Module sui::ecdsa_k1

Error if the public key cannot be recovered from the signature.

Error if the signature is invalid.

Error if the public key is invalid.

Hash function name that are valid for ecrecover and secp256k1_verify.

@param signature: A 65-bytes signature in form (r, s, v) that is signed using Secp256k1. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/narwhal/blob/5d6f6df8ccee94446ff88786c0dbbc98be7cfc09/crypto/src/secp256k1.rs The accepted v values are {0, 1, 2, 3}. @param msg: The message that the signature is signed against, this is raw message without hashing. @param hash: The hash function used to hash the message when signing.

If the signature is valid, return the corresponding recovered Secpk256k1 public key, otherwise throw error. This is similar to ecrecover in Ethereum, can only be applied to Secp256k1 signatures. May abort with EFailToRecoverPubKey or EInvalidSignature .

@param pubkey: A 33-bytes compressed public key, a prefix either 0x02 or 0x03 and a 256-bit integer.

If the compressed public key is valid, return the 65-bytes uncompressed public key, otherwise throw error. May abort with EInvalidPubKey .

@param signature: A 64-bytes signature in form (r, s) that is signed using Secp256k1. This is an non-recoverable signature without recovery id. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/fastcrypto/blob/74aec4886e62122a5b769464c2bea5f803cf8ecc/fastcrypto/src/secp256k1/mod.rs#L193 @param public_key: The public key to verify the signature against @param msg: The message that the signature is signed against, this is raw message without hashing. @param hash: The hash function used to hash the message when signing.

If the signature is valid to the pubkey and hashed message, return true. Else false.

Constants

Error if the public key cannot be recovered from the signature.

```bash
```

Error if the signature is invalid.

```bash
```

Error if the public key is invalid.

```bash
```

Hash function name that are valid for ecrecover and secp256k1_verify.

```bash
```

```bash
```

@param signature: A 65-bytes signature in form (r, s, v) that is signed using Secp256k1. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/narwhal/blob/5d6f6df8ccee94446ff88786c0dbbc98be7cfc09/crypto/src/secp256k1.rs The accepted v values are {0, 1, 2, 3}. @param msg: The message that the signature is signed against, this is raw message without hashing. @param

hash: The hash function used to hash the message when signing.

If the signature is valid, return the corresponding recovered Secpk256k1 public key, otherwise throw error. This is similar to ecrecover in Ethereum, can only be applied to Secp256k1 signatures. May abort with EFailToRecoverPubKey or EInvalidSignature .

```bash

```

```bash

```

@param pubkey: A 33-bytes compressed public key, a prefix either 0x02 or 0x03 and a 256-bit integer.

If the compressed public key is valid, return the 65-bytes uncompressed public key, otherwise throw error. May abort with EInvalidPubKey .

```bash

```

```bash

```

@param signature: A 64-bytes signature in form (r, s) that is signed using Secp256k1. This is an non-recoverable signature without recovery id. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/fastcrypto/blob/74aec4886e62122a5b769464c2bea5f803cf8ecc/fastcrypto/src/secp256k1/mod.rs#L193 @param public_key: The public key to verify the signature against @param msg: The message that the signature is signed against, this is raw message without hashing. @param hash: The hash function used to hash the message when signing.

If the signature is valid to the pubkey and hashed message, return true. Else false.

```bash

```

```bash

```

Function

@param signature: A 65-bytes signature in form (r, s, v) that is signed using Secp256k1. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/narwhal/blob/5d6f6df8ccee94446ff88786c0dbbc98be7cfc09/crypto/src/secp256k1.rs The accepted v values are {0, 1, 2, 3}. @param msg: The message that the signature is signed against, this is raw message without hashing. @param hash: The hash function used to hash the message when signing.

If the signature is valid, return the corresponding recovered Secpk256k1 public key, otherwise throw error. This is similar to ecrecover in Ethereum, can only be applied to Secp256k1 signatures. May abort with EFailToRecoverPubKey or EInvalidSignature .

```bash

```

```bash

```

@param pubkey: A 33-bytes compressed public key, a prefix either 0x02 or 0x03 and a 256-bit integer.

If the compressed public key is valid, return the 65-bytes uncompressed public key, otherwise throw error. May abort with EInvalidPubKey .

```bash
```

```

```bash

```

@param signature: A 64-bytes signature in form (r, s) that is signed using Secp256k1. This is an non-recoverable signature without recovery id. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/fastcrypto/blob/74aec4886e62122a5b769464c2bea5f803cf8ecc/fastcrypto/src/secp256k1/mod.rs#L193 @param public_key: The public key to verify the signature against @param msg: The message that the signature is signed against, this is raw message without hashing. @param hash: The hash function used to hash the message when signing.

If the signature is valid to the pubkey and hashed message, return true. Else false.

```bash

```

```bash

```

Function

@param pubkey: A 33-bytes compressed public key, a prefix either 0x02 or 0x03 and a 256-bit integer.

If the compressed public key is valid, return the 65-bytes uncompressed public key, otherwise throw error. May abort with EInvalidPubKey .

```bash

```

```bash

```

@param signature: A 64-bytes signature in form (r, s) that is signed using Secp256k1. This is an non-recoverable signature without recovery id. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/fastcrypto/blob/74aec4886e62122a5b769464c2bea5f803cf8ecc/fastcrypto/src/secp256k1/mod.rs#L193 @param public_key: The public key to verify the signature against @param msg: The message that the signature is signed against, this is raw message without hashing. @param hash: The hash function used to hash the message when signing.

If the signature is valid to the pubkey and hashed message, return true. Else false.

```bash

```

```bash

```

Function

@param signature: A 64-bytes signature in form (r, s) that is signed using Secp256k1. This is an non-recoverable signature without recovery id. Reference implementation on signature generation using RFC6979: https://github.com/MystenLabs/fastcrypto/blob/74aec4886e62122a5b769464c2bea5f803cf8ecc/fastcrypto/src/secp256k1/mod.rs#L193 @param public_key: The public key to verify the signature against @param msg: The message that the signature is signed against, this is raw message without hashing. @param hash: The hash function used to hash the message when signing.

If the signature is valid to the pubkey and hashed message, return true. Else false.

```bash

```
```

```bash
```