys, otherwise known as a brute force attack, in mere milliseconds. This same advance in computing power is also responsible for the need of new more powerful cryptographic schemes. The more powerful computers become the more useful they are for things like online banking, e-commerce, and sharing information. On the other hand the more powerful computers become the harder it becomes to come up with a secure algorithm. In the next few sections I will discuss some of these algorithms as well as describe the role the elliptic curve group takes in these schemes.

3.2 Private Key Cryptography

Private key cryptography, is any cryptographic scheme in which the same key which is used to encrypt and decrypt. This type of cryptography is also known as symmetric key cryptography (SKC). The security of these algorithms rely on a secret key that only the parties communicating know. Algorithms which rely on SKC, tend to be extremely fast, compared to others. The most notable algorithm is the Data Encryption Standard (DES), which uses a 56 bit key. However, with increasing computational power, the 56 bit key size has since been deemed ineffective against a brute force attack. Thus, Triple DES was introduced, which has a larger key size of 112 bits, providing enough security to protect against the brute force attack.

As with any cryptographic scheme, SKC has its benefits, as well as its drawbacks. For example the problem of distributing the key to trusted parties, which could involve another key and a separate scheme altogether. Also, since the same key is used for both encrypting and decrypting, anyone who has access to the key, has access to the plaintext. Consider the situation in which a bank would like to send an encrypted message to an ATM with your account information. The bank uses SKC to encrypt the message containing your account balance. If you were to somehow gain knowledge of this key and intercept the message on the way to the ATM, you could, in fact, decrypt the message, change the amount, re-encrypt the message and send it on its way. Then upon receiving this message the ATM decides to let you withdraw the extra 10,000 dollars you so generously gave yourself. This process of intercepting, changing, and passing on a message is known as a ”man in the middle” attack.

3.3 Public Key Cryptography

Public Key Cryptography, relies on both a pubic, and a private key for its security. The public key is broadcast to the intended recipient unencrypted, as plaintext. This means that anyone listening in on this transmission has access to the senders public key. This may sound unintuitive since the whole point of cryptography is to hide important information, not broadcast it, but the real security lies in the private key. There are many algorithms to create such public/private key pairs but all of them rely on some sort of "trap door" function, that is relatively easy to compute one way, and computationally infeasible to compute the inverse.

For example, Alice wants to send a message to Bob using a public key cryptosystem. She obtains Bob's public key that is known to everyone, and encrypts her message with it. She then sends this message to Bob, who is the only one who knows how to decrypt it since he is the one who generated the key pair. In other words, there are two keys, the one that encrypts the message and another key that decrypts the message. This is why sending out the public key in no way endangers the message from being decrypted.

These methods also have their drawbacks. Since everyone has access to Bob's public key, Bob can't be sure that it is in fact Alice who is sending him this message. Someone could very easily pretend to be Alice, use Bob's public key, and send him a message pretending to be Alice. This is called Masquerading, and it is the reason for certificate authorities which assure you that the sender is in fact who they say they are. Certificate authorities are however beyond the scope of this paper.

4 The Discrete Logarithm Problem

The security of elliptic curve cryptography lies in what's called a "trap door function". A trap door function, is a function in which computations are easy one way, where as computing the inverse operation is very hard. Hard enough that with current computational power this problem, it would still take many years to solve. Consider the following example:

Let G be a finite cyclic group, such that $|G| = n$. Let a be the generator of G, then $G = \langle a \rangle = \{a^0, a^1, a^2, \cdots, a^{n-1}\}$. Now to calculate any power of a by adding the exponents *mod* n, *i.e.*, $a^2 + a^6 = a^{2+6}(mod\ n)$. So as you can see calculating any power of the generator is easy, and can be computed quickly. Finding the inverse, or solving $a^x = a^y(mod\ (n))$, for x by computing $log_a(a^y) = x$. This problem is called the discrete logarithm problem.

