**Blockchain Basics: Key Takeaways**

Below you will find a number of key points from this module. Defined terms are underlined.

**Algorithms and Techniques**

**Elliptic Curve Cryptography (ECC)** family of algorithms is used in Bitcoin as well as Ethereum Blockchain for generating the key pair.

**Rivest-Shamir-Adelman (RSA)** is a commonly used implementation of public-private key in many applications, except Blockchains because of its need for a more efficient and stronger algorithm.

**Hashing** transforms and maps an arbitrary length of input data value to a unique fixed length value.

The following are two basic requirements of a hash function.

* make certain that one cannot derive the original items hashed from the hash value.
* make sure that the hash value uniquely represents the original items hashed.

A combination of hashing and encryption are used for securing the various elements of the blockchain. Private-public key pair and hashing are important foundation concepts in decentralized networks that operate beyond the trust boundary.

**Asymmetric cryptography** uses public-private key pairs to encrypt and decrypt data.

**Essentials of Trust**

A **Merkle tree** is constructed by hashing paired data (the leaves), then pairing and hashing the results until a single hash remains

**Proof of work** is a protocol that has the main goal of deterring cyber-attacks such as a distributed denial-of-service attack (DDoS) which has the purpose of exhausting the resources of a computer system by sending multiple fake requests.

Well-defined processes for handling exceptions improve trust in the blockchain.

Forks are mechanisms that add to the robustness of the Blockchain framework.

Well-managed forks help build credibility in the blockchain by providing approaches to manage unexpected faults and planned improvements.

Soft fork and hard fork in the Blockchain world is like the release of software patches and new versions of operating systems respectively.

A **Soft Fork** is a fork where updated versions of the protocol are backwards compatible with previous versions.

A **Hard Fork** is a change of the protocol that is not backwards compatible with older versions of the client. Participants would absolutely need to upgrade their software in order to recognize new blocks.

**Ommer Blocks** contribute to the security of the main chain, but are not considered the canonical "truth" for that particular chain height.