Network Working Group
Internet-Draft
Expires: August 28, 2006

M. Baugher
D. McGrew
Cisco Systems, Inc.
February 24, 2006

Diffie-Hellman Exchanges for Multimedia Sessions
draft-baugher-mmusic-sdp-dh-00.txt

Status of this Memo

Copyright Notice

Abstract

This memo defines a new Session Description Protocol (SDP) attribute for exchanging Diffie-Hellman (DH) public keys. The attribute is an SDP session-level attribute for describing DH keys, and there is a new media-level parameter for describing public keying material for SRTP key generation. The SDP attribute supports the key establishment schemes of NIST Draft Special Publication 800-56, adds domain parameters and supports external authentication of the DH endpoint without a public key infrastructure.

Table of Contents

1. Introduction: SDP Discrete Logarithm Cryptography (DLC)

RFC {sdesc} allows keys and parameters to be signaled in Session Description Protocol (SDP) by a media-level attribute. Called "crypto", this attribute establishes keys between endpoints and defines "crypto suites" for SRTP [sdesc]. However, RFC {sdesc} currently uses only symmetric cryptography and lacks a public key mechanism such as a Diffie-Hellman (DH) key exchange. This document adds a DH exchange to SDP to improve security in several ways.

1. It eliminates the need to trust the signaling devices (e.g. SIP proxies).

2. It reduces or eliminates the need for public-key infrastructure, as is needed when protecting RFC {sdesc} symmetric keys with S/MIME [RFC3851].

3. It provides better security in the presence of forked signaling.

4. It can provide perfect forward secrecy to media keys.

To these ends, DH offers several advantages over RFC {sdesc} methods. When SIP forking occurs[RFC3261], RFC {sdesc} can potentially reveal the key to an untrusted party on one of the forked devices. For example, a phone call may be forked to an executive and a receptionist. If the SDP crypto attribute contains a symmetric key, the receptionist is given enough information to perpetrate a passive eavesdropping attack on the caller. In contrast, an SDP Diffie-Hellman exchange (SDP-DH) provides a shared secret between the caller and each of the forked endpoints. With SDP-DH, the receptionist in our example would only know the public DH value of the caller and would not be able to perpetrate an attack.

1.1. Attacks and Protections

To prevent attacks during forking or any SDP message exchange, RFC {sdesc} has a stringent and challenging requirement: The SDP message needs end-to-end security when it contains a crypto attribute on any of its media lines. This attribute and its parameters may be secured end-to-end by S/MIME or by an end-to-end data security protocol such as TLS. These mechanisms are easy to provide in some situations, such as when all of the devices that handle the signaling are within a single trust domain. However, SIP generally does not have end-to-end transport connections between caller and callee, but uses SIPS for transport-level security. SIPS is a set of hop-by-hop TLS connections. There are no verifiable guarantees about the security at SIP hops that have access to the SIP message during forwarding. And, lacking the end-to-end integrity protection afforded by S/MIME,

the crypto attribute is vulnerable to both passive and active attacks by systems in the middle. An SDP Diffie-Hellman exchange provides an alternative to hop-by-hop transport security without requiring a public key infrastructure or S/MIME.

When the signaling message contains a DH key, it is sufficient to provide authentication without confidentiality on signaling traffic. The level of trust in the devices that handle the plaintext SDP message is reduced, since knowledge of the DH public key does not allow those devices to perform a passive eavesdropping attack. Certain devices in the signaling path can perform an active attack in the absence of end-to-end authentication. An active attack can happen when the attacker intercepts both signaling and media messages between the media endpoints and is able to decrypt, eavesdrop, and re-encrypt the media. However, the active attack is considerably more difficult than a passive attack.

Several existing methods can provide complete protection against active attacks. In cases where the signaling system is trusted to identify an endpoint's public key, the SIP Identity service [SIPidentity] can be used to authenticate the DH public keys. SIP Identity authenticates the SDP message so long as the phone users trust the SIP Identity provider. When properly authenticated by such means, SDP-DH establishes a pair-wise secret at each endpoint, and has no vulnerability to active attacks. When SIP Identity is unavailable, or when it is undesirable to trust that mechanism, it is possible to provide person-to-person authentication on the DH key strictly between the media endpoints. This method is used by the AT&T secure phone [Schneier], for example. In it, one person reads a fingerprint (hash) of the shared secret and the other person checks it against their hash. Following the process of checking a hash of the DH shared secret, each user can cache information about the public keys, allowing them to construct a "personal PKI" that is analogous to a PGP key ring [Zimmermann].