Theorem 4.1. *Let G be a group such that $|G| = p$ for some prime p. Then G is a cyclic group.*

Proof. Since $|G| \geq 2$ there exists $x \in G$ such that $x \neq 1$, then $\langle x \rangle \leq G$. So by LaGrange's Theorem $|\langle x \rangle|$ divides $p = |G|$. Since p is prime we get that $\langle x \rangle = 1$ or p. If $|\langle x \rangle| = 1$, then $\langle x \rangle = \{1\}$, and so $x = 1$ a contradiction since $x \neq 1$. Thus $|\langle x \rangle| = p = |G|$. Now since $\langle x \rangle \leq G$ we get $\langle x \rangle = G$. □

Consider the additive group $(\mathbb{Z}_7, +) = \{0, 1, 2, 3, 4, 5, 6\}$, then $|\mathbb{Z}_7| = 7$, hence $\mathbb{Z}_7$ is cyclic by Theorem 4.1 and $\mathbb{Z}_7 = \langle 4 \rangle$. Now consider $(3)4 = y\ (mod\ 7)$, it is easy to calculate $(3)4 = 12 \equiv 5\ (mod\ 7)$ to find y. However computing the discrete logarithm of this is not very easy, finding $y \equiv (x)4\ (mod\ 7)$ is difficult even for this trivial example. This problem becomes difficult the larger as the modulus becomes sufficiently large.

At that point checking every power of the generator becomes infeasible. Also, we see that 3 is not the only solution to this problem since $(7)4 = 0$ $(mod\ 7)$ we can see that $(7n)4 = 0$ $(mod\ 7)$ for any integer n. Hence $(3)4 + (7n)4 = 5 + (0)4 = 5$ $(mod\ 7)$, and we have infinitely many solutions. It is this situation which makes the discrete logarithm, a favorite among cryptographic schemes. In 1976 it was Whitfield Diffie and Martin Hellman who first exploited this problem in what is now known as the Diffie - Hellman key exchange algorithm described in Section 4.1. [4]

4.1 The Diffie - Hellman Key Exchange

The Diffie - Hellman key exchange algorithm is a public key algorithm which directly exploits the discrete logarithm problem. In this protocol, as in the example in the previous section, two numbers are chosen. The first number p, is a large prime roughly 300 digits long, the second number g, is the generator of the group $(\mathbb{Z}_p, \cdot)$ which is guaranteed to be cyclic by Theorem 4.1. These numbers are public and are agreed upon between the two parties communicating, Alice and Bob.

4.1.1 Diffie-Hellman Key Generation

Alice and Bob publicly agree upon p and g, and we assume that anyone listening has access to this information. Then key generation is as follows:

(1) Alice picks a number a, where $1 < a < p-1$ and then calculates $A_k = g^a$ $(mod\ p)$, and sends A_k to Bob.

(2) Bob also picks a number b, where $1 < b < p - 1$ and calculates $B_k = g^b$ $(mod\ p)$, and sends B_k to Alice.

(3) Alice receives B_k from Bob and computes B_k^a $(mod\ p) = K_{ab}$

(4) Bob receives A_k from Alice and computes $A_k^b(mod\ p) = K_{ab}$

Now K_{ab} is a symmetric key that both Alice and Bob can use to communicate with each other via some symmetric key algorithm. Alice and Bob never send a or b respectively, they are kept secret as their private keys. Then anyone wishing to gain knowledge of their private keys would have to compute $g^x = A_k\ (mod\ p)$ and $g^y = B_k\ (mod\ p)$ which is the discrete logarithm of x and y respectively.

4.1.2 Example: Diffie-Hellman Key Exchange

Alice and Bob choose $p = 17$, and $g = 3$, since $\langle 3 \rangle = \mathbb{Z}_{17}$. Alice chooses $a = 6$, and computes $A_k = 3^6\ (mod\ 17) = 15$ and sends it to Bob. Bob then chooses $b = 7$, and computes $B_k = 3^7\ (mod\ 17) = 11$ and sends it to Alice. Alice then computes $11^6(mod\ 17) = 8 = K_{ab}$ and Bob computes $15^7(mod\ 17) = 8 = K_{ab}$. Thus Alice and Bob have exchanged a shared symmetric key K_{ab} that they may use in any symmetric key algorithm they choose.

