## NAME

EVP_KDF−SSHKDF − The SSHKDF EVP_KDF implementation

## DESCRIPTION

Support for computing the **SSHKDF** KDF through the **EVP_KDF** API.

The EVP_KDF−SSHKDF algorithm implements the SSHKDF key derivation function. It is defined in RFC 4253, section 7.2 and is used by SSH to derive IVs, encryption keys and integrity keys. Five inputs are required to perform key derivation: The hashing function (for example SHA256), the Initial Key, the Exchange Hash, the Session ID, and the derivation key type.

### Identity

''SSHKDF'' is the name for this implementation; it can be used with the **EVP_KDF_fetch()** function.

### Supported parameters

The supported parameters are:

''properties'' (**OSSL_KDF_PARAM_PROPERTIES**) <UTF8 string>
''digest'' (**OSSL_KDF_PARAM_DIGEST**) <UTF8 string>
''key'' (**OSSL_KDF_PARAM_KEY**) <octet string>
These parameters work as described in ''PARAMETERS'' in **EVP_KDF** (3).

''xcghash'' (**OSSL_KDF_PARAM_SSHKDF_XCGHASH**) <octet string>
''session_id'' (**OSSL_KDF_PARAM_SSHKDF_SESSION_ID**) <octet string>
These parameters set the respective values for the KDF. If a value is already set, the contents are replaced.

''type'' (**OSSL_KDF_PARAM_SSHKDF_TYPE**) <UTF8 string>
This parameter sets the type for the SSHKDF operation. There are six supported types:

EVP_KDF_SSHKDF_TYPE_INITIAL_IV_CLI_TO_SRV
The Initial IV from client to server. A single char of value 65 (ASCII char 'A').

EVP_KDF_SSHKDF_TYPE_INITIAL_IV_SRV_TO_CLI
The Initial IV from server to client A single char of value 66 (ASCII char 'B').

EVP_KDF_SSHKDF_TYPE_ENCRYPTION_KEY_CLI_TO_SRV
The Encryption Key from client to server A single char of value 67 (ASCII char 'C').

EVP_KDF_SSHKDF_TYPE_ENCRYPTION_KEY_SRV_TO_CLI
The Encryption Key from server to client A single char of value 68 (ASCII char 'D').

EVP_KDF_SSHKDF_TYPE_INTEGRITY_KEY_CLI_TO_SRV
The Integrity Key from client to server A single char of value 69 (ASCII char 'E').

EVP_KDF_SSHKDF_TYPE_INTEGRITY_KEY_SRV_TO_CLI
The Integrity Key from client to server A single char of value 70 (ASCII char 'F').

## NOTES

A context for SSHKDF can be obtained by calling:

```
EVP_KDF *kdf = EVP_KDF_fetch(NULL, "SSHKDF", NULL);
EVP_KDF_CTX *kctx = EVP_KDF_CTX_new(kdf);
```

The output length of the SSHKDF derivation is specified via the *keylen* parameter to the **EVP_KDF_derive** (3) function. Since the SSHKDF output length is variable, calling **EVP_KDF_CTX_get_kdf_size** (3) to obtain the requisite length is not meaningful. The caller must allocate a buffer of the desired length, and pass that buffer to the **EVP_KDF_derive** (3) function along with the desired length.

## EXAMPLES

This example derives an 8 byte IV using SHA−256 with a 1K ''key'' and appropriate ''xcghash'' and ''session_id'' values:

```
EVP_KDF *kdf;
EVP_KDF_CTX *kctx;
const char type = EVP_KDF_SSHKDF_TYPE_INITIAL_IV_CLI_TO_SRV;
unsigned char key[1024] = "01234...";
unsigned char xcghash[32] = "012345...";
unsigned char session_id[32] = "012345...";
unsigned char out[8];
size_t outlen = sizeof(out);
OSSL_PARAM params[6], *p = params;

kdf = EVP_KDF_fetch(NULL, "SSHKDF", NULL);
kctx = EVP_KDF_CTX_new(kdf);
EVP_KDF_free(kdf);

*p++ = OSSL_PARAM_construct_utf8_string(OSSL_KDF_PARAM_DIGEST,
                                        SN_sha256, strlen(SN_sha256));
*p++ = OSSL_PARAM_construct_octet_string(OSSL_KDF_PARAM_KEY,
                                         key, (size_t)1024);
*p++ = OSSL_PARAM_construct_octet_string(OSSL_KDF_PARAM_SSHKDF_XCGHASH,
                                         xcghash, (size_t)32);
*p++ = OSSL_PARAM_construct_octet_string(OSSL_KDF_PARAM_SSHKDF_SESSION_ID,
                                         session_id, (size_t)32);
*p++ = OSSL_PARAM_construct_utf8_string(OSSL_KDF_PARAM_SSHKDF_TYPE,
                                        &type, sizeof(type));
*p = OSSL_PARAM_construct_end();
if (EVP_KDF_derive(kctx, out, &outlen, params) <= 0)
    /* Error */
```

## CONFORMING TO

RFC 4253

## SEE ALSO

**EVP_KDF** (3), **EVP_KDF_CTX_new** (3), **EVP_KDF_CTX_free** (3), **EVP_KDF_CTX_set_params** (3), **EVP_KDF_CTX_get_kdf_size** (3), **EVP_KDF_derive** (3), ‘‘PARAMETERS’’ in **EVP_KDF** (3)

## COPYRIGHT