توضیحات بخش هشینگ:

دلیل انتخاب

دلیل انتخاب SHA-1 به عنوان الگوریتم هش به سه بخش اصلی تقسیم میشود: قدرت، سرعت و سادگی
این الگوریتم توانایی تبدیل تمامی اسامی به یک متن هش شده را دارد و برای یک درخت خانوادگی مناسب است.
این الگوریتم دارای سرعت بالاتری نسبت به الگوریتم های SHA-3 , SHA-2 است، البته که از آن دو الگوریتم کمی ضعیف‌تر است، اما در این کیس الگوریتم SHA-1 امنیت مناسبی را برای ما مهیا میکند پس سراغ الگوریتمی می‌رویم که سرعت بالاتری دارد و ساده‌تر است.

پیچیدگی زمانی (Big O)

پیچیدگی زمانی (Big O) این کد به طول پیام ورودی بستگی دارد، کد شامل چند بخش زیر است:

۱. ضمیمه و اضافه کردن طول:
کد یک بیت '1' به پیام اضافه می کند و سپس padding را اضافه می کند تا طول پیام با 448 بیت مدول 512 مطابقت داشته باشد.
این فرآیند زمان O(1) دارد، زیرا اندازه بالشتک ثابت و مستقل از اندازه ورودی است.

۲. حلقه پردازش بلوک:
حلقه اصلی که پیام را در بلوک های 512 بیتی پردازش می کند، روی کاراکترهای پیام تکرار می شود.
حلقه هر بلوک را در زمان ثابت (O(1)) پردازش می کند زیرا روی بلوک های با اندازه ثابت عمل می کند و تعداد عملیات ثابتی را برای هر بلوک انجام می دهد.

۳. محاسبه نهایی هش:
پس از پردازش تمام بلوک ها، کد هش 160 بیتی نهایی را تولید می کند.
محاسبه هش نهایی شامل تعداد ثابتی از عملیات (زمان ثابت) است.

درنهایت، عامل غالب در پیچیدگی زمانی، حلقه پردازش بلوک است. اگر طول پیام ورودی n باشد و با فرض اینکه حلقه k بار تکرار شود (که k = n / 64)، پیچیدگی زمانی O(k) است که به O(n) ساده می شود.

بنابراین، پیچیدگی کلی زمانی اجرای SHA-1 O(n) است، که در آن n طول پیام ورودی است.

*	نام تابع	پیچیدگی زمانی تابع
۱	add	O (n)
۲	find	O (n)
۳	delete	O (n)
۴	size	O (1)
۵	dad_is_that_you mom_is_that_you bro_or_sis	O (1)
۶	relationship	O (n)
۷	same_ancestor	O (n^2)
۸	furthest	O (n^3)
۹	generations	O (n)

n : تعداد افراد موجود در خانواده