Finally, SDP-DH can provide perfect forward secrecy [DVW] to the cryptographic keys of SDP media sessions. This method protects recorded media sessions against future disclosure of either endpoint's private keys for ephemeral DH public keys. The media-stream key cannot be recovered so long as the ephemeral DH keying material that generated the secret is maintained according Section 5.6.4 of the Draft Recommendation [NIST-800-56].

Nonetheless, there are significant performance advantages to static keys over ephemeral keys for applications that wish to trade the benefits of perfect forward secrecy for the lower computational overhead that comes from reusing public-keys across multiple multimedia sessions. This memo therefore allows for both ephemeral

and static Diffie-Hellman key management schemes. A DH public key MAY be ephemeral or static.

To add Diffie-Hellman public keys to SDP messages, this memo introduces a new security attribute that is called "DH", which appears at the SDP session level. To allow SRTP to derive keys from the DH shared secret, this document defines a new media-level security parameter called "nonce", which carries public data for use in generating a pair-wise secret for the SRTP master key; the public data are salt, a nonce, optional key lifetime and optional SRTP master key index. The DH attribute and nonce parameter provide a framework for putting Diffie-Hellman key exchange protocols into SDP. Definitions are provided for IETF-standard Elliptic Curve Cryptography (ECC) [FS] and integer-based Finite Field Cryptography (FFC) [RFC3526][RFC4306]. An example DH attribute that uses a STAT_FFDH_Group_19 "DH suite", a DH dhkey parameter, and the media-level nonce parameter are shown in Figure 1.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
a=DH: STAT_ECDH_GROUP_19
 dhkey: 2tC2U5QiHPmwUeH+yleH0Jjf5jf8kLnvlF0MN3JYEYA=
         UnGgRhzbglLWHxxFb6PlmrH0WzOsz19YOJ4Fd7iZC7M=
m=video 51372 RTP/SAVP 31
a=crypto:1 AES_CM_128_HMAC_SHA1_80
 nonce:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:32
```

Figure 1: Example DH attribute and parameters.

1.2. Overview of This Document

Section 2 defines the SDP DH attribute shown in Figure 1 and it defines public key ("dhkey") encodings, offer/answer processing, and the SDP-DH grammar in Augmented Backus-Naur form (ABNF). Section 3 applies DH to the RFC{sdesc} media-level crypto attribute and gives the ABNF of a new crypto parameter called "nonce" (also shown in Figure 1) for generating media keys. The new parameter uses the output of SDP DH for input into SRTP master key generation according to Section 3. Section 4 defines a hash for person-to-person authentication. The Security Considerations of Section 5 discusses attacks and protections for DH suites, and Section 6 states the

requirements on IANA for the SDP DH attribute and the parameters defined herein.

1.3. Conformance Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. The SDP DH Attribute and Parameters

The SDP DH attribute has a discrete logarithm cryptography suite of parameters (DH suite) and an encoding of one public key. As shown in Figure 1, DH is an SDP session-level attribute whereas crypto is at the media level. The DH attribute applies to a multimedia session and the crypto nonce parameter applies to a single media session. The reason that the DH attribute operates at the SDP session level is to allow the computationally expensive DH exchange to be used across multiple media streams.

An SDP Offer/Answer exchange [RFC3264] SHALL be the default method of exchanging DH parameters between endpoints that compute a shared DH secret. SRTP key generation SHALL use the DH shared secret and public keying material. This exchange of DH public keys and keying material MUST be externally authenticated between the two parties. The authentication MAY be a person-to-person procedure or MAY use SIP Identity as described in the Security Considerations section.

The SDP DH attribute is a framework to support Diffie-Hellman key establishment "schemes" [NIST-800-56]. This document REQUIRES one key-establishment scheme as the basis for interoperability, however, and that is a static, Finite Field Cryptography (FFC) scheme. Stat_FFDH_Group_2 is the DH-suite name for the mandatory scheme. Stat_FFDH_Group_2 uses domain parameters from the IKEv2 MODP Group 2 [RFC4306]. Implementations of this specification MUST support at least the static FFC scheme. Note that an ephemeral FFC scheme can be implicitly supported with a static public key that is used exactly once. The key is more than nominally static, therefore, when it is reused over multiple multimedia sessions from a cache of public keys and shared secrets. Use of static keys trades perfect forward secrecy for savings in exponentiations and user effort, particularly when the user effort is an external person-to-person authentication "ceremony" [WE] or a check of SIP Identity. Applications that desire PFS MAY choose an ephemeral DH suite or MAY use a static public key only once.

The SDP DH attribute has two parameters, a "DH suite" and "dhkey".

1. A DH suite declares the key management scheme and parameters.

2. dhkey encodes a cryptographic key for the DH suite.

SDP DH has one output, the DH shared secret.

2.1. Mandatory DLC Suite: Stat_FFCDH_Group_2

In choosing a common DH suite for maximal interoperability, there are

trade offs in security, efficiency of computation, compactness of signaling parameters as well as known Intellectual Property Rights (IPR) claims. For Session Description Protocol signaling, compactness is a compelling requirement. From this perspective, the Elliptic Curve Cryptography Diffie-Hellman (ECC-DH) suites are attractive. But there are known IPR claims on ECC.

The Finite Field Cryptography (FFC) DH suites are without known IPR claims. The smallest group size acceptable to the Draft Recommendation is 1024 bits and this is chosen as the "Mandatory" DH suite, which is mandatory to implement but not mandatory to use. It is mandatory to implement so as to foster interoperable implementations that can always communicate using SDP DH. But the cryptographic strength of the 1024-bit DH Suite (Stat_FFCDH_Group_2) might be too weak for certain applications who MAY reject use of this suite in SDP DH offers.

Stat_FFCDH_Group_2 is based upon the "dhStatic" scheme of Table 5 in the Draft Recommendation [NIST-800-56].

2.1.1. IKE Group 2 Domain Parameters

The public key for Stat_FFCDH_Group_2 is a big integer that is 1024 bits in length and defined in IKEv2 [RFC4306]. The big prime integer field IKEv2 MODP Group 2 is quoted and copied below for the convenience of the reader. See the discussion in Appendix E of RFC 2412 for a description of how the primes were generated and suggestions on efficient implementations [RFC2412].

"This group is assigned id 2 (two).

The prime is 2^1024 - 2^960 - 1 + 2^64 * { [2^894 pi] + 129093 }. Its hexadecimal value is:

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08
8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B
302B0A6D F25F1437 4FE1356D 6D51C245 E485B576 625E7EC6 F44C42E9
A637ED6B 0BFF5CB6 F406B7ED EE386BFB 5A899FA5 AE9F2411 7C4B1FE6
49286651 ECE65381 FFFFFFFF FFFFFFFF
```

The generator is 2."