The elliptic curve group is structured in such a way that the Diffie-Hellman algorithm can be easily paralleled. In the next section we will simulate the Diffie-Hellman key exchange using the elliptic curve group over $\mathbb{F}_p$.

4.1.3 Elliptic Curve Key Exchange simulating Diffie-Hellman

Here we simulate the Diffie-Hellman key exchange using the elliptic curve group over $\mathbb{F}_p$. In this example Alice and Bob must first agree upon elliptic curve E, $\mathbb{F}_p$, and $P \in E_{(\alpha,\beta)}(\mathbb{F}_p)$. They must agree on $P \in E_{(\alpha,\beta)}(\mathbb{F}_p)$ such that $\langle P \rangle$ has a large enough order, so that the discrete logarithm of $E_{(\alpha,\beta)}(\mathbb{F}_p)$ is hard . In our case we will choose P such that $\langle P \rangle = E_{(\alpha,\beta)}(\mathbb{F}_p)$.

Let E be the curve $y^2 = x^3 + x + 1 \in \mathbb{F}_p[x]$, $\mathbb{F}_p$ be $\mathbb{Z}_{13}$, and cyclic subgroup $\langle(1,4)\rangle$. Then $|E(\mathbb{F}_p)| = 17 = |\langle(1,4)\rangle|$, this fact is illustrated in Table **??**. Alice chooses $a = 3$ and computes $A_k = 3 \cdot (1,4) = (0,12)$ and sends $A_k = (0,12)$ to Bob. Bob chooses $b = 4$ and computes $B_k = 4 \cdot (1,4) = (11,11)$ and sends $B_k = (11,11)$ to Alice. Alice then computes $B_k^3 = 3 \cdot (11,11) = (10,7) = K_{ab}$ and Bob computes $A_k^4 = 4 \cdot (0,12) = (10,7) = K_{ab}$. Alice and Bob then choose an algorithm to use this point to encrypt the message via some symmetric key algorithm. To discover either Alice, or Bob's private key, the party listening in would have to compute $log_{(1,4)}A_k$ or $log_{(1,4)}B_k$ respectively. This is precisely the discrete logarithm problem in $E(\mathbb{F}_p)$.

Points in $\langle(1,4)\rangle$

$\langle P\rangle$	nP	$(nP)^{-1}$
1 · (1, 4)	(1, 4)	(1, 9)
2 · (1, 4)	(8, 12)	(8, 1)
3 · (1, 4)	(0, 12)	(0, 1)
4 · (1, 4)	(11, 11)	(11, 3)
5 · (1, 4)	(5, 1)	(5, 12)
6 · (1, 4)	(10, 6)	(10, 7)
7 · (1, 4)	(12, 8)	(12, 5)
8 · (1, 4)	(4, 2)	(4, 11)
9 · (1, 4)	(7, 0)	(7, 0)
10 · (1, 4)	(4, 11)	(4, 2)
11 · (1, 4)	(12, 5)	(12, 8)
12 · (1, 4)	(10, 7)	(10, 6)
13 · (1, 4)	(5, 12)	(5, 1)
14 · (1, 4)	(11, 2)	(11, 11)
15 · (1, 4)	(0, 1)	(0, 12)
16 · (1, 4)	(8, 1)	(8, 12)
17 · (1, 4)	(1, 9)	(1, 4)

4.2 ElGamal Cryptosystem

The ElGamal cryptosystem, is based on the discrete logarithm problem and was named after its inventor Taher ElGamal. As in Diffie-Hellman, ElGamal relies on a very large prime p, and the primitive root α, of $(\mathbb{F}_p^*, \cdot)$. In ElGamal however, to compute the public key, an integer k is chosen and α^k is computed and (p, α, α^k) is publicly announced. The next section describes in detail the ElGamal algorithm.

4.2.1 ElGamal Key Generation

In this public key algorithm we assume that anyone listening has access to this information. Then key generation is as follows:

(1) Bob picks a large prime number p, and a primitive root α of $(\mathbb{F}_p, \cdot)$.

