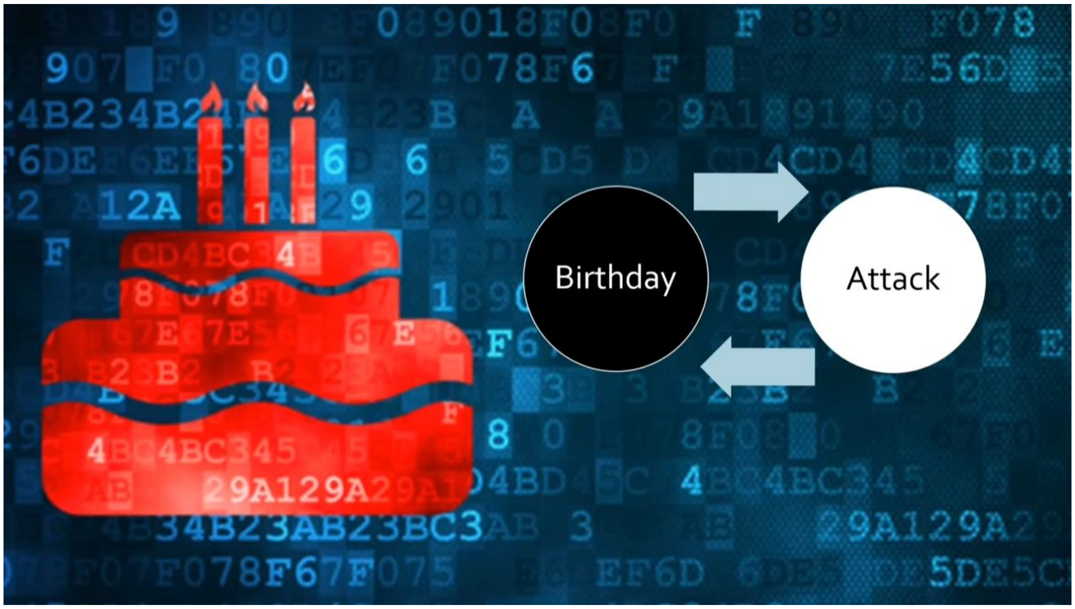
View, add and edit your notes in the app

