RSA Encryption

Each of the encryption schemes we have seen so far has the same fundamental problem: how do we securely distribute private keys so that the encryption scheme is secure? In this section of the notes we will look at key exchange principles, along with a type of public key cryptography known as RSA. These notes are very similar to those which will be in the lecture slides.

Key Exchange Algorithms

Suppose that Alice and Bob want to securely communicate using AES-128, but they are having trouble with securely agreeing a shared private key. A solution to this problem was originally proposed by Ralph Merkle, called **public key exchange**. The idea was communicated by Whitfield Diffie and Martin Hellman in 1976. However, in 1997 the British Government declassified documents pertaining to public key exchange protocols and it turned out that James Ellis had had similar thoughts about key exchange, which were further developed by Clifford Cocks and Malcolm Williamson while they worked at GCHQ.

Many algorithms in this area are really about key *agreement*, rather than key *exchange*, but it is usually understood what is meant.

Diffie–Hellman

Before describing how Diffie and Hellman articulated a version of Merkle's ideas, we need the following definition.

Definition. Let p be a prime and let $\alpha \in \{2, 3, \ldots, p-1\}$. We say that α is a **generator** (modulo p) if it has the property that

$$\alpha^{p-1} = 1 \mod p$$

and every number $n \in \{1, 2, 3, \ldots, p-1\}$ can be written as $n = \alpha^a$ for some integer value of a.

For example, taking the prime $p = 7$ we can show that $\alpha = 3$ is a generator:

$$\begin{aligned}
3^1 &= 3 \bmod 7, \\
3^2 &= 9 = 2 \bmod 7, \\
3^3 &= 3 \times 2 = 6 \bmod 7, \\
3^4 &= 3 \times 6 = 18 = 4 \bmod 7,
\end{aligned}$$

$$3^5 = 3 \times 4 = 12 = 5 \bmod 7,$$
$$3^6 = 3 \times 5 = 15 = 1 \bmod 7$$

We can now describe what is known as **Diffie–Hellman key exchange**:

1. Alice and Bob agree on a prime number p and a generator α.
2. Alice secretly chooses a number x and Bob secretly chooses a number y.
3. Alice calculates $\alpha^x \mod p$ and Bob calculates $\alpha^y \mod p$: these values are then exchanged.
4. Alice calculates $(\alpha^y)^x \mod p$ and Bob calculates $(\alpha^x)^y \mod p$.

At this point both Alice and Bob have now calculated $\alpha^{xy} \mod p$, so they both share the same value as each other, without knowing what the other party chose as their integer values (x or y). The following examples demonstrates this.

- Suppose that Alice and Bob agree to use the prime $p = 7$ and the generator $\alpha = 3$.
- Now suppose Alice chooses $x = 2$ and Bob chooses $y = 5$.
- Alice calculates $\alpha^x = 3^2 = 9 = 2 \bmod 7$ and Bob calculates $\alpha^y = 3^5 = 243 = 5 \bmod 7$.
- Alice and Bob exchange these values.
- Alice finds
$$(\alpha^y)^x = 5^2 = 25 = 4 \bmod 7.$$
- Bob finds
$$(\alpha^x)^y = 2^5 = 32 = 4 \bmod 7.$$
- Alice and Bob now share a secret value and can use this to generate their private key.

This idea can also be extended if more than two parties need to know the private key.

Further Algorithms

There are a number of other key exchange algorithms, which we do not study here. Some examples are given below:

- Station-to-Station Protocol;
- Shamir's Three Pass Protocol;

RSA: Development

Key exchange was a great innovation, but another idea about public key cryptography was quick to follow: What if we let everybody know the encryption key, but kept another key secret which was only used for decryption? Various cryptographers had this idea and knew it would rely on finding a mathematical function which was **one way**: many inputs would produce the same output.

RSA was first publicly described in 1977: United States Patent US4405829A. The authors were Ron Rivest, Adi Shamir, and Len Adleman, working at MIT at the time. (An equivalent method was developed by Clifford Cocks in 1973, again at GCHQ, but this wasn't declassified until 1997.) The scheme depends on large prime numbers, modular arithmetic, and the inherent difficulty in factorising large products.

Before describing the RSA encryption scheme, we need to know a little more about the theory of numbers.

Some Number Theory

Let p be a **prime number**: a number which is divisible only by itself and by 1. We do not include the number 1 in this definition. The first few primes are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, ... An integer a is **coprime** to p if the **highest common factor** of a and p is 1: $\text{hcf}(a, p) = 1$. This is also known as the **greatest common divisor**. E.g., $\text{hcf}(105, 195) = 15$ since $105 = 3 \times 5 \times 7$ and $195 = 3 \times 5 \times 13$.

Definition. Let n be a non-negative integer. The number of integers less than n which are coprime to n is written as $\varphi(n)$. The function φ is known as **Euler's totient function**.

For example, $\varphi(6) = 2$ since only 1 and 5 are less than 6 and coprime to 6. Also $\varphi(10) = 4$: 1, 3, 7, and 9 are coprime to 10. This is easy to compute for prime numbers: $\varphi(p) = p-1$. And for products of primes: $\varphi(pq) = (p-1)(q-1)$.

RSA: Setup

- Alice chooses two 'large' primes p and q.
- She calculates $n = pq$ (the **modulus**) and then finds $\varphi(n) = \varphi(pq) = (p-1)(q-1)$.
- Alice also selects a value of e (the **encryption exponent**) such that $1 < e < \varphi(n)$ and $\text{hcf}(e, \varphi(n)) = 1$.
- She also finds a value d (the **decryption exponent**) such that $de = 1 \bmod \varphi(n)$. (More on this later.)
- Alice then publishes (n, e) publicly: this is her **public key**.
- She keeps the value of d secret: this is her **private key**.