(2) Bob picks a random number $k \in \mathbb{Z}$ such that $2 \leq k \leq p-2$, and computes $\alpha^k \ (mod\ p)$.

(3) Bob sends (p, α, α^k) to Alice as his public key and keeps k private.

(4) Alice receives Bob's public key (p, α, α^g).

(5) Alice picks a random number $a \in \mathbb{Z}$ such that $1 < a < p-1$.

(6) Alice computes $\alpha^a \ (mod\ p)$, and $m_i \cdot (\alpha^k)^a \equiv m_i \cdot \alpha^{ak} \ (mod\ p)$, where m_i is l characters long.

(7) Alice sends ciphertext $c_i = (\alpha^a \ (mod\ p), m_i\alpha^{ak} \ (mod\ p))$.

(8) Bob receives C from Alice and computes $(\alpha^a)^{-k} \cdot m_i \cdot \alpha^{ak} \equiv m_i \cdot \alpha^{ak-ka} \equiv m_i \ (mod\ p)$

Then Bob's public key is (p, α, α^k), which can be used by Alice to encrypt her plaintext message.

4.2.2 The Elliptic Curve Cryptosystem Simulating ElGamal

As an example Alice would like to send the message $M = CIPHERTEXT$ to Bob. First Alice and Bob must agree on agree on a mapping scheme to map the character set, in this case the english alphabet, to the points on the elliptic curve. In our example we will map the powers of the group generator mod 26, so for example, if g were the generator of our group, then $g^4 = E$ as well as g^{30}. Now Bob chooses $p = 97$, so $F_p = \mathbb{Z}_{97}$ and curve $E = y^2 = x^3 + x + 1$ then $|E_{(1,1)}(\mathbb{Z}_{97})| = 97$.Then Bob chooses the point $(1, 10)$, since $|\langle(1, 10)\rangle| = 97$. Finally Bob chooses $k = 27$ and computes $27 \cdot (1, 10) = (84, 33)$. Bob sends his public key $(97,(1, 10),(84, 33))$ to Alice. Alice receives $(97,(1, 10),(84, 33))$ then splits up her messages $M = CIPHERTEXT$ into individual points on the curve, as shown in Figure 8. Now Alice chooses $a = 3$ since $1 < 3 < 95$, then computes

Plaintext	Power	Point
$C = 3$	$3 \cdot (1, 10)$	$(82, 43) = m_1$
$I = 8$	$8 \cdot (1, 10)$	$(36, 91) = m_2$
$P = 15$	$15 \cdot (1, 10)$	$(19, 34) = m_3$
$H = 7$	$7 \cdot (1, 10)$	$(38, 3) = m_4$
$E = 4$	$4 \cdot (1, 10)$	$(30, 68) = m_5$
$R = 18$	$18 \cdot (1, 10)$	$(95, 66) = m_6$
$T = 19$	$19 \cdot (1, 10)$	$(80, 42) = m_7$
$E = 4$	$4 \cdot (1, 10)$	$(30, 68) = m_8$
$X = 23$	$23 \cdot (1, 10)$	$(18, 15) = m_9$
$T = 19$	$19 \cdot (1, 10)$	$(80, 42) = m_{10}$

Figure 8: Message Encoding

$3 \cdot (1, 10) = (82, 43)$ and $3 \cdot (84, 33) = (68, 67)$ then enciphers

$$\begin{aligned}
c_1 &= (82, 43) + (68, 67) = (41, 53)\\
c_2 &= (36, 91) + (68, 67) = (81, 42)\\
c_3 &= (19, 34) + (68, 67) = (11, 81)\\
c_4 &= (38, 3) + (68, 67) = (20, 52)\\
c_5 &= (30, 68) + (68, 67) = (62, 73)\\
c_6 &= (95, 66) + (68, 67) = (36, 39)\\
c_7 &= (80, 42) + (68, 67) = (21, 12)\\
c_8 &= (30, 68) + (68, 67) = (62, 73)\\
c_9 &= (18, 15) + (68, 67) = (62, 2)\\
c_{10} &= (80, 42) + (68, 67) = (21, 12)
\end{aligned}$$

