$$(144749)88247^{-3632} \bmod 262643 = 41812$$
$$(5198)152432^{-3632} \bmod 262643 = 1111$$

Translating this into characters, this is PUP PIE SAR ESM ALL, or PUPPIESARESMALL, which was indeed what Bob sent.

The El Gamal cryptosystem provides strength comparable to other cryptosystems but uses a shorter key. It also introduces randomness into the cipher, so the same letter enciphered twice produces two different ciphertexts. This prevents attacks that depend upon repetition. However, care must be taken; if a random integer k is used twice, an attacker who obtains the plaintext for one message can easily decipher the other (see Exercise 10). Also, notice that c_2 is a linear function of m, so an attacker can forge messages that are multiples of previously enciphered messages. As an example, if (c_1, c_2) is the ciphertext of message m, (c_1, nc_2) is the ciphertext corresponding to nm. Protocols using El Gamal must prevent an attacker from being able to forge this type of message.

Network security protocols often use El Gamal due to its shorter key length. See Section 12.5.3 for an example. It can also be used for authentication (see Section 10.5.2.2).

10.3.2 RSA

The RSA cryptosystem was first described publicly in 1978 [1598]. Unknown at the time was the work of Clifford Cocks in 1973, where he developed a similar cryptosystem. This work was classified, and only became public in the late 1990s [629].

RSA is an exponentiation cipher. Choose two large prime numbers p and q, and let $n = pq$. The totient $\phi(n)$ of n is the number of numbers less than n with no factors in common with n. It can be shown that $\phi(n) = (p - 1)(q - 1)$ (see Exercise 12).

EXAMPLE: Let $n = 10$. The numbers that are less than 10 and are relatively prime to (have no factors in common with) n are 1, 3, 7, and 9. Hence, $\phi(10) = 4$. Similarly, if $n = 21$, the numbers that are relatively prime to n are 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, and 20. So $\phi(21) = 12$.

Choose an integer $e<n$ that is relatively prime to $\phi(n)$. Find a second integer d such that $ed \bmod \phi(n) = 1$. The public key is (e, n), and the private key is d.

Let m be a message. Then

$$c = m^e \bmod n$$

and

$$m = c^d \bmod n$$

Exercise 13 shows why this works.

When implementing this cipher, two issues are the computation of the modular exponentiation and finding two large primes. Exercise 19 shows how to compute the modular exponentiation quickly. Large prime numbers are found by generating large random numbers and then testing them for primality [291, 1607, 1826, 1955].

EXAMPLE: Let $p = 181$ and $q = 1451$. Then $n = 262631$ and $\phi(n) = 261000$. Alice chooses $e = 154993$, so her private key is $d = 95857$. As in the El Gamal example, Bob wants to send Alice the message "PUPPIESARESMALL," so he encodes it the same way, giving the plaintext 152015 150804 180017 041812 001111. Using Alice's public key, the ciphertext is

$$\begin{aligned}
152015^{154993} \bmod 262631 &= 220160 \\
150804^{154993} \bmod 262631 &= 135824 \\
180017^{154993} \bmod 262631 &= 252355 \\
041812^{154993} \bmod 262631 &= 245799 \\
001111^{154993} \bmod 262631 &= 070707
\end{aligned}$$

or 220160 135824 252355 245799 070707.

In addition to confidentiality, RSA can provide data and origin authentication; this is used in digital signatures (see Section 10.5.2.1). If Alice enciphers her message using her private key, anyone can read it, but if anyone alters it, the (altered) ciphertext cannot be deciphered correctly.

EXAMPLE: Suppose Alice wishes to send Bob the same message in such a way that Bob will be sure that Alice sent it. She enciphers the message with her private key and sends it to Bob. As indicated above, the plaintext is represented as 152015 150804 180017 041812 001111. Using Alice's private key, the ciphertext is

$$\begin{aligned}
152015^{95857} \bmod 262631 &= 072798 \\
150804^{95857} \bmod 262631 &= 259757 \\
180017^{95857} \bmod 262631 &= 256449 \\
041812^{95857} \bmod 262631 &= 089234 \\
001111^{95857} \bmod 262631 &= 037974
\end{aligned}$$

or 072798 259757 256449 089234 037974. In addition to origin authenticity, Bob can be sure that no letters were altered.

