

# Forensic Accounting Tools

CA Prerak Shah

prerako3@gmail.com | 91-9714744995



# Hashing

For file Integrity Checks



# Using HashCheck

\*

## What is Hashing ?

- Hashing is an algorithm that calculates a fixed-size bit string value from a file. A file basically contains blocks of data. Hashing transforms this data into a **far shorter fixed-length value** or key which represents the original string. The hash value can be considered the **distilled summary** of everything within that file.
- A good hashing algorithm would exhibit a property called the **avalanche effect**, where the resulting hash output would change significantly or entirely even when a single bit or byte of data within a file is changed. A hash function that does not do this is considered to have poor randomization, which would be easy to break by hackers.
- A hash is usually a hexadecimal string of several characters. Hashing is also a **unidirectional process** so you can never work backwards to get back the original data.
- A good hash algorithm should be complex enough such that it does not produce the same hash value from two different inputs. If it does, this is known as a **hash collision**. A hash algorithm can only be considered good and acceptable if it can offer a very low chance of collision.

# Using HashCheck

\*

## What are the benefits of Hashing ?

- Hashing is also used **to verify the integrity of a file** after it has been transferred from one place to another. To ensure the transferred file is not corrupted, a user can compare the hash value of both files. If they are the same, then the transferred file is an identical copy. Many backup softwares use this technique to ensure file integrity.
- Hashing is also used **to compare two files for equality**. Without opening two document files to compare them word-for-word, the calculated hash values of these files will allow the owner to know immediately if they are different.
- In some situations, an encrypted file may be designed to never change the file size nor the last modification date and time (for example, virtual drive container files). In such cases, it would be impossible to tell **at a glance** if two similar files are different or not, but the hash values would easily tell these files apart if they are different.

