Bir hash fonksiyonu ile şifreleme arasındaki fark nedir?



www.yzokul.com.tr

Bir hash fonksiyonu ile şifreleme arasındaki fark nedir?



Şu soruyu görüyorsun: bir hash fonksiyonu ile şifreleme arasındaki fark nedir? Kısa yanıt: bir hash fonksiyonu tek yönlüdür ve geri döndürülmek üzere tasarlanmamıştır; şifreleme ise doğru anahtarla geri döndürülmek üzere tasarlanmıştır. Hashleme, veriden sabit uzunlukta bir değer üretir. Şifreleme, veriyi daha sonra geri yüklenebilecek şekilde karıştırır. Bunlar farklı amaçlara sahip iki farklı araçtır.



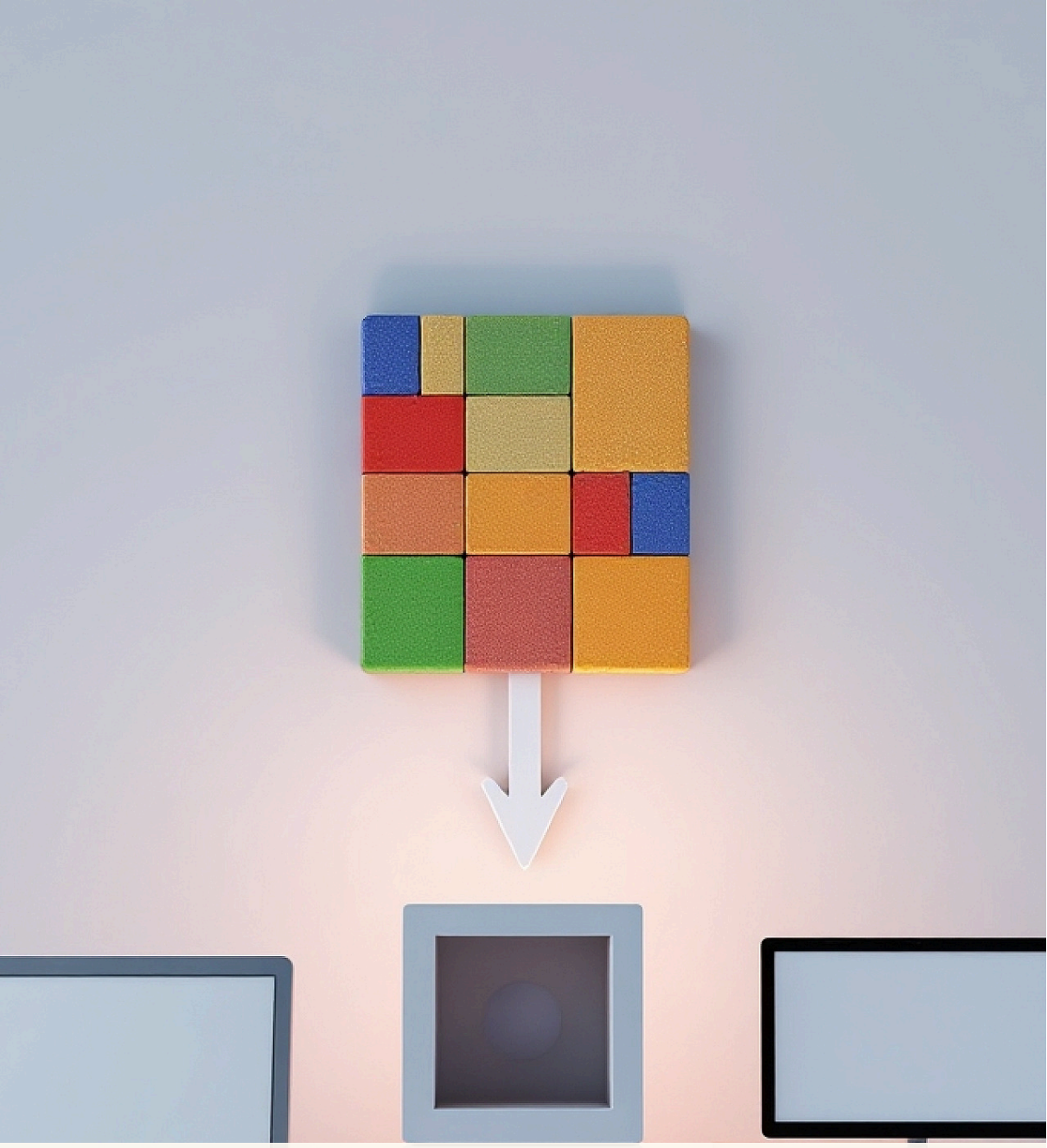
Veriyi herhangi bir içerik olarak düşünüyorsun; örneğin bir metin dosyası, bir fotoğraf veya indirilmiş bir program. Bu, hem hashlemenin hem de şifrelemenin girdi olarak alabileceği başlangıç noktasıdır. Veri, kısa bir not gibi küçük veya bir video gibi büyük olabilir. Verinin boyutu ve türü temel fikir için önemli değildir. Her iki araç da bir çıktı üretmek için veriyi işler.