and sends the tuples $((82, 43), c_i)$ to Bob. Bob receives the tuples form Alice and computes $27 \cdot (82, 43) = (68, 67)$, and then $(68, 67)^{-1} = (68, 30)$ and deciphers by computing

$$\begin{aligned}
m_1 &= (41,53)+(68,30)=(82,43)=3\cdot(1,10)=C\\
m_2 &= (81,42)+(68,30)=(36,91)=8\cdot(1,10)=I\\
m_3 &= (11,81)+(68,30)=(19,34)=15\cdot(1,10)=P\\
m_4 &= (20,52)+(68,30)=(38,3)=7\cdot(1,10)=H\\
m_5 &= (62,73)+(68,30)=(30,68)=4\cdot(1,10)=E\\
m_6 &= (36,39)+(68,30)=(95,66)=18\cdot(1,10)=R\\
m_7 &= (21,12)+(68,30)=(80,42)=19\cdot(1,10)=T\\
m_8 &= (62,73)+(68,30)=(30,68)=4\cdot(1,10)=E\\
m_9 &= (62,2)+(68,30)=(18,15)=23\cdot(1,10)=X\\
m_10 &= (21,12)+(68,30)=(80,42)=19\cdot(1,10)=T,
\end{aligned}$$

thus retrieving Alice's original message $M = CIPHERTEXT$. This completes our example of the elliptic curve cryptosystem simulating the ElGamal algorithm.

5 Factorization

The discrete logarithm is not the only problem used in cryptography, factorization for example is also a very widely used. The problem of factoring a large number into prime factors is also very difficult. It turns out that computers operating at billions of operations per second could take billions of years to factor sufficiently large primes (depending on the algorithm used). Perhaps the most notable cryptosystem that relies on the factorization problem is the RSA cryptosystem.

5.1 RSA

The RSA algorithm was named after its inventors Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA uses the product of two large prime numbers, $n = p \cdot q$ for its security. The steps in the RSA algorithm for encrypting, and decrypting a message M between Bob and Alice are as follows:

(1) Bob calculates $n = p \cdot q$ where p and q are two large prime numbers, and Euler's Totent Function $\varphi(n) = (p-1)(q-1)$.

(2) Bob calculates l, the length of the message using the base b, to which the message will be encoded as, $b^x < n < b^y$. Then $l = x$ or y.

(3) Bob then picks a number e which is relatively prime to $\varphi(n)$.

(4) Finally, Bob picks a number d such that $d \equiv e^{-1}$ $(mod\ \varphi(n))$.

(5) Bob sends Alics (n, e) as his public key, and keeps d and $\varphi(n)$ private.

(6) Alice receives (n, e) and creates ciphertext $C = M^e$ $(mod\ n)$, sends C to Bob.

(7) Bob receives C from Alice and computes $M = C^d$ $(mod\ n)$, to recover Alice's original message M.

5.1.1 RSA Example

As an example take $M = CIPHERTEXT$, we will choose $p = 37$ and $q = 47$, then $n = p \cdot q = 37 \cdot 47 = 1739$, and $\varphi(n) = (36 \cdot 46) = 1656$. In this example we will convert the message to base 26, so to calculate l we see that $26^2 < 1739 < 26^3$ and we use $l = 3$. Then we choose $e = 55$ and check that $gcd(1656, 55) = 1$ so we may continue. Now to find d, we must solve $ed \equiv 1(\text{mod } n)$ so we use the Euclidean algorithm in reverse to solve $\varphi \cdot x + e \cdot d = 1$ and we see in fact that $(-9)1656 + (271)55 \equiv 1\ (mod\, n)$ thus $d = 271$. Now we convert $M = CIPHERTEXT$ to base 26, l characters at a time and pad the message with the letter A. Then $CIPHERTEXTAA$ becomes CIP - HER - TEX - TAA, and we enciphers as follows,

$$\begin{aligned} CIP &= 2 \cdot 26^2 + 8 \cdot 26 + 15 \equiv 1575\ (mod\ 1739) \\ HER &= 7 \cdot 26^2 + 4 \cdot 26 + 17 \equiv 1375\ (mod\ 1739) \\ TEX &= 19 \cdot 26^2 + 4 \cdot 26 + 23 \equiv 798\ (mod\ 1739) \\ TAA &= 19 \cdot 26^2 + 0 \cdot 26 + 0 \equiv 261\ (mod\ 1739) \end{aligned}$$