# Birthday Attack in Cryptography | How to attack a Person | Explained In Hindi | AR Network

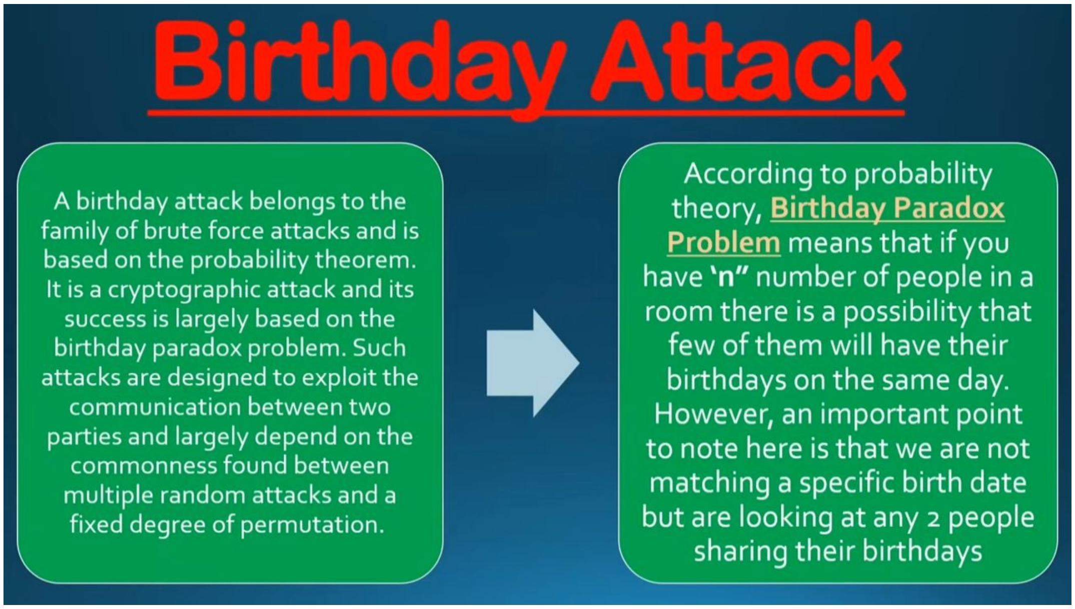
Generated on December 20, 2023

## Summary

| Notes | Screenshots | Bookmarks |
|---|---|---|
| 0 | 6 | 0 |



▷ 0:14

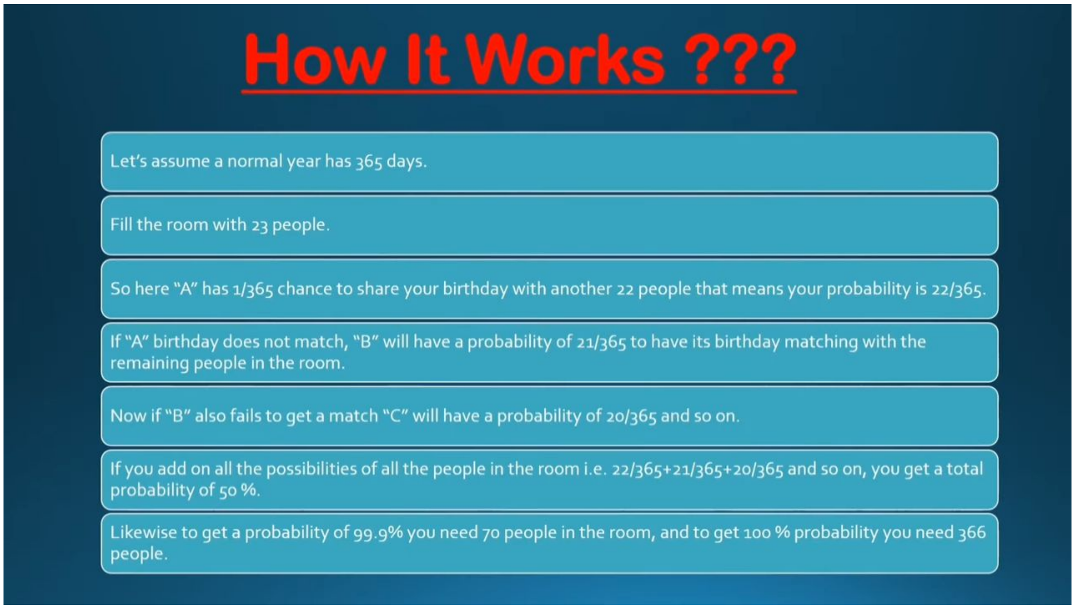


# Birthday Attack

A birthday attack belongs to the family of brute force attacks and is based on the probability theorem. It is a cryptographic attack and its success is largely based on the birthday paradox problem. Such attacks are designed to exploit the communication between two parties and largely depend on the commonness found between multiple random attacks and a fixed degree of permutation.

According to probability theory, Birthday Paradox Problem means that if you have **'n"** number of people in a room there is a possibility that few of them will have their birthdays on the same day. However, an important point to note here is that we are not matching a specific birth date but are looking at any 2 people sharing their birthdays

▷ 0:36



# How It Works ???

Let's assume a normal year has 365 days.

Fill the room with 23 people.

So here "A" has 1/365 chance to share your birthday with another 22 people that means your probability is 22/365.