2.1.2. Encoding of the DHkey Parameter

An example DH attribute is shown below, the STAT_FFDH_GROUP_2 DH suite name string is followed by an example dhkey parameter.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
a=DH: STAT_FFDH_GROUP_2
 dhkey:
  f3kHGsXFPDWlPU7/9Vn+Elcb32j7X5xRVVbWZVN9xaTq9v0MiKCovTwVA6K/17SW0
  P3BrY3lGmkhJHRbmuqiurPLCBDGYqUdl8HLPlqX0Jd9qYJ58ILszdSgyrjnQzrLGu
  lgHoOQXeZef8SGEXY5qtaugNSg57Qxdn4e1MJQpAk=
m=video 51372 RTP/SAVP 31
a=crypto:1 AES_CM_128_HMAC_SHA1_80
 nonce:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:32
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
 nonce:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
m=application 32416 udp wb
a=orient:portrait
```

Figure 3: Example STAT_FFDH_Group_2 DLC Suite

An ABNF grammar for the DH attribute and dhkey parameter is given in the Grammar Section below. Stat_FFCDH_Group_2 has a public key value that is 172 bytes long when it is base-64 encoded. Some applications that use SDP over User Datagram Protocol (UDP) might have a problem with such a large field. The ECC DH suites have a much shorter public key and are described below.

2.1.3. Computation of the Shared Secret

Computation of the shared secret value Z for this DH suite is defined in Section 5.7.1.1 of the Draft Recommendation[NIST-800-56].

2.2. The Stat_ECDH_Group_19 DLC Suite

The Stat_ECDH_Group_19 key establishment scheme is based upon the "Cofactor Static Unified Model" of Table 5 in the Draft Recommendation [NIST-800-56]. This scheme uses two static keys, one at the offerer and one at the answerer but no ephemeral keys. "Stat_ECDH_Group_19_SHA256" is an alternative designation since SHA-256 [FIPS-180-2] is the hash function used by this DH Suite. But SHA-256 is mandated for this key establishment scheme by Table 2 of the Draft Recommendation, and so "SHA256" is redundant. Furthermore, the "Group_19" in the DH Suite name is based on IKE Group 19, which

is an Elliptic Curve Diffie-Hellman based on the nineteenth IKE group.

2.2.1. IKE Group 19 Domain Parameters

Stat_ECDH_Group_19 SHALL use the IKE Group 19 domain parameters. For the convenience of reader, the definition of the nineteenth IKE group is reproduced below [FS].

"The curve is based on the integers modulo the generalized Mersenne prime p given by

p = 2^(256)-2^(224)+2^(192)+2^(96)-1.

The equation for the elliptic curve is: y^2 = x^3 - 3 x + b.

Field size: 256

Group Prime/Irreducible Polynomial:
FFFFFFFF 00000001 00000000 00000000
00000000 FFFFFFFF FFFFFFFF FFFFFFFF

Group Curve b:
5AC635D8 AA3A93E7 B3EBBD55 769886BC
651D06B0 CC53B0F6 3BCE3C3E 27D2604B

Group order:
FFFFFFFF 00000000 FFFFFFFF FFFFFFFF
BCE6FAAD A7179E84 F3B9CAC2 FC632551

The group was chosen verifiably at random using SHA-1 as specified in IEEE P1363 [IEEE-1363] from the seed:
C49D3608 86E70493 6A6678E1 139D26B7 819F7E90

The generator for this group is given by g=(gx,gy) where
gx: 6B17D1F2 E12C4247 F8BCE6E5 63A440F2
    77037D81 2DEB33A0 F4A13945 D898C296

gy: 4FE342E2 FE1A7F9B 8EE7EB4A 7C0F9E16
    2BCE3357 6B315ECE CBB64068 37BF51F5"

