

21: BIT COMMITMENT

Alice and Bob agree to decide some matter by tossing a coin. Alice tosses the coin and Bob calls. Suppose, however, they are in different rooms and so Bob can not see Alice tossing the coin. If they do not trust one another then they need a way to ensure that Bob does not change his call when he hears the result, nor Alice change the reported result when she knows Bob's guess.

One way to do this is for Bob to write down his guess and put it in a sealed envelope, which he gives to Alice. Then Alice tosses the coin and reports the result. Together they open the envelope and see if Bob's guess was correct. This is called a *commit and reveal* process. Bob commits his guess to paper and then, after Alice has announced the result, he reveals it to Alice.

Similar problems arise when sending messages. For example, if someone is selling racing tips but needs to ensure that he is paid before revealing the tip. Or if a poll is being conducted online where the results will be revealed to all participants but only after everyone has voted. We can use a similar commit and reveal process. It is usually called *bit commitment*. Bob chooses a bit 0 or 1 and commits this to a message sent to Alice. However the message is sent in such a way that Alice can not read it without further information and yet Bob can not alter his choice. There are a variety of methods for achieving this.

21.1 Bit Commitment using a Public Key Cipher

Let e_B and d_B be the encrypting and decrypting functions for a public key cipher used by Bob. The encrypting function e_B is published but the decrypting function d_B is kept secret by Bob. Now Bob makes his choice $m \in \mathbb{F}_2$ and commits to Alice the encrypted message $c = e_B(m)$. Provided that the cipher is secure, Alice can not decipher this. When the time comes to reveal his choice, Bob sends to Alice his private key so that she can find the decrypting function d_B . Now she can compute

$$d_B(c) = d_B(e_B(m)) = m$$

and find out what Bob's choice was. She can also check that d_B and e_B are inverse functions so she can be confident that Bob has not sent the wrong private key so as to pretend his choice was different.

21.2 Bit Commitment using a Noisy Channel

We can also use our earlier work on noisy channels to give an alternative method for bit commitment. Suppose that Alice and Bob have two ways to communicate. They can use a clear channel or a noisy channel. The clear channel transmits without errors but the noisy channel is a binary symmetric channel with error probability $0 < p < \frac{1}{2}$. We assume that the noisy channel corrupts bits independently of any action by Alice or Bob, so neither can affect its behaviour.

Now Alice chooses a linear binary code with codebook $C \subset \mathbb{F}_2^N$ and minimum distance d . Bob chooses a random, non-trivial, linear map $\theta : C \rightarrow \mathbb{F}_2$. Both publish their choices. In order to send a bit $m \in \mathbb{F}_2$, Bob chooses a random code word $\mathbf{c} \in C$ with $\theta(\mathbf{c}) = m$ and sends this to Alice via the noisy channel. So Alice receives a vector $\mathbf{r} = \mathbf{c} + \mathbf{e}$ in which certain components of \mathbf{c} have been altered.

The expected value for $d(\mathbf{e}, \mathbf{0})$ is Np and N is chosen so that this is very much larger than d . This means that Alice can not tell what the original codeword \mathbf{c} was and hence can not find $\theta(\mathbf{c}) = m$ and determine Bob's choice.

When the time comes for Bob to reveal his choice, he sends the codeword \mathbf{c} to Alice via the clear channel. Alice can now check that $d(\mathbf{c}, \mathbf{r})$ is close to the expected value Np . If it is, she accepts that $\theta(\mathbf{c})$ was Bob's choice. If not, then she rejects it. There is, of course, a small chance that many more or many fewer bits of \mathbf{c} were corrupted by the noisy channel and so Alice rejects Bob's choice even though he has not cheated. We choose the parameters N, d so that the probability of this occurring is very small. If it does occur, Alice and Bob should just repeat the entire process.

