

Discover Packages > Standard library > crypto > ecdh 

ecdh





package

standard library

Version: [go1.20.1](#) **Latest** | Published: Feb 14, 2023 | License: [BSD-3-Clause](#) | Imports: [11](#) |

Imported by: [11](#)

Details

- ✓ Valid [go.mod](#) file 
- ✓ Redistributable license 
- ✓ Tagged version 
- ✓ Stable version 



[Learn more](#)

Repository

cs.opensource.google/go/go

Links

 [Report a Vulnerability](#)

 Documentation 

<> Documentation

Overview

Package ecdh implements Elliptic Curve Diffie-Hellman over NIST curves and Curve25519.

Index

type Curve

- [func P256\(\) Curve](#)
- [func P384\(\) Curve](#)
- [func P521\(\) Curve](#)
- [func X25519\(\) Curve](#)

type PrivateKey

- [func \(k *PrivateKey\) Bytes\(\) \[\]byte](#)
- [func \(k *PrivateKey\) Curve\(\) Curve](#)
- [func \(k *PrivateKey\) ECDH\(remote *PublicKey\) \(\[\]byte, error\)](#)
- [func \(k *PrivateKey\) Equal\(x crypto.PrivateKey\) bool](#)
- [func \(k *PrivateKey\) Public\(\) crypto.PublicKey](#)
- [func \(k *PrivateKey\) PublicKey\(\) *PublicKey](#)

type PublicKey

- [func \(k *PublicKey\) Bytes\(\) \[\]byte](#)
- [func \(k *PublicKey\) Curve\(\) Curve](#)
- [func \(k *PublicKey\) Equal\(x crypto.PublicKey\) bool](#)

Constants

This section is empty.

Variables

This section is empty.

Functions

This section is empty.

Types

type **Curve**

```
type Curve interface {
    // GenerateKey generates a new PrivateKey from rand.
    GenerateKey(rand io.Reader) (*PrivateKey, error)

    // NewPrivateKey checks that key is valid and returns a PrivateKey.
    //
    // For NIST curves, this follows SEC 1, Version 2.0, Section 2.3.6, which
    // amounts to decoding the bytes as a fixed length big endian integer and
    // checking that the result is lower than the order of the curve. The zero
    // private key is also rejected, as the encoding of the corresponding public
    // key would be irregular.
    //
    // For X25519, this only checks the scalar length.
    NewPrivateKey(key []byte) (*PrivateKey, error)

    // NewPublicKey checks that key is valid and returns a PublicKey.
    //
    // For NIST curves, this decodes an uncompressed point according to SEC 1,
    // Version 2.0, Section 2.3.4. Compressed encodings and the point at
    // infinity are rejected.
    //
    // For X25519, this only checks the u-coordinate length. Adversarially
    // selected public keys can cause ECDH to return an error.
    NewPublicKey(key []byte) (*PublicKey, error)
    // contains filtered or unexported methods
}
```

func **P256**

```
func P256() Curve
```

P256 returns a Curve which implements NIST P-256 (FIPS 186-3, section D.2.3), also known as secp256r1 or prime256v1.

Multiple invocations of this function will return the same value, which can be used for equality checks and switch statements.

func P384

```
func P384() Curve
```

P384 returns a Curve which implements NIST P-384 (FIPS 186-3, section D.2.4), also known as secp384r1.

Multiple invocations of this function will return the same value, which can be used for equality checks and switch statements.

func P521

```
func P521() Curve
```

P521 returns a Curve which implements NIST P-521 (FIPS 186-3, section D.2.5), also known as secp521r1.

Multiple invocations of this function will return the same value, which can be used for equality checks and switch statements.

func X25519

```
func X25519() Curve
```

X25519 returns a Curve which implements the X25519 function over Curve25519 ([RFC 7748, Section 5](#)).

Multiple invocations of this function will return the same value, so it can be used for equality checks and switch statements.

type PrivateKey

```
type PrivateKey struct {  
    // contains filtered or unexported fields  
}
```

PrivateKey is an ECDH private key, usually kept secret.

These keys can be parsed with [crypto/x509.ParsePKCS8PrivateKey](#) and encoded with [crypto/x509.MarshalPKCS8PrivateKey](#). For NIST curves, they then need to be converted with [crypto/ecdsa.PrivateKey.ECDH](#) after parsing.

func (*PrivateKey) Bytes

```
func (k *PrivateKey) Bytes() []byte
```

Bytes returns a copy of the encoding of the private key.

func (*PrivateKey) Curve

```
func (k *PrivateKey) Curve() Curve
```

func (*PrivateKey) ECDH

```
func (k *PrivateKey) ECDH(remote *PublicKey) ([]byte, error)
```

ECDH performs a ECDH exchange and returns the shared secret.

For NIST curves, this performs ECDH as specified in SEC 1, Version 2.0, Section 3.3.1, and returns the x-coordinate encoded according to SEC 1, Version 2.0, Section 2.3.5. The result is never the point at infinity.

For X25519, this performs ECDH as specified in [RFC 7748, Section 6.1](#). If the result is the all-zero value, ECDH returns an error.

func (*PrivateKey) Equal

```
func (k *PrivateKey) Equal(x crypto.PrivateKey) bool
```

Equal returns whether x represents the same private key as k.

Note that there can be equivalent private keys with different encodings which would return false from this check but behave the same way as inputs to ECDH.

This check is performed in constant time as long as the key types and their curve match.

func (*PrivateKey) Public

```
func (k *PrivateKey) Public() crypto.PublicKey
```

Public implements the implicit interface of all standard library private keys. See the docs of `crypto.PrivateKey`.

func (*PrivateKey) PublicKey

```
func (k *PrivateKey) PublicKey() *PublicKey
```

type PublicKey

```
type PublicKey struct {  
    // contains filtered or unexported fields  
}
```

PublicKey is an ECDH public key, usually a peer's ECDH share sent over the wire.

These keys can be parsed with [crypto/x509.ParsePKIXPublicKey](#) and encoded with [crypto/x509.MarshalPKIXPublicKey](#). For NIST curves, they then need to be converted with [crypto/ecdsa.PublicKey.ECDH](#) after parsing.

func (*PublicKey) Bytes

```
func (k *PublicKey) Bytes() []byte
```

Bytes returns a copy of the encoding of the public key.

func (*PublicKey) Curve

```
func (k *PublicKey) Curve() Curve
```

func (*PublicKey) Equal

```
func (k *PublicKey) Equal(x crypto.PublicKey) bool
```

Equal returns whether x represents the same public key as k.

Note that there can be equivalent public keys with different encodings which would return false from this check but behave the same way as inputs to ECDH.

This check is performed in constant time as long as the key types and their curve match.



Source Files

[View all](#) 

[ecdh.go](#)

[nist.go](#)

[x25519.go](#)

Why Go

[Use Cases](#)

[Case Studies](#)

Get Started

[Playground](#)

[Tour](#)

[Stack Overflow](#)

[Help](#)

Packages

[Standard Library](#)

[About Go Packages](#)

About

[Download](#)

[Blog](#)

[Issue Tracker](#)

[Release Notes](#)

[Brand Guidelines](#)

[Code of Conduct](#)

Connect

[Twitter](#)

[GitHub](#)

[Slack](#)

[r/golang](#)

[Meetup](#)

[Copyright](#)

[Terms of Service](#)

[Privacy Policy](#)

[Report an Issue](#)

