HASH NEDİR HASH ALMANIN ÖNEMİ

Bu yazımızda hash fonksiyonlarının tanımını, adımlarını, tarihçesini, önemini ve kullanım alanlarını ele alacağız. Türkçe'de özetleme fonksiyonu olarak adlandırılan bu kavramı, yazı boyunca İngilizce ifadesiyle kullanacağız. İlk olarak hash fonksiyonunun tanımını yaparak başlayalım. Hash fonksiyonu, çeşitli girdileri benzersiz sabit boyutlu çıktılara dönüştüren matematiksel bir fonksiyondur.

Hash fonksiyonunun temel özelliklerini ve matematiksel ifadesini şu şekilde maddeleyebiliriz (hash fonksiyonu yerine h(x) kullanalım):

- Bir hash fonksiyonuna verilen girdinin uzunluğu çıktının uzunluğunu etkilemez. Hash fonksiyonunun çıktısı her zaman sabittir.

  $$h(x_1) = a \; ve \; h(x_2) = b => length(a) = length(b)$$

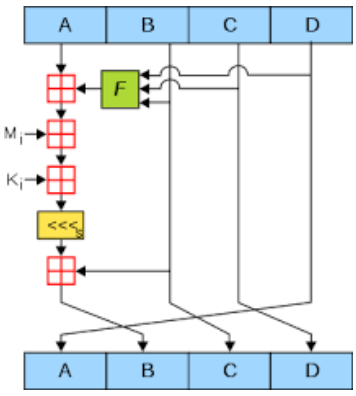
- Güvenli bir hash fonksiyonu farklı iki girdiye aynı çıktıyı vermemelidir. (İki farklı girdinin aynı çıktıyı verme olasılığı çok düşük olmalıdır.)

  $$h(x_1) = a \; ve \; h(x_2) = b => a \neq b$$

- Bir hash fonksiyonunun tersi elde edilemez olmalıdır. Bu özellik ters görüntüye dayanıklılık olarak isimlendirilir.

  $$h(x_1) = a \quad h^{-1}(x) \; bulunamaz \; olmalıdır.$$

1979'de icat edilen Merkle-Damgård yapısına dayalı hash fonksiyonu, bilinen ilk hash fonksiyonlarından biridir. Bu tarihin öncesinde bazı algoritmalar icat edilmiştir. Ama günümüzdeki özelliklere yakın ilk hash fonksiyonu Merkle-Damgård sayılabilir. Bu fonksiyon, hash fonksiyonlarının evriminde önemli bir kilometre taşı olarak kabul edilir. Merkle-Damgård yapısının kullanılması, veriyi bloklara bölen ve ardından bu blokların işlenmesini sağlayan bir yöntemi içerir. Bu yönteme dayanan başka bir hash fonksiyonu olan MD5'in adımlarını inceleyelim:



1- Fonksiyonumuz aldığı girdiyi daha öncesinde uzunluğu tanımlanmış bloklara böler. Örneğin 512 bitlik bloklara böler.
2- İlk blok diğer bloklarla mantıksal işlemlere tabi tutulur. Bu mantıksal işlemler daha önceden belirlenmiş fonksiyonlardır ve her aşamada farklılık gösterir.
3- Mantıksal işlemler uygulandıktan sonra farklı matematiksel ve kaydırma işlemlerinden geçirilir. Blok normalde bulunduğu konumdan 1 blok kaydırılarak çıktıdaki yerini alır.

Çok temel bir örnekle başka bir hash fonksiyonu olan SHA1 üzerinden konuyu detaylandıralım. SHA1 fonksiyonuna girdi olarak 'merhaba dünya' ifadesini verelim sonrasında küçük bir değişiklik yaparak çıktıların arasındaki farkı inceleyelim:

$$x1 = '\; merhaba \; dünya' \quad SHA1(x_1) = \; 62f481c664a51f826f258785427df4a4d85b84c8$$

$$x2 = 'Merhaba \; dünya' \quad SHA1(x_2) = 472ba9d6db95e5266d4b2a5dab63914c59760594$$

İdeal hash fonksiyonları gereken tüm şartları sağlar. Ama dünyamızda ideal ancak bir teoriden ibarettir. Başlangıçta yukarıdaki özellikleri sağlayan hash fonksiyonlarında, - zamanla donanımların gelişmesiyle - bazı zafiyetler tespit edilmiştir. Bu yüzden SHA0, SHA1 gibi hash fonksiyonlarının yerini daha uzun çıktılar üreten SHA256 ve SHA512 gibi fonksiyonlar almıştır.

Hash fonksiyonları devamlı testlere tabi tutulmaktadır. Eğer bir zafiyet tespit edilirse bu konuda geliştirmeler yapılmakta ve zafiyet tespit edilemeyen fonksiyonların kullanımına geçilmektedir. Örneğin SHA0 fonksiyonunda farklı girdilerin aynı çıktıyı verdiği bazı örnekler bulunmuştur. Bu sebeple SHA0 aktif olarak kullanılmamaktadır. Bu konuya başka bir örnek de tuzlama tekniğidir. Bu teknik hash fonksiyonunun girdisini hash işlemi yapılmadan önce belirlenmiş rastgele bir ifade ile birleştirdikten sonra hash işleminin yapılması anlamına gelir. Bu şekilde çıktının çözümlenmesi daha da zorlaştırılır.