We have seen that Alice can not read Bob's guess until he has revealed it. We also want to show that Bob can not cheat by changing his guess and sending a different code word $\mathbf{c}' \neq \mathbf{c}$ to Alice via the clear channel. Bob knows the code word \mathbf{c} that he originally chose but he does not know how it was corrupted by the noisy channel. So, all he knows is that the vector \mathbf{r} received by Alice is at a Hamming distance of about Np from \mathbf{c} . If he sends \mathbf{c}' , then he must ensure that $d(\mathbf{r}, \mathbf{c}') \approx Np$. The probability that this happens is small unless he chooses \mathbf{c}' very close to \mathbf{c} . However, any two codewords are at least distance d apart, so we see that Bob can not cheat.

21.3 Semantic Security

We have shown how to produce ciphers that are secure in the sense that an enemy who knows the ciphertext can not find the corresponding plaintext in polynomial time. However, we have not considered whether the enemy can, nonetheless, obtain some partial information about the plaintext.

When this too is impossible, then the cipher is *semantically secure*. This means, in particular, that it is impossible to decipher even a single bit of the ciphertext.

Semantically secure ciphers have been produced in recent years by encrypting the plaintext randomly. So, rather than replacing a single letter $a \in \mathcal{A}$ by a fixed codeword $c(a)$, we choose from a large set of possible codewords for a . This makes it very much harder to find a and hence to break the cipher.

Goldwasser and Micali gave an example of such a scheme that uses quadratic residues. Recall that a number k with $(k, N) = 1$ is a *quadratic residue modulo N* if $k \equiv x^2 \pmod{N}$ for some x . It is a *quadratic non-residue modulo N* when there is no such x . For a prime p , we know that there is a primitive root γ modulo N , so k is a quadratic residue modulo p when $k = \gamma^{2r}$ for some integer r . This gives Euler's criterion: k is a quadratic residue modulo the prime p if and only if $k^{(p-1)/2} \equiv 1 \pmod{p}$. Now consider $N = pq$ the product of two distinct primes, p and q . The Chinese remainder theorem shows that k is a quadratic residue modulo N if and only if it is a quadratic residue modulo p and modulo q . This means that it is simple to determine whether k is a quadratic residue provided that we know how to factorize N .

Bob wishes to send to Alice a binary message $m = a_1 a_2 \dots a_K$ where each $a_j \in \mathbb{F}_2$. First Alice needs to set a key. She selects two different large primes p, q and sets $N = pq$. Then she chooses a random integer y modulo N with y a quadratic non-residue modulo N . It is easy to find such a y since Alice knows the factors of N . She publishes (N, y) as her public key.

Now Bob encrypts his message m as $e(a_1)e(a_2)\dots e(a_K)$ where

$$e(a_j) \equiv \begin{cases} x_j^2 \pmod{N} & \text{when } a_j = 0 \\ yx_j^2 \pmod{N} & \text{when } a_j = 1 \end{cases} .$$

Here x_j is coprime to N and is chosen randomly for each letter a_j . Note that the length of the ciphertext is very much longer than that of the plaintext, since each bit of m is encrypted as a string of $\log_2 N$ bits.

To decrypt the ciphertext, we need to decide if each letter $c_j = e(a_j)$ is a quadratic residue modulo N , in which case $a_j = 0$, or a quadratic non-residue, in which case $a_j = 1$. This is simple for Alice since she knows the factors p and q of N . Using Euler's criterion she can determine whether c_j is a quadratic residue or non-residue modulo p and q . Then it is only a quadratic residue modulo N if it is a quadratic residue modulo both p and q .

If an enemy tries to decipher the ciphertext without knowing the factors p and q , then the problem becomes very hard. It is thought that this is as hard as finding the factors. Goldwasser and Micali proved that this scheme is semantically secure against an enemy who has polynomially bounded resources, at least provided that there is no polynomial time algorithm for determining whether a number is a quadratic residue modulo N .

Random encryption like the Goldwasser – Micali scheme gives greatly improved security but also greatly expands the length of the ciphertexts.