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

Instructor: Dakshita Khurana
Scribe: Lawson Probasco
Date: October 22, 2020

LECTURE
18

Pairing-Based Zero Knowledge

The intention of this lecture is to expand the idea of zero knowledge proofs, in particular non-interactive zero knowledge proofs, using a new pairing based protocol. As a refresher, remember that in zero knowledge protocols, the goal is for a prover to convince a verifier that they know some secret, without leaking information about the secret in the process. This generally involves the process of commitment, in which the prover commits to some value for the secret, and verification, for which the verifier proves some important fact about the secret.
In order to build towards our pairing based zero knowledge protocol, the lecture content will first review the idea of a pairing. We will further explore properties related to this pairing, as well as a set of useful assumptions that allow us to build our new pairing based zero knowledge protocol. When all these key pieces are motivated and expanded on, we will begin building the protocol - first by exploring the nature of its commitments, and then understanding the method for which a commitment can be proven to be on either 0 or 1.

18.1 Pairing-based cryptography

In order to motivate the idea of pairing-based cryptography, we have looked at the hardness of certain problems such as computation Diffie-Hellamn (CDH) and decisional Diffie-Hellman (DDH) in particular groups. Because of the properties of pairing however, the nature of the hardness of these problems changes, and allows for protocols to take advantage of this in interesting ways - for example, bilinear mappings in cyclic groups lead to DDH being easily solvable with the use of the pairing function. This can be used to build efficient implementations of pairing-based zero knowledge explored later in this lecture.

Definition 18.1. Let G_0, G_1, G_T be cyclic groups of prime order, where $g_0 \in G_0$ and $g_1 \in G_1$ are generators. A **pairing** $e : G_0 \times G_1 \to G_T$ is an efficiently computable function such that:

1. $g_T = e(g_0, g_1)$ is a generator of G_T
2. $\forall (u, u') \in G_0$ and $(v, v') \in G_1$:
    - $e(u \cdot u', v) = e(u, v) \cdot e(u', v)$

- $e(u, v \cdot v') = e(u, v) \cdot e(u, v')$

THEOREM 18.2. *As a result of property 2 above, a pairing also has the following property.*

- $e(g_0^a, g_1^b) = e(g_0, g_1)^{ab} = e(g_0^b, g_1^a)$

Proof. Using the above property on a slightly simplified example, $e(g_0^a, g_1)$ can be expanded a times to be $e(g_0, g_1) \cdot e(g_0, g_1) \cdot ... \cdot e(g_0, g_1) = e(g_0, g_1)^a$. □

COROLLARY 18.3. *Exponents in pairing groups multiply, namely:*

- $e(g^a, g^b) = g_T^{ab}$

It should be noted that this means DDH is easily solved in pairing groups.

18.2 Commitments with pairings

When building a commitment scheme using pairing based cryptography, it is important to introduce the following **subgroup hiding assumption** that will come in handy later when reasoning about the binding and hiding properties of this scheme.

DEFINITION 18.4. The *Subgroup Hiding Assumption* is one that assumes the advantage of an adversary $\mathcal{A}$ in the following game to be negligible.

- Let G, G_T be cyclic groups of compost order $N = pq$, and let (g, g_T) denote generators of (G, G_T). Also define the pairing $e : G \times G \to G_T$,
- Define $G_{(q)}$ to be a unique order-q subgroup of G
- Pick a generator g of G
- Set $h_0 \leftarrow$ random generator of G, $h_1 \leftarrow$ random generator of $G_{(q)}$
- Send (g, h_b) to $\mathcal{A}$: $\mathcal{A}$ wins if it correctly guesses b

To put it plainly, this assumption states that generators in G and $G_{(p)}$ are indistinguishable.

With the subgroup hiding assumption in mind, we can build commitments in the following way.

- Public parameters: $(g \leftarrow Gen(G), h \leftarrow Gen(G_{(q)}))$
- $\text{Commit}(m; r) = g^m h^r$

This commitment is perfectly binding, meaning $\forall m_1 \neq m_2 \in Z_p, Supp(\text{Commit}(m_1; r)) \cap Supp(\text{Commit}(m_2; r)) = \{\}$. This property does take advantage of the fact that the subgroup $G_{(q)}$ that we are concerned with is of the form $\{g^p, g^{2p}, ...\}$.
We can also say this commitment is computationally hiding by invoking the subgroup hiding assumption. Namely, since when $h \leftarrow Gen(G)$ the commitment is perfectly hiding, $h \leftarrow Gen(G_{(q)})$ being computationally indistinguishable from $h \leftarrow Gen(G)$ means this commitment is computationally hiding.
This commitment belongs to a class of commitments obeying the following interesting property, and is referred to as a Dual-Mode Commitment.

