

# Chain-key signatures

The main application of *chain-key signatures* is to enable direct interoperability with other blockchains as part of Chain Fusion Technology. Using chain-key signatures, canister smart contracts obtain control over cryptographic keys: They have a public key, from which a blockchain address on another blockchain like Bitcoin or Ethereum can be derived, and they can sign transactions relative to that address. Two major features of ICP that build on chain-key signatures are ckBTC and ckETH, representations of BTC and ETH on the Internet Computer, in which a canister smart contract cryptographically controls the bitcoin and ether that backs the ckBTC and ckETH tokens 1:1. Indeed, using chain-key signatures is the strongest, most decentralized way of integrating blockchains as no additional trust assumptions besides that of the two blockchains are required, particularly no additional parties that manage signature keys or their shares.

Just like chain-key technology, a key component of chain-key signatures is threshold cryptography. The threshold signature scheme used to implement chain-key cryptography is based on BLS signatures. While BLS signatures have distinct advantages, they are simply not compatible with other blockchains. In order to work with other blockchains, the IC must use threshold signatures that are compatible with the digital signature schemes of those other blockchains. By far, the most commonly used signature scheme used on other blockchains (including Bitcoin and Ethereum) is the ECDSA signature scheme. Because of this, *threshold ECDSA* signatures are currently supported on the IC, with implementations of other threshold signature schemes in the planning stages.

Canister smart contracts can have an ECDSA public key. The corresponding secret key is threshold-shared among the nodes of the subnet holding that canister smart contract. This is a prerequisite for the direct integration between the Internet Computer and Bitcoin and Ethereum.

Implementing a secure and efficient threshold signing protocol for ECDSA is much more challenging than for BLS signatures. While there has been a flurry of research on threshold ECDSA in recent years, none of these protocols meet the demanding requirements of the Internet Computer: they all either assume a *synchronous network* (meaning that the protocols will fail or become insecure if messages are unexpectedly delayed) or provide *no robustness* (meaning that the ability to produce signatures is completely lost if a *single* node should crash) or *both*. Neither of these assumptions are acceptable on the IC: security and liveness must hold even in an *asynchronous network* with many faulty nodes.

The DFINITY R&D team has designed, analyzed, and implemented a new threshold ECDSA signing protocol that works over an *asynchronous network* and is quite *robust* (it will still produce signatures if up to a third of the nodes in a subnet are crashed or corrupt) while still delivering acceptable performance. This signing protocol has been published in two research papers that describe the protocol in detail and prove the key elements of its security. The NNS DAO decided to adopt threshold ECDSA on the Internet Computer and to roll it out, such that canister smart contracts are able to have an ECDSA public key.