Bir hash fonksiyonunun her boyuttaki veriyi alıp sabit uzunlukta bir hash değeri ürettiğini görüyorsun. Aynı girdi her zaman aynı hash değerini verir. Örneğin, hello metnini hashlemek harf ve rakamlardan oluşan kısa bir dizi verebilir. Bir hash fonksiyonu, veriyi bir özet koduna dönüştüren matematiksel bir süreçtir. Hızlı ve tekrarlanabilir olacak şekilde tasarlanmıştır.



Minik bir dosya ile devasa bir dosyayı karşılaştırıyorsun ve ikisinin de aynı uzunlukta hash değerleri ürettiğini görüyorsun. Hashin uzunluğu, girdinin uzunluğuna bağlı değildir. Örneğin, tek kelimelik bir dosya ile bin sayfalık bir kitap, diyelim ki 64 karakterlik bir hash verir. Bu sabit uzunluk, hashleri saklamayı ve karşılaştırmayı kolaylaştırır.



Veriden hash değerine giden bir ok görüyorsun, ama geri giden çalışan bir ok yok. Bir hash fonksiyonunun geri döndürülebilir olması amaçlanmamıştır, bu yüzden orijinal veriyi hashten geri alamazsın. Hash değerine sahip olsan bile, dosyayı elde etmek için süreci geriye doğru çalıştıramazsın. Bu tek yönlü özellik, hashlemenin temel bir özelliğidir. Orijinal verinin yeniden oluşturulmasını engeller.



Bir hash değerinin dosyanın sıkıştırılmış, açılabilir bir kopyası olmadığını görüyorsun. Birçok farklı dosya aynı hash uzunluğunu paylaşabilir ve hash, dosyanın içeriğini barındırmaz. Örneğin, bir fotoğrafın hashi fotoğrafın daha küçük bir sürümü değildir. Orijinal veriyi geri almak için bir hashi açamaz veya paketinden çıkaramazsın. Bir hash yalnızca bir parmak izidir, bir kap değildir.



Şifrelemenin veriyi ve bir anahtarı alıp şifreli metin adı verilen karıştırılmış bir çıktı ürettiğini görüyorsun. Amaç veriyi özetlemek değil, gizlemektir. Örneğin, bir mesajı bir parolayla şifrelemek onu okunamaz bir metne dönüştürür. Yalnızca doğru anahtara sahip biri onun şifresini çözebilir. Şifreleme gizlilik ve daha sonra geri yükleme ile ilgilidir.