# Using HashCheck

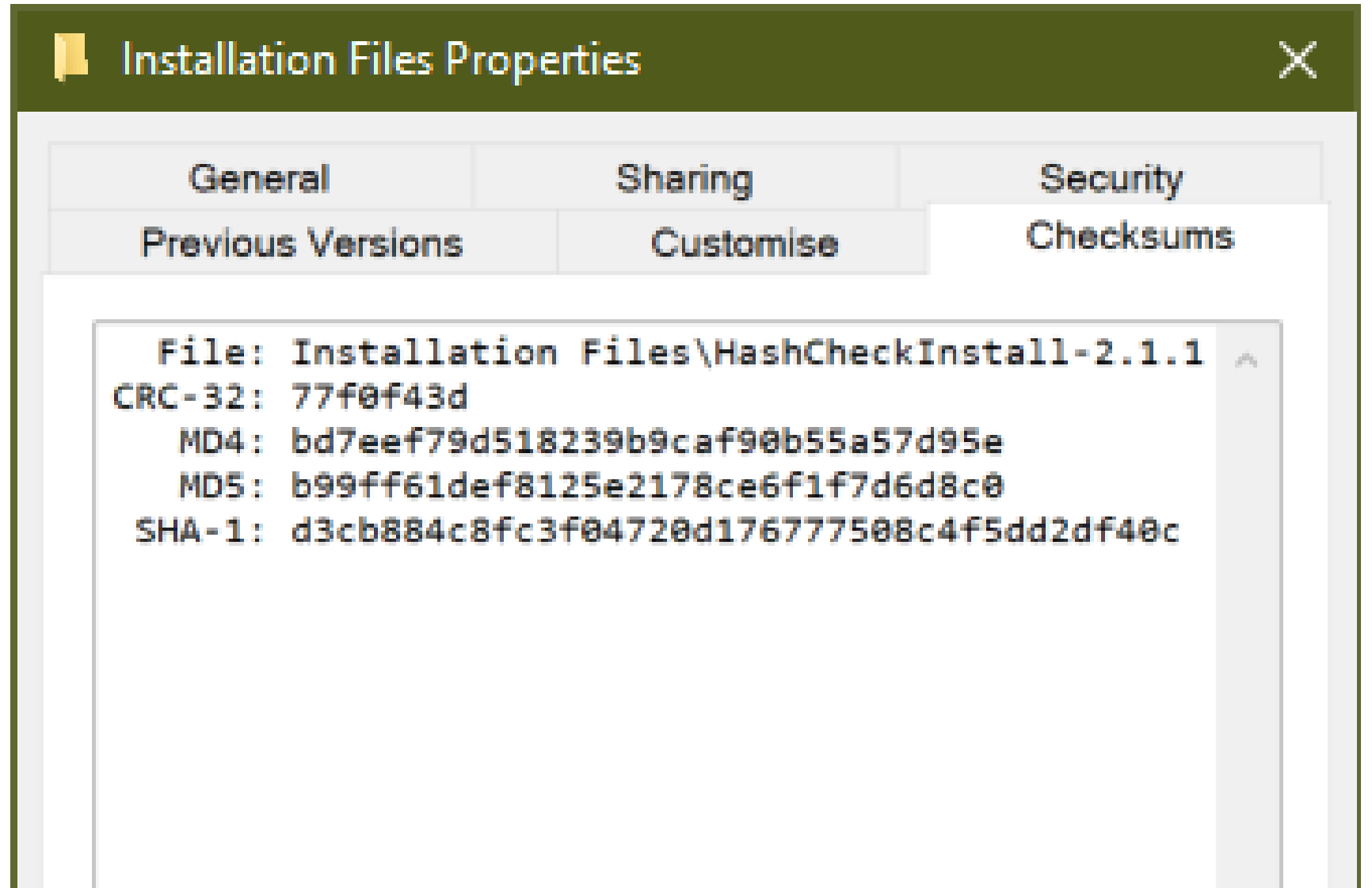
\*

What are the  
most common  
hashing  
algorithms ?

- **MD5** - An MD5 hash function encodes a string of information and encodes it into a 128-bit fingerprint. MD5 is often used as a checksum to verify data integrity. However, due to its age, MD5 is also **known to suffer from extensive hash collision** vulnerabilities, but it's still one of the most widely used algorithms in the world.
- **SHA-2** - SHA-2, developed by the National Security Agency (NSA), is a **cryptographic hash function**. SHA-2 includes significant changes from its predecessor, SHA-1. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: **SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256**.
- **CRC32** - A cyclic redundancy check (CRC) is an **error-detecting code** often used for detection of accidental changes to data. Encoding the same data string using CRC32 will always result in the same hash output, thus CRC32 is sometimes used as a hash algorithm for file integrity checks. These days, CRC32 is rarely used outside of Zip files and FTP servers

# HashCheck Utility

Integrates itself into Windows Shell



The screenshot shows a Windows dialog box titled "Installation Files Properties" with a close button (X) in the top right corner. The dialog has several tabs: "General", "Sharing", "Security", "Previous Versions", and "Customise". The "Security" tab is selected, and the "Checksums" sub-tab is active. The main content area displays the following information:

```
File: Installation Files\HashCheckInstall-2.1.1 ^
CRC-32: 77f0f43d
MD4: bd7eef79d518239b9caf90b55a57d95e
MD5: b99ff61def8125e2178ce6f1f7d6d8c0
SHA-1: d3cb884c8fc3f04720d176777508c4f5dd2df40c
```

# Using Precisa / Perfios

Illustrating Bank Statements Analyzer – Fundflow trail, etc.

(<https://webapp.precisa.in/login>)

## Using Precisa / Perfios

- These are AI / ML based bank statement spending analysers platforms basically used for risk profiling by financial institutions.





# Using Funds Trail Analysers

Illustrating major techniques used...



# Using Funds Trail Analysers

Feature	Descriptions
Fund-flow Report	Highlighting all transactions along with a visual representation
High Value Transactions Report	List of all transactions and entities, potential frauds or siphoning / diversion
Investments Report	Summarization of utilization of funds by suspect entities - purchase of stocks, mutual funds, investments by suspect.
Cash Transaction Report	Large cash transactions by suspects

# Using Funds Trail Analysers

Feature	Description
FIFO Report	Report investigators with the end utilization and beneficiary helping to evaluate justified use or diversion of funds
Related Party Transactions Report	Lists all RPTs transactions for crosschecking with proper approvals
Non-Bank Party Fund Flow Report	Both payment and receipt transactions along with its abnormal nature
Transaction Matching & Reconciliation Report	Alerting for unreconciled transactions between books and bank statements