2.2.2. DHkey Parameter Encoding

The public key for Stat_ECDH_Group_19 is a point on an elliptic curve, which is encoded in the SDP DH "dhkey" parameter. An example of a Stat_ECDH_Group_19 DH suite is shown in Figure 1.

The data in the dhkey parameter of Figure 1 is a point with an x coordinate (gx) followed by a y coordinate (gy) and encoded in base 64. This is given in Augmented Backus-Naur form in the grammar section of this document. The dhkey value is shown in Figure 1 as a pair of base-64 encoded values that follow "dhkey:". These sum to 88 bytes in length. Also shown in the figure is a new parameter for crypto called "nonce", which uses the DH secret in deriving an SRTP master key. DHkey is defined in a later section.

2.2.3. Computation of the Shared Secret

Computation of the Diffie-Hellman shared secret, Z, for this DH suite is defined in Section 5.7.1.2 of the Draft Recommendation [NIST-800-56].

2.3. The Ephem_ECDH_Group_19 DLC Suite

The Ephem_ECDH_Group_19 key establishment scheme is based upon the Cofactor Ephemeral Unified Model of Table 5 in the Draft Recommendation [NIST-800-56]. This scheme uses two ephemeral keys and no static keys.

2.3.1. IKE Group 19 Domain Parameters

Ephem_ECDH_Group_19 SHALL use the IKE Group 19 domain parameters. For the convenience of reader, the definition of the nineteenth IKE group is reproduced in the Stat_ECDH_Group_19 section above.

2.3.2. DHkey Parameter Encoding

The public key for Ephem_ECDH_Group_19 is a point on an elliptic curve, which is encoded in an SDP DH parameter called "dhkey". The dhkey parameter encoding for Ephem_ECDH_Group_19 is identical to that of the Stat_ECDH_Group_19 given above. The Ephem_ECDH_Group_19 grammar is given in the Grammar section below.

2.3.3. Computation of the Shared Secret

Computation of the Diffie-Hellman shared secret, Z, for this DH suite is defined in Section 5.7.1.2 of the Draft Recommendation [NIST-800-56].

2.4. The Stat_FFDH_Group_14 DLC Suite

This section defines a Finite Field Cryptography (FFC) exchange, "Stat_FFDH_Group_14." This key management scheme uses (only) pair-wise static keys and no ephemeral keys. Stat_FFDH_Group_14 is based upon the "dhStatic" scheme of Table 5 in the Draft Recommendation

[NIST-800-56] as well as X.9 and IEEE P1363 [IEEE1363]. This memo defines Stat_FFDH_Group_14 to use the IKE Group 14 domain parameters [RFC3526].

2.4.1. IKE Group 14 Domain Parameters

For the convenience of the reader, the IKE Group 14 definition is copied from the standard [RFC3526] below.

"This group is assigned id 14. This prime is:
2^2048 - 2^1984 - 1 + 2^64 * { [2^1918 pi] + 124476 }

Its hexadecimal value is:

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
670C354E 4ABC9804 F1746C08 CA18217C 32905E46 2E36CE3B
E39E772C 180E8603 9B2783A2 EC07A28F B5C55DF0 6F4C52C9
DE2BCBF6 95581718 3995497C EA956AE5 15D22618 98FA0510
15728E5A 8AACAA68 FFFFFFFF FFFFFFFF

The generator is: 2."

2.4.2. DHkey Parameter Encoding

The dhkey for a Stat_FFDH_Group_14 public key is a 2048 byte integer that is base-64 encoded as shown in the example below.