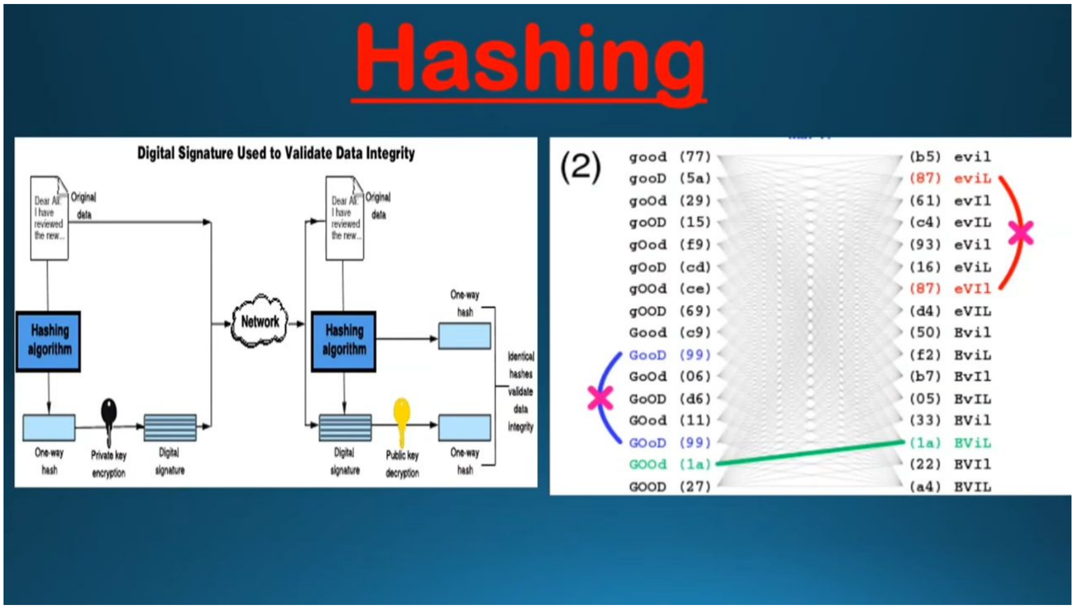
If "A" birthday does not match, "B" will have a probability of 21/365 to have its birthday matching with the remaining people in the room.

Now if "B" also fails to get a match "C" will have a probability of 20/365 and so on.

If you add on all the possibilities of all the people in the room i.e. 22/365+21/365+20/365 and so on, you get a total probability of 50 %.

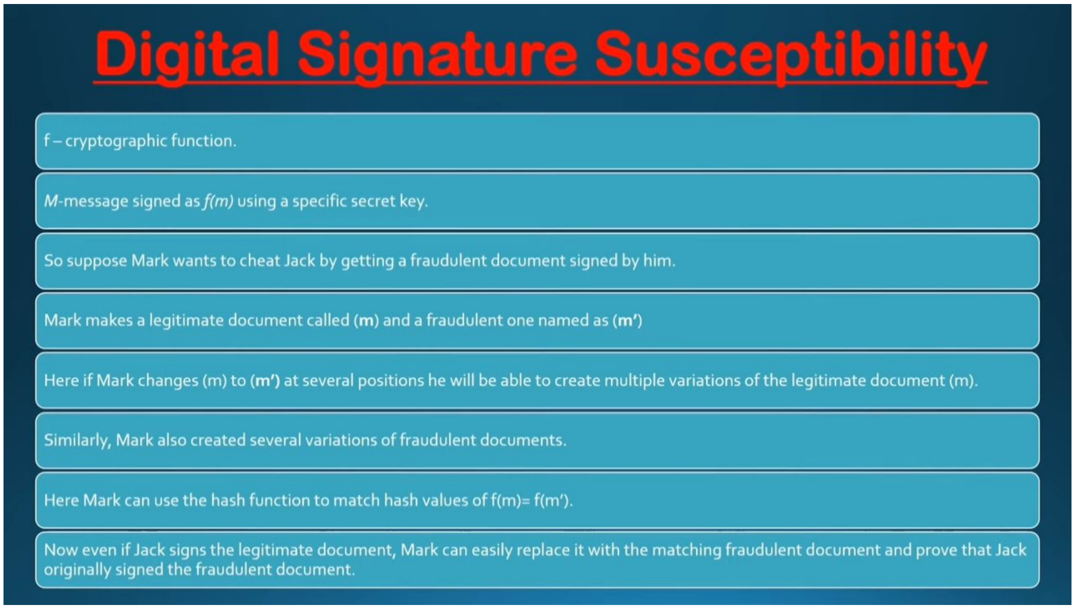
Likewise to get a probability of 99.9% you need 70 people in the room, and to get 100 % probability you need 366 people.