Now we compute $C = M^d\ (mod\ n)$,

$$1575^{55} \equiv 169\ (mod\ 1739)$$
$$1375^{55} \equiv 549\ (mod\ 1739)$$
$$198^{55} \equiv 798\ (mod\ 1739)$$
$$261^{55} \equiv 386\ (mod\ 1739),$$

and convert this back to text as,

$$169 = 0 \cdot 26^2 + 6 \cdot 26 + 13 = AGN$$
$$549 = 0 \cdot 26^2 + 21 \cdot 26 + 3 = AVD$$
$$798 = 1 \cdot 26^2 + 4 \cdot 26 + 18 = BES$$
$$386 = 0 \cdot 26^2 + 14 \cdot 26 + 4 = AOE,$$

and we see that $C = AGNAVDBESAOE$. To decrypt, $AGNAVDBESAOE$ becomes AGN-AVD-BES-AOE and decrypt as follows,

$$AGN = 0 \cdot 26^2 + 6 \cdot 26 + 13 \equiv 169\ (mod\ 1739)$$
$$AVD = 0 \cdot 26^2 + 21 \cdot 26 + 3 \equiv 549\ (mod\ 1739)$$
$$BES = 1 \cdot 26^2 + 4 \cdot 26 + 18 \equiv 798\ (mod\ 1739)$$
$$AOE = 0 \cdot 26^2 + 14 \cdot 26 + 4 \equiv 386\ (mod\ 1739),$$

then we compute $M = C^d\ (mod\ n)$,

$$169^{271} \equiv 1575\ (mod\ 1739)$$
$$549^{271} \equiv 1375\ (mod\ 1739)$$
$$798^{271} \equiv 198\ (mod\ 1739)$$
$$386^{271} \equiv 261\ (mod\ 1739),$$

and convert this back to text as,

$$1575 = 2 \cdot 26^2 + 8 \cdot 26 + 15 = CIP$$
$$1375 = 7 \cdot 26^2 + 4 \cdot 26 + 17 = HER$$
$$198 = 19 \cdot 26^2 + 4 \cdot 26 + 23 = TEX$$
$$261 = 19 \cdot 26^2 + 0 \cdot 26 + 0 = TAA.$$

Hence we arrive at our or original message $M = CIPHERTEXTAA$.

To discover the plaintext message, the party listening is would have to factor n, into $p \cdot q$ which, with sufficiently large p and q this is infeasible to do. There are however, algorithms that cut down on factoring time, in the next section I will describe Lenstra's elliptic curve factoring method.

5.2 Lenstra's Elliptic Curve Factoring Method

Definition 5.1. *Suppose that P_1, P_2 are points on $E(\mathbb{Q})$ where $P_1 + P_2 \neq \infty$ adn the denominators of P_1, P_2 are prime to n. Then $P_1 + P_2$ has coordinates having denominators prime to n if and only if there does not exist a prime $p \mid n$ such that*

$P_1 + P_2 = \infty\ (mod\ p)$ *on the elliptic curve* $E(\mathbb{Z}/n\mathbb{Z})$. *[2, p.]*

Lenstra's method of factorization using elliptic curves, is a factoring algorithm which has strong evidence of an expected running time of $\bigcirc(e^{\sqrt{(2+\epsilon)ln(p(ln(lnp)))}}ln(n)^2)$. Here p is the smallest prime factor of the number we are factoring n, and ϵ goes to zero as p gets sufficiently large. In this algorithm we take the elliptic curve group $E(\mathbb{Z}/n\mathbb{Z})$, which as you may notice is not over $\mathbb{F}_p$. It is this situation which is exploited from Definition 5.1 to find the prime factors of n. The algorithm for factoring using Lenstra's method is as follows:

