Ciphers Where Alice and Bob do NOT need to Meet

Based on notes by William Gasarch

1 Our Goal

The following problem plagues all of the systems we have considered: Alice and Bob must meet in secret to establish a key.

Is there a way around this? Is there a way for Alice and Bob to NEVER meet, and yet establish a secret key? That is, can they, by talking *in public* establish a shared secret key?

The answer will be yes, assuming that whoever is listening in has some limits on what they can compute.

2 Needed Math

We'll use multiplication modulo p in the set $Z_p = \{1, 2, \ldots, p-1\}$, where p is a prime number. It will be useful to find an element $g \in Z_p$, called a "generator", for which the sequence $g^0, g^1, g^2, \ldots, g^{p-2}$, taken modulo p, contains all of the elements of Z_p.

Let's look at $p = 11$. Notice that

$$\begin{aligned}
2^0 &\equiv 1 \bmod 11 \\
2^1 &\equiv 2 \bmod 11 \\
2^2 &\equiv 4 \bmod 11 \\
2^3 &\equiv 8 \bmod 11 \\
2^4 &\equiv 5 \bmod 11 \\
2^5 &\equiv 10 \bmod 11 \\
2^6 &\equiv 9 \bmod 11 \\
2^7 &\equiv 7 \bmod 11 \\
2^8 &\equiv 3 \bmod 11 \\
2^9 &\equiv 6 \bmod 11
\end{aligned}$$

These calculations are not hard if you use that $2^n \equiv 2 \times 2^{n-1} \bmod 11$. Notice that $\{2^0 \bmod 11, 2^1 \bmod 11, \ldots, 2^9 \bmod 11\} = \{1, 2, \ldots, 10\}$.

Do all elements of Z_{11} generate the entire set? No:

$$
\begin{aligned}
5^0 &\equiv 1 \bmod 11\\
5^1 &\equiv 5 \bmod 11\\
5^2 &\equiv 3 \bmod 11\\
5^3 &\equiv 4 \bmod 11\\
5^4 &\equiv 9 \bmod 11\\
5^5 &\equiv 1 \bmod 11\\
5^6 &\equiv 5 \bmod 11\\
5^7 &\equiv 3 \bmod 11\\
5^8 &\equiv 4 \bmod 11\\
5^9 &\equiv 9 \bmod 11
\end{aligned}
$$

Notice that $\{5^0 \bmod 11, 5^1 \bmod 11, \ldots, 5^9 \bmod 11\} = \{1, 3, 4, 5, 9\}$. This is NOT all of Z_{11}.

Convention 2.1 We will be using a prime p. We will assume that p is LARGE but that $\log p$ is not too large. Hence if Eve needs a computation of p steps to crack a code we will consider it a good code. Even if Eve needs a computation of $\sqrt{p}$ steps (or p^ϵ steps where $\epsilon > 0$) this is a long time and we will consider it a good code. Also, if Alice and Bob have to do operations that take $\log p$ steps, that's okay, they can do that. Even if they have to take $(\log p)^2$ (or some larger polynomial in $\log p$) thats okay, they can do that.

Convention 2.2 For the rest of this document when we say "roughly p" we will mean p^ϵ for some ϵ, $\epsilon > 0$. When we say "roughly $\log p$" we will mean $(\log p)^a$ for some $a \in N$.

Theorem 2.3 *For every prime p there is a g such that $\{g^0 \bmod p, g^1 \bmod p, \ldots, g^{p-2} \bmod p\} = Z_p = \{1, \ldots, p-1\}$. There is an algorithm which will, given p, find such a generator g in roughly $\log p$ steps.*

We have already seen that $+, -, \times$, and (if p is prime) division can be done modulo p. We now have a way to do LOGARITHMS modulo p.

Definition 2.4 Let p be a prime and g be a generator of Z_p. Let $x \in Z_p$. The *Discrete Logarithm of x with base g* is the $y \in \{0, \ldots, p-2\}$ such that $g^y \equiv x \bmod p$. We denote this $DL_g(x)$.