Providing both confidentiality and authentication requires enciphering with the sender's private key and the recipient's public key.

EXAMPLE: Suppose Alice wishes to send Bob the message "PUPPIESARESMALL" in confidence and authenticated. Again, assume that Alice's private key

is 95857. Take Bob's public key to be 45593 (making his private key 235457). The plaintext is represented as 152015 150804 180017 041812 001111. The encipherment is

$$(152015^{95857} \bmod 262631)^{45593} \bmod 262631 = 249123$$
$$(150804^{95857} \bmod 262631)^{45593} \bmod 262631 = 166008$$
$$(180017^{95857} \bmod 262631)^{45593} \bmod 262631 = 146608$$
$$(041812^{95857} \bmod 262631)^{45593} \bmod 262631 = 092311$$
$$(001111^{95857} \bmod 262631)^{45593} \bmod 262631 = 096768$$

or 249123 166008 146608 092311 096768.

The recipient uses the recipient's private key to decipher the message and the sender's public key to authenticate it. Bob receives the ciphertext above, 249123 166008 146608 092311 096768. The decipherment is

$$(249123^{235457} \bmod 262631)^{154993} \bmod 262631 = 152012$$
$$(166008^{235457} \bmod 262631)^{154993} \bmod 262631 = 150804$$
$$(146608^{235457} \bmod 262631)^{154993} \bmod 262631 = 180017$$
$$(092311^{235457} \bmod 262631)^{154993} \bmod 262631 = 041812$$
$$(096768^{235457} \bmod 262631)^{154993} \bmod 262631 = 001111$$

or 152015 150804 180017 041812 001111. This corresponds to the message Alice sent.

The use of a public key system provides a technical type of nonrepudiation of origin. The message is deciphered using Alice's public key. Because the public key is the inverse of the private key, only the private key could have enciphered the message. Because Alice is the only one who knows this private key, only she could have enciphered the message. The underlying assumption is that Alice's private key has not been compromised, and that the public key bearing her name really does belong to her.

In practice, no one would use blocks of the size presented here. The issue is that, even if n is very large, if one character per block is enciphered, RSA can be broken using the techniques used to break symmetric substitution ciphers (see Sections 10.2.2 and 12.1.3). Furthermore, although no individual block can be altered without detection (because the attacker presumably does not have access to the private key), an attacker can rearrange blocks and change the meaning of the message.

EXAMPLE: A general sends a message to headquarters asking if the attack is on. Headquarters replies with the message "ON" enciphered using an RSA cipher with a 2,048-bit modulus, but each letter is enciphered separately. An attacker

intercepts the message and swaps the order of the blocks. When the general deciphers the message, it will read "NO," the opposite of the original plaintext.

Moreover, if the attacker knows that headquarters will send one of two messages (here, "NO" or "ON"), the attacker can use a technique called "forward search" or "precomputation" to break the cipher (see Section 12.1.1). For this reason, plaintext is usually padded with random data to make up a block. This can eliminate the problem of forward searching, because the set of possible plaintexts becomes too large to precompute feasibly.

A different general sends the same request as in the example above. Again, headquarters replies with the message "ON" enciphered using an RSA cipher with a 2,048-bit modulus. Each letter is enciphered separately, but the first 10 bits of each block contain the number of the block, the next 8 bits contain the character, and the remaining 2,030 bits contain random data. If the attacker rearranges the blocks, the general will detect that block 2 arrived before block 1 (as a result of the number in the first 10 bits) and rearrange them. The attacker also cannot precompute the blocks to determine which contains "O," because she would have to compute 2^{2030} blocks, which is computationally infeasible.

10.3.3 Elliptic Curve Ciphers

Miller [1351] and Koblitz [1082] proposed a public key scheme based on *elliptic curves*. This scheme can be applied to any scheme that depends on the discrete logarithm problem. Here, we show a version of El Gamal using elliptic cryptography.

Definition 10–3. An *elliptic curve* is an equation of the form $y^2 = x^3 + ax + b$.

