We now present the Rabin-Blum Oblivious Transfer or Coin Flipping Protocol. In it, Alice reveals a secret to Bob with probability 0.5.

In the Oblivious Transfer version, Alice doesn't know whether Bob got the secret or not (and this outcome must be acceptable to both participants).

In the Coin Tossing version, Bob tells Alice whether he got the secret. He wins the coin toss if he did get it; loses otherwise.

Activity – 10 minutes.

Alice and Bob are miles apart and communicate by text messages. They want to decide whether to go to a movie or a concert. Bob prefers the movie; Alice the concert. They agree to toss a coin to decide.

Design a protocol of text messages they can use to “toss a coin.”

Your protocol must be fair to both parties and have a probability of exactly 0.5 for each outcome.

You may use any material mentioned in this class and anything else. Alice and Bob may communicate only by text messages.

Alice's secret is the factorization of a number $n = pq$ which is the product of two large primes $p \equiv q \equiv 3 \pmod 4$.

1. Alice sends $n$ to Bob.

2. Bob picks a random $x$ in $\sqrt{n} < x < n$ with $\gcd(x, n) = 1$. Bob computes $a = x^2 \bmod n$ and sends $a$ to Alice.

3. Knowing $p$ and $q$, Alice computes the four solutions to $x^2 \equiv a \pmod n$. They are $x$, $n - x$, $y$ and $n - y$, for some $y$. These are just four numbers to Alice. She doesn't know which ones are $x$ and $n - x$. She chooses one of the four numbers at random and sends it to Bob.

4. If Bob receives $x$ or $n - x$, he learns nothing. But, if Bob receives $y$ or $n - y$, he can factor $n$ by computing $\gcd(x + y, n) = p$ or $q$.

Why can Bob factor $n$ if he gets $y$ or $n - y$?

**Theorem**. If $n = pq$ is the product of two distinct primes, and if $x^2 \equiv y^2 \pmod{n}$, but $x \not\equiv \pm y \pmod{n}$, then gcd$(x + y, n) = p$ or $q$.

**Proof**: We are given that $n$ divides $(x+y)(x-y)$ but not $(x+y)$ or $(x-y)$. Hence, one of $p$, $q$ must divide $(x + y)$ and the other must divide $(x - y)$.

It is easy to modify the Oblivious Transfer protocol to let Alice give Bob the content of an arbitrary file with probability 0.5. Alice's secret is the content of the file.

Alice enciphers the file using AES with secret key $K$. She gives the ciphertext of the file to Bob.

Alice chooses two large primes $p \equiv q \equiv 3 \pmod 4$, sets $n = pq$ and chooses $0 < e < n$ with $\gcd(e, (p-1)(q-1)) = 1$. This sets up an RSA public key cipher with public key $n$ and $e$. Alice enciphers $K$ as $C = K^e \bmod n$. Alice gives Bob $C$ and $e$.

Then Alice and Bob do the Oblivious Transfer protocol, Alice sending $n$ to Bob in Step 1.

If Bob learns the factorization of $n = pq$ in Step 4, then Bob finds $d$ with $ed \equiv 1 \pmod{(p-1)(q-1)}$ by extended Euclid. He finds $K = C^d \bmod n$, and deciphers the file using $K$ as the AES key.

Activity – 10 minutes.

Alice knows a secret. Bob does not know it. Bob is skeptical that Alice knows the secret. Alice wants to prove to Bob that she knows the secret.

They engage in a protocol of text messages to each other. At the end of this protocol, either Bob discovers that Alice did not know the secret or Bob becomes convinced that Alice does know the secret. During the protocol, Bob learns neither the secret nor anything that would help him deduce the secret.

Design a protocol that does this. You may choose the kind of secret Alice claims to know.

# Zero-Knowledge Proofs

This protocol is closely related to the oblivious transfer protocol. The difference is that Alice wants to convince Bob that she knows the factors of $n = pq$, but does not want to reveal the factors to Bob.

Alice (the prover) convinces Bob (the verifier) that she knows the prime factorization of a large composite number $n$, but does not give Bob any hint which would help him find the factors of $n$. Bob learns nothing about the factorization of $n$ during the protocol that he could not have deduced on his own without Alice's help.

Roughly speaking, Bob gives Alice some quadratic residues modulo $n$ and Alice replies with their square roots. The difficulty with this simple approach is that when Alice replies to Bob with a square root, there is a 50% chance that she will reveal the factorization of $n$ to Bob, as in the oblivious transfer protocol.

Here is a good way to do the zero-knowledge proof protocol:

Alice knows $n$, $p$ and $q$. Bob knows $n$ but not $p$ or $q$.

1. Alice chooses $a$ in $\sqrt{n} < a < n$ and computes $b = a^2 \bmod n$.

2. At the same time, Bob chooses $c$ in $\sqrt{n} < c < n$ and computes $d = c^2 \bmod n$.

3. Alice sends $b$ to Bob and Bob sends $d$ to Alice.

4. Alice receives $d$ and solves $x^2 \equiv bd \pmod{n}$. (Note that this is possible because $bd$ is a QR and she can compute its square root because she knows the factors of $n$.) Let $x_1$ be one solution of this congruence.

5. At the same time, Bob tosses a fair coin and gets Heads or Tails each with probability 0.5. Bob sends H or T to Alice.

6. If Alice receives H, she sends $a$ to Bob. If Alice receives T, she sends $x_1$ to Bob.

7. If Bob sent H to Alice, then he receives $a$ from Alice and checks that $a^2 \equiv b \pmod{n}$. If Bob sent T to Alice, then he receives $x_1$ from Alice and checks that $x_1^2 \equiv bd \pmod{n}$.

Alice and Bob repeat steps 1 through 7 many (20 or 30) times.

If the check in step 7 is always okay, then Bob accepts that Alice knows the factorization of $n$.

But if Alice ever fails even one test, then Bob concludes that Alice is lying.

Why does this protocol work?

Why does Bob not learn the factors of $n$?

Can either Alice or Bob cheat?