University of Illinois, Urbana Champaign
CS/ECE 498AC3/4 Applied Cryptography

Instructor: Dakshita Khurana
Scribe: Jeffery Haag
Date: october 27, 2020

LECTURE
19

Pairing-based ZK: II

19.1 Previous Lecture

Pairing-Based Cryptography

We went over how a pairing is an efficiently computable function $e(G_0 x G_1) \rightarrow G_T$ so that

1. $g_T = e(g_0, g_1)$ is a Generator of G_T
2. For $(u, u') \subseteq G_0 and (v, v') \subseteq G_1$

$E \subseteq (u.u', v) = e(u, v).e(u', v)$ and $e(u, v.v') = e(u, v).e(u, v')$

This is interesting to us because we can efficiently multiply exponents and can aid us in zero knowledge non interactive systems.

Commitments

To commit to a message m, you raise $g^m h^r$ where g is Gen(g) and h is Gen(G_q). This commitment is perfectly binding, which means that for any distinct m chosen there will be no intersect with other commits no matter the r value. Also h is indistinguishable from $h \leftarrow$ Gen(g)(computational hiding). We also wanted to prove whether a string was a commitment to 0 or to 1. We found that if $h = cg^{-1}$, $e(c, h) = 1$ if c is truly a commitment to 0 or 1.

19.2 Proof That A commitment is to 0 or to 1

Given Parameters: (g, h) where $g \leftarrow$ Gen(G) and $h \leftarrow$ Gen($G(q)$).

Commit $(m, r) = c = g^m h^r$ Verify (y1, y2, y3) where $s \leftarrow Zn*, t = s^{-1} \mod n$,

$u = g^{2m-1}h^r$, $y_1 = h^s$, $y_2 = u^{rt}, y_3 = g^s$ To verify $(c, y1, y2, y3) \rightarrow check$
We will do the second check then the first check because it makes more sense in this order.

The second check is whether $e(y_1, g) = e(h, y_3)$
This is to simply check that paring y_1 with g is the same as paring y_3 with h. This is because for any y_3 it is must be some g raised to the power of some s and y_1 must be some h to the same power of s for this check to pass. This establishes that y_1 is h^s for some s. If not this test will fail.

The first check is: $e(c, cg^{-1}) = e(y_1, y_2)$
From the second check we know that $y_1 = h^s$ for some s. If the prover is behaving honestly $e(y_1, y_2) \equiv e(h^r, u^{st})$. st is just one so it becomese(h^r, u). Again if the prover is behaving honestly $u = g^{2m-1}h^r$.
Given what we have above we know that if the message is 0, $m = 0$ and $c = h^r$. We substitute so it is now $e(h^r, g^{2(0)-1}h^r) = e(h^r, g^{-1}h^r) = e(c, c^{g-1})$ which is of course equal to $e(c, c^{g-1})$.
If however it was a commitment to 1 we know that $c = gh^r$ and the proof becomes

$$\begin{aligned} e(h_r, g^{2(1)-1}h^r) &= e(h^r, g^1h^r) \\ &= e(c/g, c) \\ &= e(c, cg^{-1}) \end{aligned}$$

which is clearly equal to $e(c, cg^{-1})$ so it passes the test. This lecture we worked on the soundness and touched on the zero knowledge aspect of pairing before briefly saying how it could be of use.

19.3 Proof that a commitment is to 0 or 1: Soundness

We now want to be able to have the message m not be in 0,1 but the prover is still being honest. From above we have $e(y_1, y_2) = e(h^r, g^{2m-1}h^r)$ (if prover is still being honest) Recall that h looks like g raised to some $\alpha * p$. (note p is some prime and α is just some number) So we substitute $e(h^r, g^{2m-1}h^r) = e(g^{\alpha*p*r}, g^{2m-1}g^{\alpha*p*r})$

Since we can pull exponents out this becomes $e(g, g)^{(\alpha*p*r)*(2m-1+\alpha*p*r)}$.