Figure 10–5 shows the plot of the curve $y^2 = x^3 + 4x + 10$. Consider two points on the curve, P_1 and P_2. If $P_1 \neq P_2$, draw a line through them. If $P_1 = P_2$, then draw a tangent to the curve at P_1. Suppose that line intersects the curve at a third point, $P_3 = (x_3, y_3)$. Take $P_4 = (x_3, -y_3)$. We define P_4 to be the sum of P_1 and P_2. Otherwise, the line is vertical, so take $P_1 = (x, y)$ and treat ∞ as another point of intersection with the curve. The third point of intersection is $P_2 = (x, -y)$, so given the above definition of addition, we have $P_1 + \infty = (x, y) = P_1$. Hence the point at ∞ is the identity in addition. It is also its own inverse.

More precisely, let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Define

$$m = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{when } P_1 \neq P_2 \\ \dfrac{3x_1^2 + a}{2y_1} & \text{otherwise} \end{cases}$$

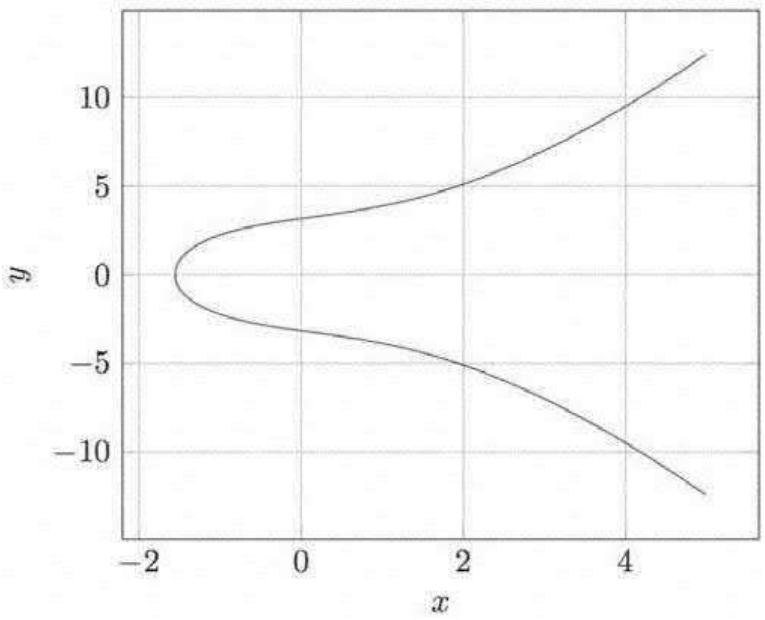

Figure 10–5 Plot of the elliptic curve $y^2 = x^3 + 4x + 10$.

Then $P_3 = P_1 + P_2$, where

$$x_3 = m^2 - x_1 - x_2$$
$$y_3 = m(x_1 - x_3) - y_1$$

Also, if $P_4 = -P_3$, then $x_4 = x_3$ and $y_4 = -y_3$.

This can be turned into a cryptosystem using modular arithmetic, where the modulus used is a prime number p. Thus, the curve of interest is of the form

$$y^2 = x^3 + ax + b \bmod p$$

with p a prime number and $4a^3 + 27b^2 \neq 0$.[3] Suppose we add a point P to itself n times. Call the result Q, so $Q = nP$. If n is large, it is generally very hard to compute from Q and P. This is the basis for the security of the cryptosystem.

Thus, an elliptic curve cryptosystem has four parameters: (a, b, p, P). The private key is a randomly chosen integer $k<p$; in practice, one chooses this number to be less than the number of (integer) points on the curve. The corresponding

[3]More generally, elliptic curves can be over any finite field. When the size of the finite field is a power of 2, the equation has the form $y^2+xy = x^3+ax^2+b$; the rules for addition are also slightly different.

public key is $K = kP$. In the following examples, we shall use the shorthand $(x, y) \bmod p$ to mean $(x \bmod p, y \bmod p)$. Also, $a^{-1} \bmod p$ is the value x that satisfies the equation $ax \bmod p = 1$ (see Section B.3).