Example 2.5 We rewrite the table above for $p = 11$ and add to it. The Discrete Logarithm lines follow from the prior line. We assume $g = 2$ and denote DL_2 by just DL.

$$\begin{aligned}
2^0 &\equiv 1 \bmod 11 \\
DL(1) &= 0 \\[1em]
2^1 &\equiv 2 \bmod 11 \\
DL(2) &= 1 \\[1em]
2^2 &\equiv 4 \bmod 11 \\
DL(4) &= 2 \\[1em]
2^3 &\equiv 8 \bmod 11 \\
DL(8) &= 3 \\[1em]
2^4 &\equiv 5 \bmod 11 \\
DL(5) &= 4 \\[1em]
2^5 &\equiv 10 \bmod 11 \\
DL(10) &= 5 \\[1em]
2^6 &\equiv 9 \bmod 11 \\
DL(9) &= 6 \\[1em]
2^7 &\equiv 7 \bmod 11 \\
DL(7) &= 7 \\[1em]
2^8 &\equiv 3 \bmod 11 \\
DL(3) &= 8 \\[1em]
2^9 &\equiv 6 \bmod 11 \\
DL(6) &= 9
\end{aligned}$$

COMMON BELIEF: It is believed that the problem of computing the discrete logarithm *requires* roughly p steps. This is a long time, so we assume Eve cannot do this.

Fact 2.6

1. *Given p, finding a generator for Z_p can be done in roughly* $\log p$ *steps.*
2. *Given L, finding a prime of size around L can be done in roughtly* $\log L$ *steps.*
3. *Given p, $a \in \{0, 1, \ldots, p-1\}$, and m, determining $a^m \bmod p$ takes roughly* $\log m$ *steps. (This is by repeated squaring.)*

3 Diffie Helman Key Exchange

We can USE this mathematics to have Alice and Bob exchange information in public and in the end they have a shared secret key.

1. Alice generates a large prime p and a generator g (this takes roughly $\log p$ steps) and sends it to Bob over an open channel. So now Alice and Bob know p, g but so does Eve.

2. Alice generates a random $a \in \{0, \ldots, p-2\}$. Bob generates a random $b \in \{0, \ldots, p-2\}$. They keep these numbers private. Note that even Alice does not know b, and even Bob does not know a.

3. Alice computes $g^a \bmod p$. Bob computes $g^b \bmod p$. Both use repeated squaring so it takes roughly $\log p$.

4. Alice sends Bob $g^a \bmod p$ over an open channel. Notice that Eve will NOT be able to compute a if computing DL_g is hard (which is the common belief). Even Bob won't know what a is.

5. Bob sends Alice $g^b \bmod p$. Notice that Eve will NOT be able to compute b if computing DL_g is hard. Even Alice won't know what b is.

6. RECAP: Alice now has a and g^b. SHE DOES NOT HAVE b. Bob has b and g^a. HE DOES NOT HAVE a. Eve has g^a and b^b. SHE DOES NOT HAVE a OR b.

7. Alice computes $(g^b)^a \bmod p = g^{ab} \bmod p$. Bob computes $(g^a)^b \bmod p = g^{ab} \bmod p$. They both use repeated squaring so this is fast.

8. SO at the end of the protocol they BOTH know $g^{ab} \bmod p$. This is their shared secret key. Eve likely does NOT know g^{ab} since she only gets to see g^a and g^b.

This scheme LOOKS good but we must be very careful about what is known about it.

1. Alice and Bob can execute the scheme quickly.

2. If Eve can compute DL_g quickly then she can crack the code.

3. There MIGHT BE other ways for Eve to crack the code. That is, being able to compute DL_g quickly is sufficient to crack this scheme, but might not be neccesary.

4. This scheme can be used for Alice and Bob to establish a secret key without meeting. This can then be used in other schemes such as the one-time pad.

5. Reality: This scheme is used in the real world for secret key exchange. The RSA algorithm is used for Public Key Cryptography (which is similar).