# Using Funds Trail Analysers \* Sample Cases

Sample Case	Features
Diversion, Siphoning, Embezzlement	Fund-flow Report, High Value Transactions Report, Investments Report , Cash Transaction Report, Transaction Matching & Reconciliation Report, Non-Bank Party Fund Flow Report
Non Performing Assets	Fund-flow Report, FIFO Reports, Related Party Transactions, Cash Transaction Report
During Insolvency Proceedings	Fund-flow Report, Related Party Transactions, High Value Transactions Report, Transaction Matching & Reconciliation Report
Ponzi Schemes	Fund-flow Report, FIFO Report, High Value Transactions Report, Non-Bank Party Fund Flow Report
ASM audits	Fund-flow Report, FIFO Reports, High Value Transactions Report, Related Party Transactions, Cash Transaction Report, Non-Bank Party Fund Flow Report



# Using MsExcel

Illustrating Benford's Law



# Benford's Law

First Digit	Probability
1	30.1%
2	17.6%
3	12.5%
4	9.7%
5	7.9%
6	6.7%
7	5.8%
8	5.1%
9	4.6%
<b>Total</b>	<b>100%</b>

- Benford's law, also known as the "first-digit law," is a statistical phenomenon that describes the frequency distribution of the first digit in many naturally occurring datasets.
- According to Benford's law, the probability of a number's first digit being  $d$  (where  $d$  can be any digit from 1 to 9) is given by:
- $P(d) = \log_{10}(1 + 1/d)$
- This means that the digit 1 occurs as the first digit about 30% of the time, while the probability of the digit 9 appearing as the first digit is only about 5%.
- The law applies to a wide range of datasets, including financial statements, population numbers, and scientific data, among others. The datasets that follow Benford's law tend to exhibit a logarithmic distribution, meaning that the frequency of each digit is proportional to its logarithm.
- Benford's law has practical applications in a variety of fields, such as fraud detection, where deviations from the expected frequency distribution of digits can signal anomalies or irregularities in the data.



# Using Tableau

Illustrating Benford's Law

(<https://www.acfeinsights.com/acfe-insights/what-is-benfords-law>)



# Benford's Law

Table 1: Probabilities Predicted by Benford's Law for the First and Higher-Order Digits\*

$d_i$	$P(d_1)$	$P(d_2)$	$P(d_3)$	$P(d_4)$
0		0.11968	0.10178	0.10018
1	0.30103	0.11389	0.10138	0.10014
2	0.17609	0.10882	0.10097	0.10010
3	0.12494	0.10433	0.10057	0.10006
4	0.09691	0.10031	0.10018	0.10002
5	0.07918	0.09668	0.09979	0.09998
6	0.06695	0.09337	0.09940	0.09994
7	0.05799	0.09035	0.09902	0.09990
8	0.05115	0.08757	0.09864	0.09986
9	0.04576	0.08500	0.09827	0.09982

\*Figures are adapted from Nigrini 1996.



# Where to and where not to apply?

- **Distributions that can be expected to obey Benford's law**
  - When the mean is greater than the median and the skew is positive
  - Numbers that result from mathematical combination of numbers: e.g. quantity × price
  - **Transaction level** data: e.g. disbursements, sales
- **Distributions that would not be expected to obey Benford's law**
  - Where numbers are assigned **sequentially**: e.g. check numbers, invoice numbers
  - Where numbers are influenced by **human thought**: e.g. prices set by psychological thresholds (\$9.99)
  - Accounts with a large number of firm-specific numbers: e.g. accounts set up to record \$100 refunds
  - Accounts with a **built-in minimum or maximum**
  - **Distributions that do not span** an order of magnitude of numbers.



# Using Idea/ACL

Illustrating powerful in-built tools for auditing



# Using Idea/ACL

- Gap Detection
- Benford's Law
- Statistics
- Stratification
- Sampling
- Aging
- Charts
- Pivots
- Relationships
- Trail / Documentation
- Macros



# Other Tools

# Other Tools

- Fuzzy Lookup Excel Add-in
- PowerBI
- Python
- Generic Case Management Suite



# Thank You

:)

CA Prerak Shah

prerako3@gmail.com | 91-9714744995