ÖRNEK:

$$x1 = 'parola' + '\$7H3\$YVYkoRt' \quad SHA1(x_1) = 55b77b307cbe726496d86f33ce8a83cfebbbfc0e$$

Bu bilgileri edindiğimizde aklımıza bazı sorular takılabilir. Önce soruları soralım sonra örnekleriyle cevaplandıralım: “Neden hash fonksiyonlarına ihtiyacımız var, kullanım alanları neler?“

1- PAROLALARI SAKLAMAK:

Kullanıcının herhangi bir uygulamaya kayıt yaptırması sonrası parolasının daha sonra doğrulanma amacıyla veri tabanında tutulması gerekmektedir. Aynı zamanda veri tabanında tutulan bu parolanın uygulama yöneticisi ve geliştiriciler tarafından bilinmemesi gerekir. Bu sebeple kullanıcı bir sisteme kaydolduğunda parolası hash fonksiyonundan geçirildikten sonra veri tabanına kaydedilir. Kullanıcı daha sonra giriş yapmak istediğinde parolasını girer. Parolası hash fonksiyonundan geçirilir eşleşme gerçekleşirse kullanıcı başarılı şekilde giriş yapmış olur. Bu sayede kullanıcının parolası üzerinde tam koruma sağlanır.

2- VERİ BÜTÜNLÜĞÜNÜ SAĞLAMA:

Adli bilişimde bir zararlı yazılımın bulaştığı bilgisayarın analiz edilmesi gerekmektedir. Adli bilişim uzmanı analiz becerilerine sahiptir. Uzman, cihazın imajını almadan önce imajın hash fonksiyonu çıktısını ayrı bir yerde saklar. Çünkü bu imajlar sadece analiz etmek için kullanılacaktır ve analiz esnasında bütünlüğün bozulmamış olması gerekmektedir. Analiz bittiğinde zararlı yazılımın anatomisi çıkarılır. İmajın son hali üzerinde bir daha hash fonksiyonu çıktısı alınarak başlangıçta elde edilen çıktıyla eşleştirme yapılır. Analizin doğru olduğunu kanıtlamak için bu yöntem elzemdir.

3- KİMLİK DOĞRULAMA:

Ahmet Bey’in bir belgeyi imzalaması gerekmektedir. Ama tüm belgeleri çevrimiçi ortamdadır. Bunun için geliştirilmiş elektronik imza teknolojisinden faydalanmak istemektedir. Elektronik imza atmaya karar verir. İmzalama sırasında imzalanacak dosyanın bir hash fonksiyonu çıktısı alınır. Sonrasında kişinin imzasının açık anahtarı ve hash değeri şifrelenir. Bu imzanın normal imzadan hiçbir farkı olmadığı ve inkâr edilemez olduğunu unutmayalım. Kısaca Ahmet Bey bu kolaylığı hash fonksiyonlarına borçludur.

Bu maddeler dışında başka maddeler de hash fonksiyonlarının kullanım amaçlarına eklenebilir. Ama hash fonksiyonlarının burada incelediğimiz 3 temel başlığı diğer amaçların altyapısını oluşturmaktadır.

Sonuç olarak, hash fonksiyonları, bilişim dünyasında parolaları saklamak, veri bütünlüğü ve kimlik doğrulama gibi temel unsurları destekleyen kritik araçlardır. Bu alanlarda sürekli evrim geçiren hash fonksiyonları, gelecekte de yeni teknolojik gelişmelere uyum sağlayarak önemini sürdürecektir.

Kaynakça:

www.beyaz.net/tr/guvenlik/makaleler/kriptografik_hash_fonksiyonu_nedir.html

www.beyaz.net/tr/guvenlik/makaleler/hash_fonksiyonu_ozetleme_fonksiyonu_ve_birthday_attack.html

en.wikipedia.org/wiki/Merkle%E2%80%93Damg%C3%A5rd_construction

en.wikipedia.org/wiki/Collision_resistance

spectrum.ieee.org/hans-peter-luhn-and-the-birth-of-the-hashing-algorithm

www.geeksforgeeks.org/merkle-damgard-scheme-in-cryptography/

pentestmag.com/breaking-down-md5-algorithm/

cheapssl.com.tr/blog/sha-1-sha-2-ve-sha-256-hash-algoritmalari-arasindaki-farklar.html

www.btk.gov.tr/e-imza-ile-ilgili-sikca-sorulan-sorular

www.innova.com.tr/blog/dijital-imza-nedir-nasil-kullanilir

https://www.youtube.com/watch?v=2AmKrvTdH-g