Merkle Tree Nedir?

Hash ağaçları olarak da adlandırılan Merkle Tree, verilerin güvenli ve verimli bir şekilde doğrulanmasını sağlayan blockchain teknolojisinin zorunlu bir bileşenidir. ++

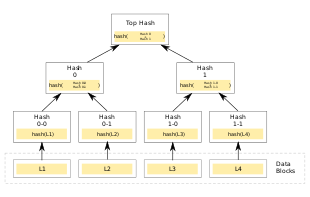
**Merke Tree Nasıl Oluşturulur?**

* Her veri için bir hash değeri bulalım, bu değerler Merkle Tree’nin yapraklarını ifade ediyor.
* Yaprak düğümleri soldan sağa doğru ikili şekilde gruplandıralım.
* Çift olan iki yaprak için yeni bir hash oluşturup bir düğüm oluşturalım. ++
* Oluşturduğumuz düğümleri tekrardan ikili şekilde gruplayalım.
* Grupladığımız düğümlerin hashlarini birleştirerek yeni bir düğüm oluşturalım.
* Bu olayı kök dizine ulaşana kadar yapalım. ++

Merkle kökü, Merkle ağacının bir temsilcisidir. Ağaç tarafından temsil edilen veri kümesinin dijital parmak izi görevini görür. ++

Verilerdeki küçük bir değişikliğin bile yaprağın karma değerini değiştireceğini ve bunun bu yaprak düğümünü kök düğüme bağlayan tüm düğümlerin karma değerlerini değiştireceğini ve sonunda kök karmasında değişikliğe neden olur. ++

Örnek olarak:



Yukarıda bulunan Merkle Tree’yi incelemek istersek. L1,L2,L3,L4 bizim verilerimiz bu verileri ikili gruplandırdıktan sonra her veri için bir hash alıp gruplandırma sonucu yeni bir hash üretiyoruz. ++

Hash(L1)+Hash(L2) için Hash(L1,L2), Hash(L3)+Hash(L4) için Hash(L3,L4) oluşturuyoruz. Ardından bu Hash(L1,L2), Hash(L3,L4) için son kez bir hash oluşturarak kökü elde ediyoruz. Burada anlaşılması gereken temel sonuç: ++

Bir merkle tree içinde herhangi bir veriyi değiştirirsek o değişen verinin nerede olduğunu kolaylıkla bulabiliyoruz. Çünkü değişen yapıdan itibaren üst katmanların Hash verisi değişiyor. ++

**Neden Merkle Tree?**

Merkle ağaçları, depolama ve zaman açısından verimli veri yapılarıdır. Daha az depolama alanı gerektiren verilerin karma değerlerini depolarlar. Tüm ağacı aramak yerine Merkle yolunu üreterek kolayca veri aranabilir. ++

Merkle ağacının verimliliği ayrıca hash fonksiyonlarının seçimine de bağlıdır.Merkle ağaçlarının, verilerin bir ağdaki birden çok taraf tarafından çoğaltıldığı ve depolandığı çeşitli dağıtılmış uygulamalarda kullanıldığını unutmamalıyız. ++

Herhangi biri bir konumdaki verileri değiştirmeye çalışırsa, karma değerleri değişir ve kök karmasında bir değişikliğe yol açar. Böylece ağdaki diğer kişiler, kök karmadaki uyumsuzluğa dayanarak bunu algılayabilir.