```
   dhkey: ////////////JD9qiIWjCNMTGYouA3BzRKQJOCIpnzHQCC76mO
xObIlFKCHmONATd75UZs806QxswKwpt8l8UN0/hNW1tUcJF5IW1dmJefsb0T
ELppjftawv/XLb0Brft7jhr+1qJn6WunyQRfEsf5kkoZlHs5Fs9wgB8uKFjv
wWY2kg2HFXTmmkWP6j9JM9fg2VdI9yjrZYcYvNWIIVSu57VKQdwlpZtZww1T
kq8mATxdGwIyhghfDKQXkYuNs474553LBgOhgObJ4Oi7Aeij7XFXfBvTFLJ3
ivL9pVYFxg5lUl86pVq5RXSJhiY+gUQFXKOWoqsqmj////////w==
```

An ABNF grammar for Stat_FFDH_Group_14 is given in the Grammar section of this document.

2.4.3. Computation of the Shared Secret

Computation of the Diffie-Hellman shared secret, Z, for this DH suite is defined in Section 5.7.1.1 of the Draft Recommendation [NIST-800-56].

2.5. The Ephem_FFDH_Group_14 DLC Suite

This section defines a Finite Field Cryptography (FFC) exchange, "Ephem_FFDH_Group_14." This key management scheme uses (only) pair-wise ephemeral keys and no static keys. Ephem_FFDH_Group_14 is based upon the "dhEphem" of the Draft Recommendation [NIST-800-56] as well as X.9 and IEEE P1363 [IEEE1363].

2.5.1. IKE Group 14 Domain Parameters

Ephem_FFDH_Group_14 SHALL use the IKE Group 14 domain parameters [RFC3526]. For the convenience of the reader, the IKE Group 14 definition is given in the Stat_FFDH_Group_14 section above.

2.5.2. DHkey Parameter Encoding

The dhkey for an Ephem_FFDH_Group_14 public key is a 2048 byte integer that is base-64 encoded. The encoding for Ephem_FFDH_Group_14 encoding is identical to that of Stat_FFDH_Group_14 and is shown in the section above. An ABNF grammar for Ephem_FFDH_Group_14 is given in the Grammar section of this document.

2.5.3. Computation of the Shared Secret

Computation of the Diffie-Hellman shared secret, Z, for this DH suite is defined in Section 5.7.1.1 of the Draft Recommendation [NIST-800-56].

2.6. ABNF Grammar for DH Attribute and DLC_Suites