(1) (Select an Elliptic Curve): Choose a random pair (E, P) where $E = E(\mathbb{Z}/n\mathbb{Z})$ is an elliptic curve:

$$y^2 = x^3 + \alpha\ x + \beta \text{ and } P \text{ is a point on } E$$

Check that $g = gcd(n, 4\alpha^3 + 27\beta^2) = 1$. If not, then we have split n if $1 < g < n$, and we may terminate the algotithm. Otherwise, we may select another (E, P) pair.

(2) Select $M \in \mathbb{N}$ and bounds $A, B \in \mathbb{N}$ such that the canonical prime factorization for M is $M = \prod_{j=1}^{l} p_j^{a_{p_j}}$ for small primes $p_1 < p_2 < ... < p_l \leq B$ where $a_{p_j} = \lfloor ln(A)/ln(p_j) \rfloor$ is the largest exponent such that $p_j \leq A$. Set $j = k = 1$.

(3) Use addition as described in Definition 2.1, to calculate p_jP

(4) (Calculating the gcd):

- If $p_jP \not\equiv \infty(mod\ n)$, then set $P = p_jP$, and set k to $k + 1$
  - (i) If $k \leq a_{p_j}$, then go to step 3.

- (ii) If $k > a_{p_j}$, then reset j to $j+1$, and reset k to $k = 1$, if $j < l$, then go to setp 3. Otherwise go to step 5.

- If $p_j P \equiv \infty (mod\ n)$, then compute $g = gcd(m_2, n)$ where m_2 is the denominator of the slope calculation. If $n > g$, terminate the algorithm, since we have split n. If $g = n$, go to step 5.

(5) (Selecting a new pair): Set $r = r - 1$. If $r > 0$, go to step (1). Otherwise, terminate with "failure". [2]

5.2.1 Example Lenstra's Method

Now we will attempt to split $n = 38411$, using Lenstra's Method. First we choose $(E, P) = (y^2 = x^3 + 2x + 9, (0, 3))$ and we check thats

$$gcd(38411, 4 \cdot 2^3 + 27 \cdot 9^2) = gcd(38411, 2219) = 1$$

so we may continue. Now we choose our bounds, so let $A = 3$, and $B = 4$, then $M = 6 = 2 \cdot 3$ and our exponents are,

$$a_{p_1} = \lfloor ln(A)/ln(p_1) \rfloor = \lfloor ln(3)/ln(2) \rfloor = 1$$
$$a_{p_2} = \lfloor ln(A)/ln(p_2) \rfloor = \lfloor ln(3)/ln(3) \rfloor = 1.$$

Now we calculate $p_1 P = 2(0, 3) \equiv (4268, 11378) \not\equiv \infty\ (mod\ n)$. Now we set $P = (4268, 11378)$, and compute $p_2 P \equiv 3P \equiv (26652, 211) \equiv \infty\ (mod\ n)$ so we must check $gcd(m_2, n) = gcd(5609, 38411) = 71$, thus we have split $n = 71 \cdot 541$.

References

[1] England, Matthew. *Elliptic Curve Cryptography.* Diss. Heriot-Wall University, Summer 2006. N.p.:n.p.n.d. Print.

[2] Forouzam, Behrouz A. *Introcuction to Cryptography and Network Security.* Boston: McGraw-Hill Higher Education, 2008. Print

[3] Mollin, Richard A. *An Introduction to Cryptography.* Boca Raton: Chapman & Hall/CRC, 2007. Print.

[4] ”3.6.1 What Is Diffie-Hellman?” RSA Laboratories. EMC Corporation, 2012. Web. 11 Aug. 2012. ¡http://www.rsa.com/rsalabs/node.asp?id=2248¿.

[5] Washington, Lawrence C. *Elliptic Curves: Number Theory adn Cryptography.* Boca Raton, FL: Chapman & Hall/CRC, 2008. Print.