

[Discover Packages](#) > [Standard library](#) > [crypto](#) > [dsa](#) 

dsa





package

standard library

Version: [go1.20.1](#) **Latest** | Published: Feb 14, 2023 | License: [BSD-3-Clause](#) | Imports: 4 |

Imported by: 3,037

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 dsa implements the Digital Signature Algorithm, as defined in FIPS 186-3.

The DSA operations in this package are not implemented using constant-time algorithms.

Deprecated: DSA is a legacy algorithm, and modern alternatives such as Ed25519 (implemented by package crypto/ed25519) should be used instead. Keys with 1024-bit moduli (L1024N160 parameters) are cryptographically weak, while bigger keys are not widely supported. Note that FIPS 186-5 no longer approves DSA for signature generation.

Index

Variables

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

```
func GenerateParameters(params *Parameters, rand io.Reader, sizes ParameterSizes) error
```

```
func Sign(rand io.Reader, priv *PrivateKey, hash []byte) (r, s *big.Int, err error)
```

```
func Verify(pub *PublicKey, hash []byte, r, s *big.Int) bool
```

```
type ParameterSizes
```

```
type Parameters
```

```
type PrivateKey
```

```
type PublicKey
```

Constants

This section is empty.

Variables

[View Source](#)

```
var ErrInvalidPublicKey = errors.New("crypto/dsa: invalid public key")
```

ErrInvalidPublicKey results when a public key is not usable by this code. FIPS is quite strict about the format of DSA keys, but other code may be less so. Thus, when using keys which may have been generated by other code, this error must be handled.

Functions

func GenerateKey

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

GenerateKey generates a public&private key pair. The Parameters of the PrivateKey must already be valid (see GenerateParameters).

func GenerateParameters

```
func GenerateParameters(params *Parameters, rand io.Reader, sizes ParameterSizes) error
```

GenerateParameters puts a random, valid set of DSA parameters into params. This function can take many seconds, even on fast machines.

func Sign

```
func Sign(rand io.Reader, priv *PrivateKey, hash []byte) (r, s *big.Int, err error)
```

Sign signs an arbitrary length hash (which should be the result of hashing a larger message) using the private key, priv. It returns the signature as a pair of integers. The security of the private key depends on the entropy of rand.

Note that FIPS 186-3 section 4.6 specifies that the hash should be truncated to the byte-length of the subgroup. This function does not perform that truncation itself.

Be aware that calling Sign with an attacker-controlled PrivateKey may require an arbitrary amount of CPU.

func Verify

```
func Verify(pub *PublicKey, hash []byte, r, s *big.Int) bool
```

Verify verifies the signature in r, s of hash using the public key, pub. It reports whether the signature is valid.

Note that FIPS 186-3 section 4.6 specifies that the hash should be truncated to the byte-length of the subgroup. This function does not perform that truncation itself.

Types

type ParameterSizes

```
type ParameterSizes int
```

ParameterSizes is an enumeration of the acceptable bit lengths of the primes in a set of DSA parameters. See FIPS 186-3, section 4.2.

```
const (  
    L1024N160 ParameterSizes = iota  
    L2048N224  
    L2048N256  
    L3072N256  
)
```

type Parameters

```
type Parameters struct {  
    P, Q, G *big.Int  
}
```

Parameters represents the domain parameters for a key. These parameters can be shared across many keys. The bit length of Q must be a multiple of 8.

type PrivateKey

```
type PrivateKey struct {  
    PublicKey  
    X *big.Int  
}
```

PrivateKey represents a DSA private key.

type PublicKey

```
type PublicKey struct {  
    Parameters  
    Y *big.Int  
}
```

PublicKey represents a DSA public key.



Source Files

[View all](#)

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](#)

[Golang Weekly](#)

[Copyright](#)

[Terms of Service](#)

[Privacy Policy](#)

[Report an Issue](#)



Google