RSA: Encryption

- Alice has published her public key (n, e) and kept her private key d secret.
- Bob can now encrypt messages to send to Alice. He selects a message value $0 \leq m < n$ and calculates $c = m^e \bmod n$. He sends this to Alice.
- Alice can now decrypt this by finding $c^d = m^{de} = m \bmod n$.

That all of this works relies on a result in number theory called Euler's Theorem:

Theorem. Let n be a non-negative integer and let a be an integer coprime to n. Then

$$a^{\varphi(n)} = 1 \bmod n.$$

Below we see an example of how this is used for specific values.

- Suppose that Alice has chosen primes $p = 7$ and $q = 13$, and **encryption exponent** $e = 5$.
- Her **modulus** is $n = pq = 7 \times 13 = 91$.
- She also calculates $\varphi(n) = \varphi(91) = \varphi(7 \times 13) = 6 \times 12 = 72$.
- Alice finds $d = 29$, so that $de = 29 \times 5 = 145 = 1 \bmod 72$.
- She publishes the public key $(n, e) = (91, 5)$.
- Bob wants to send the message '`Y`' to Alice. He could convert this to ASCII (0101 1001 in binary, 89 in denary) so that the message value is $m = 89$.
- Bob calculates $c = m^e = 89^5 \bmod 91$. He might do this by repeated squaring if the powers are large. (See previous notes on modular arithmetic.)
- $c = m^e = 89^5 = (-2)^5 = -32 = 59 \bmod 91$
- Alice decrypts this by calculating $c^d = 59^{29} \bmod 91$.

RSA: Decryption Exponent

When Alice sets up her RSA scheme, she chooses large primes p and q. Using these she calculates $n = pq$ and $\varphi(n) = (p-1)(q-1)$, but she also requires a value e such that $1 < e < \varphi(n)$ with $\text{hcf}(e, \varphi(n)) = 1$ and a value d such that $de = 1 \bmod \varphi(n)$. This can be hard to find by trial and error, so we can introduce the **Euclidean Algorithm**.

For example, how could we easily check that the highest common factor of $e = 5$ and $\varphi(n) = 72$ is actually equal to 1?

$$\begin{aligned} 72 &= 14 \times 5 + 2 \\ 5 &= 2 \times 2 + 1 \\ 2 &= 2 \times 1 + 0 \end{aligned}$$

The least number before 0 is the highest common factor. Here this is 1, so e and $\varphi(n)$ are coprime.

Euclidean Algorithm

The RSA example used the value of $d = 29$, but didn't explain where this came from. At each stage, the pair of coordinates (a, b) corresponding to a number

	$\varphi(n)$		e	
$(1, 0)$	72		5	$(0, 1)$
		$\times 14$		
$(0, 14)$	70		4	$(2, -28)$
		$\times 2$		
$(1, -14)$	2		1	$(-2, 29)$
		$\times 2$		
$(-4, 58)$	2			
$(5, -72)$	0			

c in the table tells us that $72a + 5b = c$. E.g., the pair $(2, -28)$ corresponds to the number 4 in the table, so we know that $72 \times 2 + 5 \times (-28) = 4$. Similarly, we know that $72 \times (-2) + 5 \times 29 = 1$.

We need a value of d such that $de = 1 \bmod \varphi(n)$, so in our case such that $5d = 1 \bmod 72$. Now we know that $72 \times (-2) + 5 \times 29 = 1$, so $5 \times 29 = 1 \bmod 72$. Hence $d = 29$.

RSA: Examples

Alice changes her RSA scheme to use $p = 11$, $q = 19$, and $e = 7$. She calculates $n = pq = 11 \times 19 = 209$ and $\varphi(n) = (p-1)(q-1) = 10 \times 18 = 180$. She publishes $(n, e) = (209, 7)$ and calculates d using the extended Euclidean algorithm.

	$\varphi(n)$		e	
$(1, 0)$	180		7	$(0, 1)$
		$\times 25$		
$(0, 25)$	175		5	$(1, -25)$
		$\times 1$		
$(1, -25)$	5		2	$(-1, 26)$
		$\times 2$		
$(-2, 52)$	4			
$(3, -77)$	1			

The algorithm suggests that $d = -77$, but we want d to be positive. Since we want $de = 1 \bmod \varphi(n)$ and $\varphi(n) = 180$, we can add 180 to -77 to find that

$$d = -77 = -77 + 180 = 103 \bmod 180.$$

Concept Checks

Test Yourself Visit the URL below to try a numbas exam:
https://numbas.mathcentre.ac.uk/question/155037/rsa-encryption/embed

RSA in Practice

The main idea behind RSA is that your encryption key (n, e) can be made **public**: anybody can know it without any security issues. But, only you have access to the decryption key (n, d). Theoretically, an attacker could factor the modulus n and use the primes to calculate the decryption exponent d. The security of RSA depends on this being a very difficult problem.

Our examples only include small primes so that we can understand how RSA works, but in practice the modulus will be 2048 or 3072 bits as recommended by NIST in 2015. This means that the modulus will be of the order of 600+ decimal digits. In general, RSA is not recommended for encrypting whole messages as it is computationally impractical. Instead it finds most of its use in key distribution (alongside Diffie-Hellman key exchange) and digital verification/authentication schemes.