Encryption, encoding and hashing, these terms are commonly interchanged and used incorrectly; knowing the differences, when and why to use each is important. Organizations have had breaches that sourced back to using the wrong data transforming method, and have gotten flack when using the terms incorrectly in their press releases as it indicates they are not knowledgeable and potentially careless with user data. A prime example was a breach Adobe suffered.

**Encoding**

Encoding data is a process involving changing data into a new format using a scheme. Encoding is a reversible process; data can be encoded to a new format and decoded to its original format. Encoding typically involves a publicly available scheme that is easily reversed. Encoding data is typically used to ensure integrity and usability of data and commonly used when data cannot be transferred in its current format between systems or applications.

Encoding is not used to protect or secure data because it is easy to reverse. An example of encoding is: Base64.

**Hashing**

Hashing involves computing a fixed-length mathematical summary of data, the input data can be any size. In contrast to encoding, hashing cannot be reversed. It is not possible to take a hash and convert it back to the original data. Hashing is commonly used to verify the integrity of data, commonly referred to as a checksum. If two pieces of identical data are hashed using the same hash function, the resulting hash will be identical. If the two pieces of data are different, the resulting hashes will be different and unique.

As an example, say Alice wants to send Bob a file and verify that Bob has the exact file and that no changes occurred in the transferring process. Alice will email Bob the file along with a hash of the file. After Bob downloads the file, he can verify the file is identical by performing a hash function on the file and verify the resulting hash is the same as Alice provided.

An example of a hash function is: SHA512

In addition to verifying the integrity of data, hashing is the recommended data transformation technique in authentication processes for computer systems and applications. It is recommended to never store passwords and instead store only the hash of the “salted password”. A salt is a random string appended to a password that only the authentication process system knows; this guarantees that if two users have the same password the stored hashes are different.

When a user inputs a password to a web application, the password is sent to the web server. The web server then appends the salt to the password and performs a hash function on the password and a salt and compares this output hash with the hash stored in the database for the user. If the hashes match for that user, the user is granted access. Hashing ensures in the event of a breach or malicious insider the original passwords can never be retrieved. Salting ensures that, if a breach does occur an attacker cannot determine which users have the same passwords.

**Encryption**

Encryption is the process of securely encoding data in such a way that only authorized users with a key or password can decrypt the data to reveal the original. There are two basic types of encryption; symmetric key and public key. In symmetric key, the same key is used to encrypt and decrypt data, like a password. In public key encryption, one key is used to encrypt data and a different key is used to decrypt the data.

Encryption is used when data needs to be protected so those without the decryption keys cannot access the original data. When data is sent to a website over HTTPS it is encrypted using the public key type. While encryption does involve encoding data, the two are not interchangeable terms; encryption is always used when referring to data that has been securely encoded. Encoding data is used when talking about data that is not securely encoded. An example of encryption is: AES 256.