For the first proof we want to see if

$$\begin{aligned} e(g, g)^{(\alpha*p*r)*(2m-1+\alpha*p*r)} &= e(c, c^{g-1}).e(c, c^{g-1}) \\ &= e(g^m h^r, g^{m-1}h^r) \\ &= (g, g)^{(m+\alpha*p*r)*(m-1+\alpha*p*r)}. \end{aligned}$$

So is $e(g, g)^{(\alpha*p*r)*(2m-1+\alpha*p*r)} = e(g, g)^{(m+\alpha*p*r)*(m-1+\alpha*p*r)}$?

By observation we know $\alpha * p * r$ is just some multiple of p. We know that $m < p$ to be a legal message. For the two sides to be equal one of the terms on the right $(m + \alpha * p * r)$ or $(m - 1 + \alpha * p * r)$ must be equal to a multiple of p because on the left side $\alpha * p * r$ is of course a multiple of p. Since $m < p$ this is impossible. We reject it because it fails the first test. This is good for us so we can verify that they only committed to 0 or to 1 if

they behave honestly. This isn't enough to prove soundness though because the prover may generate y_1, y_2, y_3 any way they want. We do know that they must set $y_1 = h^s$ and $y_3 = g^s$ or the second test will fail. So really the only one they can modify and still pass the second test is y_2. We do know that $y_2 = g^\beta$ where β is just some number. To pass the second test it must pass $e(c, cg^{-1}) = e(y_1, y_2)$ and $e(y_1, y_2) = e(h^s, g^\beta)$ recall from above that h looks like g raised to some $\alpha * p$. so it becomes $e(g, g)^{(\alpha * p * s * \beta)}$. No matter what the prover does the paring of $e(y_1, y_2)$ will be of the form g^β for some β. This is some multiple of p and we can exploit the same logic as before because $m < p$. So still, unless $m = 0$ or $m = 1$, we will reject. What if however c is maliciously generated? Well this means nothing because we know that for the prover setting $c = g^\gamma$ for any γ will be equivalent to $c = g^m h^r$ for some m and some r so the rest of our tests will still hold.

19.4 Proof that a commitment is to 0 or 0: ZK(Zero Knowledge)

Remember from the beginning that $h \leftarrow \text{Gen}(g)$ and $h \leftarrow \text{Gen}(g_q)$ can't be differentiated from each other. From the verifier point of view these h's are indistinguishable.

So h can be thought of as $h = g^\alpha$ where $\alpha \subseteq 1, ..N - 1 | p, \ldots 2p, \ldots q, \ldots 2q$. We design a simulator. Given some commitment c it must generate a proof that it is a commitment to 0 or to 1. It will do this by cheating and sample $h = g^\alpha$ and the simulator will know what α is. With this secret information it can create an indistinguishable proof from the normal one by creating specific y_1, y_2, y_3. This can be used as a building block to prove arbitrary general statements about the behavior of circuits without revealing any secrets about them.

19.5 Putting it all Together

All statements can be converted to a circuit of NAND gates. Suppose we have circuit C from NAND gates. Prove there exist assignments to wires w $=(w_1, w_2 \ldots, w_o ut)$., such that $C(w) = 1$. The prover knows of some way to make it output 1 and the circuit is large and complex enough to where the verifier can't go through it all. Each bit is thought of as a commitment on each wire. To convince the verifier that it outputs 1 it is enough to show that the final commitment is 1 and all the previous commitments are to 0 or to 1(we can prove this easily by using the same technique as shown above). Given the commitment of 2 inputs and 1 output of a random NAND gate we want to prove that the output commitment is the result of the NAND operation.

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

19.6 Summary

To summarize the prover sends the verifier a circuit and wants to prove there is some assignment of wires in the circuit that would lead the circuit to output 1. To do this the

prover commits to all assignments and proves that all these commitments are to 0 or to 1 and also proves the output of each gate is in fact the output of the NAND operation. By this we can prove statements to the verifier a statement without revealing any information about our circuit.