ed25519

package | standard library

Version: go1.20.1 Latest | Published: Feb 14, 2023 | License: BSD-3-Clause | Imports: 8 | Imported by: 2,874

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 ed25519 implements the Ed25519 signature algorithm. See https://ed25519.cr.yp.to/.

These functions are also compatible with the “Ed25519” function defined in RFC 8032. However, unlike RFC 8032's formulation, this package's private key representation includes a public key suffix to make multiple signing operations with the same key more efficient. This package refers to the RFC 8032 private key as the “seed”.

▶ Example (Ed25519ctx)

Index

func (priv PrivateKey) Sign(rand io.Reader, message []byte, opts crypto.SignerOpts) (signature []byte, err error)

type PublicKey

func (pub PublicKey) Equal(x crypto.PublicKey) bool

Examples

Package (Ed25519ctx)

Constants

View Source

```
const (
    // PublicKeySize is the size, in bytes, of public keys as used in this package.
    PublicKeySize = 32
    // PrivateKeySize is the size, in bytes, of private keys as used in this package.
    PrivateKeySize = 64
    // SignatureSize is the size, in bytes, of signatures generated and verified by this
    SignatureSize = 64
    // SeedSize is the size, in bytes, of private key seeds. These are the private key
    SeedSize = 32
)
```

Variables

This section is empty.

Functions

func GenerateKey

```
func GenerateKey(rand io.Reader) (PublicKey, PrivateKey, error)
```

GenerateKey generates a public/private key pair using entropy from rand. If rand is nil, crypto/rand.Reader will be used.

func Sign

```
func Sign(privateKey PrivateKey, message []byte) []byte
```

Sign signs the message with privateKey and returns a signature. It will panic if len(privateKey) is not PrivateKeySize.

func Verify

```
func Verify(publicKey PublicKey, message, sig []byte) bool
```

Verify reports whether sig is a valid signature of message by publicKey. It will panic if len(publicKey) is not PublicKeySize.

func VerifyWithOptions

added in go1.20

```
func VerifyWithOptions(publicKey PublicKey, message, sig []byte, opts *Options) error
```

VerifyWithOptions reports whether sig is a valid signature of message by publicKey. A valid signature is indicated by returning a nil error. It will panic if len(publicKey) is not PublicKeySize.

If opts.Hash is crypto.SHA512, the pre-hashed variant Ed25519ph is used and message is expected to be a SHA-512 hash, otherwise opts.Hash must be crypto.Hash(0) and the message must not be hashed, as Ed25519 performs two passes over messages to be signed.

Types

type Options

added in go1.20

```
type Options struct {
    // Hash can be zero for regular Ed25519, or crypto.SHA512 for Ed25519ph.
    Hash crypto.Hash

    // Context, if not empty, selects Ed25519ctx or provides the context string
    // for Ed25519ph. It can be at most 255 bytes in length.
    Context string
}
```

Options can be used with PrivateKey.Sign or VerifyWithOptions to select Ed25519 variants.

func (*Options) HashFunc

added in go1.20

```
func (o *Options) HashFunc() crypto.Hash
```

HashFunc returns o.Hash.

type PrivateKey

```
type PrivateKey []byte
```

PrivateKey is the type of Ed25519 private keys. It implements crypto.Signer.

func NewKeyFromSeed

```
func NewKeyFromSeed(seed []byte) PrivateKey
```

NewKeyFromSeed calculates a private key from a seed. It will panic if len(seed) is not SeedSize. This function is provided for interoperability with RFC 8032. RFC 8032's private keys correspond to seeds in this package.

func (PrivateKey) Equal

added in go1.15

```
func (priv PrivateKey) Equal(x crypto.PrivateKey) bool
```

Equal reports whether priv and x have the same value.

func (PrivateKey) Public

```
func (priv PrivateKey) Public() crypto.PublicKey
```

Public returns the PublicKey corresponding to priv.

func (PrivateKey) Seed

```
func (priv PrivateKey) Seed() []byte
```

Seed returns the private key seed corresponding to priv. It is provided for interoperability with RFC 8032. RFC 8032's private keys correspond to seeds in this package.

func (PrivateKey) Sign

```
func (priv PrivateKey) Sign(rand io.Reader, message []byte, opts crypto.SignerOpts) (s
ignature []byte, err error)
```

Sign signs the given message with priv. rand is ignored.

If opts.HashFunc() is crypto.SHA512, the pre-hashed variant Ed25519ph is used and message is expected to be a SHA-512 hash, otherwise opts.HashFunc() must be crypto.Hash(0) and the message must not be hashed, as Ed25519 performs two passes over messages to be signed.

A value of type Options can be used as opts, or crypto.Hash(0) or crypto.SHA512 directly to select plain Ed25519 or Ed25519ph, respectively.

type PublicKey

```
type PublicKey []byte
```

PublicKey is the type of Ed25519 public keys.

func (PublicKey) Equal

added in go1.15

```
func (pub PublicKey) Equal(x crypto.PublicKey) bool
```

Equal reports whether pub and x have the same value.

Source Files

View all

ed25519.go