**Collision Hash Attack**

**Description 4**

The product uses a hashing algorithm that produces a hash value that can be used to determine the original input, or to find an input that can produce the same hash, more efficiently than brute force techniques.

This weakness is especially dangerous when the hash is used in security algorithms that require the one-way property to hold. For example, if an authentication system takes an incoming password and generates a hash, then compares the hash to another hash that it has stored in its authentication database, then the ability to create a collision could allow an attacker to provide an alternate password that produces the same target hash, bypassing authentication. **The common consequence of this attack in the scope of access control scope is bypassing protection mechanisms. [4]**

**Description 1**

A **Collision Attack** is an attempt to find two input strings of a hash function that produce the same hash result. Because hash functions have infinite input length and a predefined output length, there is inevitably going to be the possibility of two different inputs that produce the same output hash. If two separate inputs produce the same hash output, it is called a collision. **This collision can then be exploited by any application that compares two hashes together – such as password hashes, file integrity checks, etc.**

Practically speaking, there are several ways a hash collision could be exploited. **if the attacker was offering a file download and showed the hash to prove the file’s integrity, he could switch out the file download for a different file that had the same hash, and the person downloading it would be unable to know the difference. The file would appear valid as it has the same hash as the supposed real file. [1]**

**Description 2**

In cryptography, a **collision attack** on a [cryptographic hash](https://en.wikipedia.org/wiki/Cryptographic_hash) tries to find two inputs producing the same hash value, i.e. a [hash collision](https://en.wikipedia.org/wiki/Hash_collision). ( i.e. Find two different messages *m1* and *m2* such that *hash(m1)* = *hash(m2)*. )

Digital signature schemes are often vulnerable to hash collisions. The usual attack scenario goes like this: [2]

1. Mallory creates two different documents A and B that have an identical hash value, i.e., a collision. Mallory seeks to deceive Bob into accepting document B, ostensibly from Alice.
2. Mallory **sends document A to Alice**, who agrees to what the document says, signs its hash, and sends the signature to Mallory.
3. Mallory attaches the signature from document A to document B.
4. Mallory then **sends the signature and document B to Bob**, claiming that Alice signed B. Because the digital signature matches document B's hash, Bob's software is unable to detect the substitution.

**Description 3 (relationship between collision attack and repudiation attacks)**

The basic idea behind the collision attack on a hash algorithm used in a digital-signature protocol is that the attacker creates two messages that have the same hash value, causes one of them to be signed, and then uses that signature over the other message for some nefarious purpose. The specifics of the attack depend on the

protocol being used and what the victim does when presented with the signed message.

**The canonical example is where you create two messages, one of which says "I will pay $10 for doing this job" and the other of which says “I will pay $10,000 for doing this job". You present the first message to the victim, get them to sign it, do the job, substitute the second message in the signed authorization, present the altered signed message (whose signature still verifies), and demand the higher amount of money. If the victim refuses, you take them to court and show the second signed message.** [3]

**Refrences**

**[1]** <https://learncryptography.com/hash-functions/hash-collision-attack>

**[2]** <https://en.wikipedia.org/wiki/Collision_attack>

**[3]** [**https://tools.ietf.org/html/rfc4270**](https://tools.ietf.org/html/rfc4270)

**[4]** <https://cwe.mitre.org/data/definitions/328.html>