Şifre çözmenin, doğru anahtarı kullanarak şifreli metni orijinal veriye geri dönüştürdüğünü görüyorsun. Şifreleme, orijinal veriyi geri yüklemeyi amaçlar; hashlemeden temel farkı budur. Örneğin, bir dosyayı bir parolayla şifrelersin, sonra aynı parolayla şifresini çözerek dosyayı geri alırsın. Anahtar olmadan şifre çözme çok zor olmalıdır. Bu geri döndürülebilirlik kasıtlıdır.



Aynı dosyayı hash fonksiyonundan iki kez geçiriyorsun ve her seferinde birebir aynı hash değerini alıyorsun. Bu tekrarlanabilirlik, hashleri karşılaştırma için kullanışlı kılan şeydir. Örneğin, bir belgeyi bugün ve yarın tekrar hashlersen, belge değişmediyse hash aynı olacaktır. En küçük bir değişiklik farklı bir hash üretir. Bu tutarlılık güvenilirdir.



İndirilen bir dosya örneğini görüyorsun: dosyanın değişip değişmediğini kontrol etmek için hashini beklenen hashle karşılaştır. Hashler eşleşirse dosya değişmemiştir; eşleşmezse bir şey farklıdır. Örneğin, bir web sitesi indirdiğin bir programın hashini listeleyebilir. Hashi kendin hesaplar ve karşılaştırırsın. Bu yaygın bir dosya kontrolüdür.



Bir dosyanın küçük bir bölümünü değiştiriyorsun ve hash değerinin tamamen farklı çıktığını görüyorsun. Bu, bir hashin verinin değiştiğini nasıl ortaya koyduğunu gösterir. Örneğin, bir metin dosyasındaki tek bir harfi değiştirmek tamamen yeni bir hash üretir. Hash kademeli olarak değişmez; çarpıcı biçimde değişir. Bu, küçük değişikliklerin bile kolayca tespit edilmesini sağlar.



Hashleme ile şifrelemeyi yan yana koyuyorsun: hashleme veriyi kontrol eder ve özetler, şifreleme ise veriyi gizler ve daha sonra geri yükler. Farklı sorunları çözerler. Örneğin, bir indirmeyi doğrulamak için hash, bir mesajı korumak için şifreleme kullanırsın. Hashleme bütünlük içindir, şifreleme gizlilik içindir. Birbirlerinin yerine geçmezler.



Bir hashin açabileceğin sıkıştırılmış bir dosya olduğu yanılgısını düzeltiyorsun. Bir hash değeri orijinal veriye geri dönüştürülemez. Örneğin, bir belgenin hashini alıp belgeyi okumak için açamazsın. Sıkıştırma geri döndürülebilir; hashleme değildir. Bir hash yalnızca veriyi temsil eden kısa bir koddur.



Bir hashin yalnızca şifrelenmiş veri olduğu yanılgısını düzeltiyorsun. Şifreleme bir anahtarla geri döndürülebilir, ama hashlemenin hiç geri döndürülmemesi amaçlanmıştır. Örneğin, bir hashin kilitli bir kutu gibi olduğunu düşünebilirsin, ama onu açacak anahtar yoktur. Hashleme ve şifreleme farklı işlemlerdir. Bir hash bir şifreleme biçimi değildir.



Bir anahtarın bir hashi veriye geri dönüştürmesini bekleme yanılgısını düzeltiyorsun. Bir hash fonksiyonu için böyle bir anahtar yoktur; anahtar şifreleme ve şifre çözmeye aittir. Örneğin, orijinal dosyayı elde etmek için bir hashi geri döndüren bir parola bulamazsın. Hashlemenin şifre çözme adımı yoktur. Anahtarlar şifreleme içindir, hashleme için değil.



Temel yanıtı hatırlıyorsun: hashleme sabit uzunlukta bir değer verir ve tek yönlüdür, şifreleme ise doğru anahtarla geri döndürülebilir. İndirme kontrolü örneğini ve bir hashin sıkıştırılmış bir kopya olmadığını hatırla. Veriyi kontrol etmek için hashleme, veriyi gizlemek için şifreleme kullan. İki amacı zihninde ayrı tut. Fark budur.