To use the elliptic curve version of El Gamal, choose a point P on the curve, and a private key k_{priv}. Then compute $Q = k_{priv}P$. Using the elliptic curve above, this means that the public key is (P, Q, a, p). To encipher a message m, it is first expressed as a point on the elliptic curve. The sender then selects a random number k and computes

$$c_1 = kP$$
$$c_2 = m + kQ$$

and sends those to the recipient. To decipher the message, the recipient computes

$$m = c_2 - k_{priv}c_1$$

EXAMPLE: Alice and Bob now decide to use the elliptic curve version of El Gamal to encipher their messages. They use the same elliptic curve and point as in the previous example. Bob chooses a random number $k_{Bob} = 1847$ as his private key. He then computes his public key $K_{Bob} = k_{Bob}P = 1847(1002, 493) \bmod 2503 = (460, 2083)$.

Alice wants to send Bob the message $m = (18, 1394)$. To encipher it, she chooses a random number $k = 717$, computes

$$c_1 = kP = 717(1002, 493) \bmod 2503 = (2134, 419)$$
$$c_2 = m + kK_{Bob} = (18, 1394) + 717(460, 2083) \bmod 2503 = (221, 1253)$$

and sends c_1 and c_2 to Bob.

To decipher the message, Bob computes

$$k_{Bob}c_1 = 1847(2134, 419) \bmod 2503 = (652, 1943)$$

and uses this to compute

$$m = c_2 - c_1 = (221, 1253) - (652, 1943) \bmod 2503 = (18, 1394)$$

thereby recovering the plaintext message.

The generation of elliptic curves suitable for cryptography is a complex question. In particular, it requires a careful selection of parameters. For example, when $b = 0$ and $p \bmod 4 = 3$, or when $a = 0$ and $p \bmod 3 = 2$, the discrete log problem underlying elliptic curve cryptography becomes significantly easier

to solve. Thus, choosing these parameters weakens the cryptosystem. Ways to generate elliptic curves are being studied [440]. Several parameter sets have been recommended for use. The U.S. NIST recommends curves P-192, P-224, P-256, P-384, or P-521 for elliptic curves using a prime modulus, and degree 163, 233, 283, 409, or 571 binary fields [2148]. Certicom recommends these as well, except that the degree 233 binary field is replaced by a degree 239 binary field [2231]. These curves are widely used. Some questions have been raised about the strength of these curves [181]. Other proposed curves include those of the Brainpool standard [1203], Curve1174 [182], Curve25519 [180], and several others [271].

The advantage to using elliptic curves over other forms of public key cryptography is that the keys can be shorter, and hence the computation time is shorter. As an example, elliptic curve cryptography with a key length of 256 to 383 bits provides a level of security comparable to RSA with a modulus of 3,072 bits [126]. Koblitz, Koblitz, and Menezes [1081] review how elliptic curve cryptography became widely accepted, with a discussion of it and RSA.

10.4 Cryptographic Checksums

Suppose Alice wants to send Bob a message of n bits. She wants Bob to be able to verify that the message he receives is the same one that was sent. So she applies a mathematical function, called a *checksum function*, to generate a smaller set of k bits from the original n bits. This smaller set is called the *checksum* or *message digest*. Alice then sends Bob both the message and the associated checksum. When Bob gets the message, he recomputes the checksum and compares it with the one Alice sent. If they match, he assumes that the message has not been changed; if they do not match, then either the message or the checksum has changed, and so they cannot be trusted to be what Alice sent him.

Of course, an adversary can change the message and alter the checksum to correspond to the message. For the moment, assume this will not happen; we will relax this assumption in Section 10.5.

EXAMPLE: The parity bit in the ASCII representation is often used as a single-bit checksum. If *odd parity* is used, the sum of the 1 bits in the ASCII representation of the character, and the parity bit, is odd. Assume that Alice sends Bob the letter "A." In ASCII, the representation of "A" using odd parity is $p1000001$ in binary, where p represents the parity bit. Because two bits are set, the parity bit is 1 for odd parity.

When Bob gets the message 11000001, he counts the 1 bits in the message. Because this number is odd, Bob believes that the message has arrived unchanged.