▷ 2:33



# Hashing

▷ 5:14



# Digital Signature Susceptibility

f – cryptographic function.

*M*-message signed as *f(m)* using a specific secret key.

So suppose Mark wants to cheat Jack by getting a fraudulent document signed by him.

Mark makes a legitimate document called (**m**) and a fraudulent one named as (**m’**)

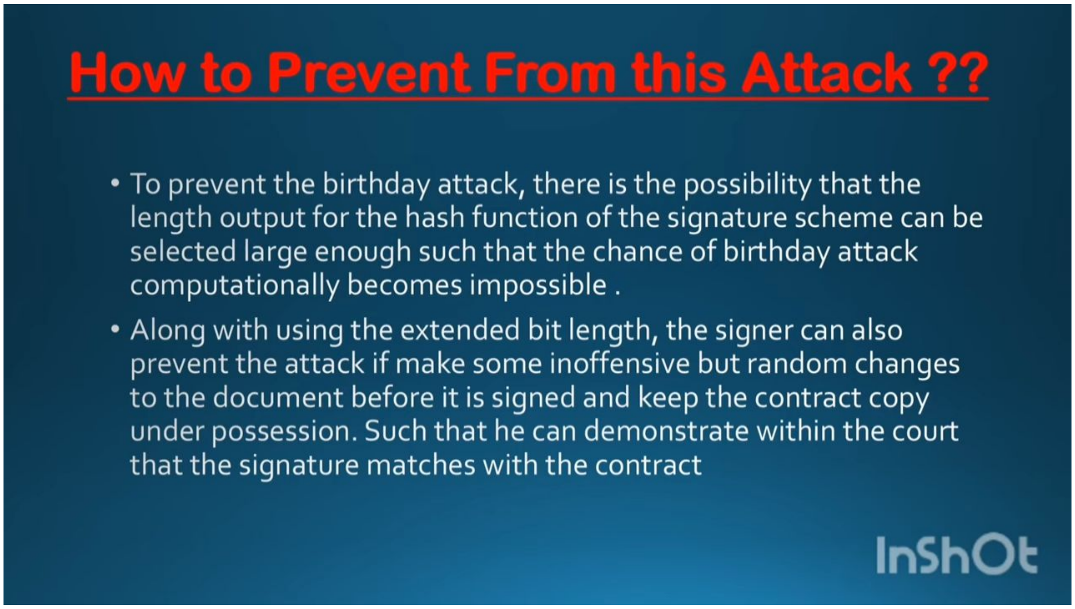
Here if Mark changes (m) to (**m’**) at several positions he will be able to create multiple variations of the legitimate document (m).

Similarly, Mark also created several variations of fraudulent documents.

Here Mark can use the hash function to match hash values of f(m)= f(m’).

Now even if Jack signs the legitimate document, Mark can easily replace it with the matching fraudulent document and prove that Jack originally signed the fraudulent document.

▷ 6:19



# How to Prevent From this Attack ??

- To prevent the birthday attack, there is the possibility that the length output for the hash function of the signature scheme can be selected large enough such that the chance of birthday attack computationally becomes impossible .
- Along with using the extended bit length, the signer can also prevent the attack if make some inoffensive but random changes to the document before it is signed and keep the contract copy under possession. Such that he can demonstrate within the court that the signature matches with the contract

▷ 7:50