# Title slide:

My topic is a cryptographic primitive known as zero knowledge proofs

Before going into that let’s take a look at … <spacebar>

# IPS:

… interactive proof system.

These systems exist to prove a piece of knowledge or a statement. They consist of a prover who is trying to prove the statement to the verifier. They communicate by exchanging messages of the type commitment – a kind of setup message, challenge and response. Then the verifier can verify the response wrt the commitment. And accept if it all goes well. <spacebar>

# ZKPs:

On to ZKPs. They were first introduced in 1985 as an extension of the interactive proofs. <spacebar> The motivation came from the fact that interactive proofs leak the statement being proved. For example If I tell you prove that a number is composite – you can just show me its factorization. But then I’ll know the factors of that number – which may not be desirable.

<spacebar> One last thing. All these proof systems are probabilistic – as in there’s always a very small probability that the cheater can guess the correct responses to all the challenges. Good proofs try to minimize this probability. <spacebar>

# Enter the cave:

This is a classic example of a ZKP. Here we have a cave. The front wall is glass so that we can see through it. The tunnel is separated into two parts – L & R – by a locked door. Victor is our verifier. Peggy is our prover. I did not come up with the names. And the goal of this exercise is for peggy to prove to victor that she has the key to this door without showing him the key. <spacebar>

# Commitment:

The commitment is as follows: Victor closes his eyes. Peggy chooses a random side – L or R and goes into that side. Point is that victor does not know where peggy went. Lets say peggy went to R. <spacebar>

# Challenge:

Now victor chooses a side at random. And tells peggy to come out that side. Now if victor chose the same side as peggy – R – then peggy doesn’t need her key. She can just walk back. But if victor chooses L …<spacebar>

# Response:

… then peggy must use her key to open the door and come out the other side. Now the chance of this happening is 50%. So theres a 50% chance that peggy cheated and she doesn’t have the key. Those are not good odds. But we can improve them … <spacebar>

# More rounds:

… By repeating this procedure many times. <spacebar><spacebar> Do it twice and we have a 25% chance that peggy cheated. <spacebar><spacebar> if we do it 40 times, then there’s only a one in a million million chance that peggy cheated – that she and victor chose the same side 40 times. So now victor can say with confidence that she has the key – but victor has no idea what the key looks like. That is zero-knowledge. <spacebar>

# Properties:

Every ZKP must have 3 properties: <spacebar> The first two are properties of every interactive proof – they basically say that it’s always possible to proof something that’s true, and always impossible to prove something that’s false. This protects against cheating provers. <spacebar>

The third property is what this presentation is about. The zero knowledge property protect against cheating verifiers. It states that after a successful proof the only piece of knowledge revealed is that the statement is true – and nothing else. <spacebar>

# Simulators:

So how can we tell if a proof is indeed zero-knowledge? The formal way is to see if we can simulate it. In the cave example if we recorded victor and peggy going through the whole proof and then showed it to somebody – they would be convinced that peggy has the key. Now can we make a fake recording? If peggy didn’t have the key and she conspired with victor they could choose a preplanned sequence of Ls and Rs. So that she succeeds every time. And the resulting recording would be indistinguishable from the actual one. So we don’t need to know the secret to efficiently simulate the proof – so the original proof did not leak the secret. Contrast with the earlier example – If we prove by showing the factorization of a number then making these fake transcripts is impossible without actually factoring the number. <spacebar>

# Examples:

There are many different types of ZKPs as you can see. The most influential is the proof of knowledge or identity. <spacebar> There are many protocols that are ZK. Fiat-Shamir was the first such protocol. Schorrs identity scheme has been adapted into a signature scheme - a variant of that is DSA which is used almost everywhere. <spacebar>

# FS-1:

For a concrete example let’s take a look at the Fiat-Shamir identity scheme. Now in an identity scheme the prover tries to prove their identity to the verifier – but the system must make sure the verifier can’t impersonate as the prover later. The fiat-shamir scheme is based on the quadratic residure problem – that is we have a number in Znthen finding out its square root is a hard problem if n is composite and we don’t know it’s factorization. It’s equivalent to the integer factorization problem.

To setup we need a trusted authority to choose n for us. Now for security n needs to be a blum integer – that is a product of two primes which are 3 mod 4. It also select k and t which are security parameters. <spacebar>

# FS-2:

Now the prover must select a private key. They do this by selecting k numbers s1 – sk that are coprime to n, called s-values. And we also need a k-bit vector b1 – bk. This is our secret key.

We compute the public key from the secret key by taking inverse square of each si, called v-values. We publish the v-values, and keep the s-values secret. <spacebar>

# FS-3:

Each round in fiat-shamir goes like this. First the commitment. Prover choses a random number r, and sends r^2 mod n to the verifier. Then the challenge. The verifier sends a k-bit string. This is a selection vector. He’s choosing which of the s-values are part of this round. Then the response is this. r times the selective produst of s-values. Then the verifier can verify by checking this. You can see that y^2 is r^2 times s^2. So that times v^2 which is equal to 1/s^2 will cancel the s^2 and leave r^2 which is x.

# FS-4:

Now we repeat this t times and fiat-shamir guarantees that we will accept a false proof with probability 2^-kt. So kt determines how good the proof is.

But is this a ZKP? Well….<spacebar> for sake of simplicity we can assume k=1. So that there is only one s-value and v-value.

Now it is complete because if prover knows s, then she can respond correctly no matter what the verifier challenges us, e=0 or 1.

Now it is sound, because if prover doesn’t know s, then they can only answer correctly if they either knew the challenge beforehand or could solve the quadratic residue problem.

And it is zero knowledge because we can simulate it efficiently. No need to know s, we just set x accordingly. Transcripts will be indistinguishable. <spacebar>

# Application and attacks:

Finally… we can apply ZKPs in a lot of places. The main one is in creating Signature schemes. The Fiat-shamir heuristic is a mechanism that can turn any ZKP into an equivalent signature scheme. This is how DSA was created. We can use them for anonymous voting, proofs of membership in auctions and may other places. <spacebar>

And of course no cryptographic protocol is complete without attacks targeting said protocol. Of these replay attacks are the most dangerous where you just copy a successful transcript and use it again later, perhaps to impersonate someone. <spacebar>

And that’s it. Here are my references. Thanks. <The End>