REMARK 18.5. When $(g, h \in Gen(G_{(q)}))$, this commitment is perfectly binding and computationally hiding (as above). However, if $(g, h \in Gen(G), N)$, then the commitment is computationally binding and perfectly hiding.

This commitment is also homomorphic, e.g. $\text{Commit}(m_1; r_1) \cdot \text{Commit}(m_2, r_2) = g^{m_1+m_2} h^{r_1+r_2} = \text{Commit}(m_1 + m_2; r_1 + r_2)$

18.3 Proof that a commitment is to 0 or 1

As with other zero knowledge protocols, it is important that some fact can be proven about a commitment even when the specifics of the commitment are hidden from view. In the simplest case, for this commitment we will be concerned with proving that when given some c it is a commitment to 0 or 1.
In the case that we are concerned with, $c = g^m h^r$ can assume two values (note the bracket notation with subscript r refers to the set of all elements over variations of r).

- $c = \{g^0 h^r\}_r \cup \{g^1 h^r\}_r$

In the case of a commitment to 0, $c = g^0 h^r = h^r$, and thus $c \in G_{(q)}$ and $c^q = 1_G$. In the case of a commitment to 1, $c = g^1 h^r = h^r$, and thus $cg^{-1} \in G_{(q)}$ and $(cg^{-1})^q = 1_G$. Therefore, one way in which we can go about proving a commitment is to 0 or 1 is by checking if at least one of these quantities c^q or $(cg^{-1})^q = 1_G$. In an attempt to do this, we can use the pairing function coupled with the following theorem.

THEOREM 18.6. $\forall A, A', e(1_G, A) = 1_T = e(A', 1_G)$.

Proof.

$$\begin{aligned} e(1_G, A) &= e(g^N, A) && 1_G = g^N \\ &= (e(g, A))^N && \text{Defn. 18.1 (2)} \\ &= 1_T && \text{element in } T \text{ raised to power of } N \end{aligned}$$

□

As a result of this theorem, a good first step is the idea that a check for if $e(c, cg^{-1})^q = 1_T$ would determine if c is a commitment on 0 or 1, because if so then either $c^q = 1$ or $(cg^{-1})^q = 1$, and thus $e(1, .) = e(., 1) = 1_T$ by theorem 18.6. However, q is not part of the set of public parameters, and thus this approach isn't possible. Instead, we can modify our approach to proof in the following way.

Proof $= (y_1, y_2, y_3)$ where

- $s \leftarrow Z_n^*, t = s^{-1} mod n, u = g^{2m-1} h^r$
- $y_1 = h^s$
- $y_2 = u^{rt}$
- $y_3 = g^s$

This new method of establishing a proof yields the following verification scheme over (c, y_1, y_2, y_3):

Verify(c, y_1, y_2, y_3)

- check $e(y_1, g) = e(h, y_3)$ to confirm that y_1 is of the form h^s for some s
  - suppose $y_1 = g^s \notin G(q)$, then $e(h, g^{\hat{s}}) = e(h, g)^{\hat{s}} = e(y_1, g)$ iff $y_1 = h^{\hat{s}}$
- next, check $e(c, cg^{-1}) = e(y_1, y_2)$ to determine if $c \in G_{(q)}$ or $cg^{-1} \in G_{(q)}$
  - the check $e(y_1, y_2)$ can be re-written as $e(y_1, y_2) = e(h^s, u^{rt}) = e(h^r, u^{st}) = e(h^r, u) = e(h^r, g^{2m-1} h^r)$
  - if $m = 0$, this check $e(y_1, y_2) = e(h^r, g^{-1} h^r) = e(c, cg^{-1})$
  - if $m = 1$, this check $e(y_1, y_2) = e(h^r, gh^r) = e(cg^{-1}, c) = e(c, cg^{-1})$

This proof is now in terms of known quantities, and as a result circumvents the earlier dependency on the unknown value q.

Acknowledgement

These scribe notes were prepared by editing a light modification of the template designed by Alexander Sherstov.