```
"a=DH:" [tag] dlc-suite
tag = *WSP 1*9DIGIT
dlc-suite = 1*WSP ( Stat_FFDH_Group_2 /
                    Stat_ECDH_Group_19  /
                    Stat_FFDH_Group_14 /
                    Ephem_ECDH_Group_19 /
                    Ephem_FFDH_Group_14 )

Stat_FFDH_Group_2   = "Stat_ECDH_Group_2"   1*LWSP "dhkey:" GexpX
Stat_ECDH_Group_19  = "Stat_ECDH_Group_19"  1*LWSP "dhkey:" gx gy
Stat_FFDH_Group_14  = "Stat_FFDH_Group_14"  1*LWSP "dhkey:" GexpW
Ephem_ECDH_Group_19 = "Ephem_ECDH_Group_19" 1*LWSP "dhkey:" gx gy
Ephem_FFDH_Group_14 = "Ephem_FFDH_Group_14" 1*LWSP "dhkey:" GexpW

gx =  *LWSP  44(BASE64)
gy =  1*LWSP 44(BASE64)
GexpX  = 172( *LWSP BASE64 )
GexpW  = 344( *LWSP BASE64 )

BASE64 = %x21-3A / %x3C-7E ; base-64 character set

NOTES:
 1. Gexp* is G^o in the offer where o is the offerer's private key.
 2. Gexp* is G^a in the answer where a is the answerer's private key.
 3. WSP, LWSP, ALPHA, DIGIT and ABNF grammar are from RFC 4234.
```

Figure 7: SDP DH Grammar

2.7. Offer/Answer Processing

An Offerer who uses a DH attribute in an SDP message MUST number that attribute if it is one offer among multiple offers. An Offerer SHOULD NOT number the DH attribute if there is only one offer but an answerer MUST accept a single offer that has a tag and use that tag in a successful reply. Each numbered offer MUST have a unique tag from the other SDP DH offers. Each offer MUST differ in at least one other parameter from the other offers.

Upon receipt of a single offer, the answerer SHALL accept or reject the message according to the answerer's security policy for processing DH. An implementation's security policy MAY allow ECDH or not, for example. Or a policy MAY choose to use only ephemeral keys.

An answerer that does not recognize an SDP DH attribute will of course ignore it [RFC2327], and there will be no DH attribute in the answer. In this case, the offerer MUST abort the offer/answer exchange. When it cannot accept an offer, the answerer MAY return a

DH suite that it will accept as a hint in its answer. The offerer MAY choose to re-run the SDP offer/answer exchange if the alternative DH suite is within its security policy but MUST NOT re-run the exchange if the answerer's DH suite "bids down" its minimum acceptable policy (see the "Security Considerations" below).

Upon receipt of multiple offers, the answerer SHALL accept one offer according to its security policy or it MUST reject all offers. A selected offer MUST have the same tag value in the answerer's DH attribute as it does on the offerer's. The answerer MUST use the same DH suite as the selected offer and MUST provide its public key in the dhkey parameter. When it selects an offer and answers with its public key, the answerer SHALL compute the Diffie-Hellman secret. This secret SHALL be used for nonce processing as defined below.

2.8. Adding New DH Suites

Of the more than one dozen key-management schemes found in the draft document, "Recommendations for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" [NIST-800-56], this memo selects several for SDP. The selection was made on the bases of general usefulness to SDP applications and consideration of known patent encumbrances. Nonetheless, it is likely that new DH suites might be needed and these can be added in the future through a published RFC. The RFC SHOULD reference NIST 800-56 and assign a DH suite name to identify its scheme including whether it is static or ephemeral, Finite Field or Elliptic Curve Cryptography, and give the domain parameter set.

All DH suites in this document use single key-pair key agreement. Future DH suites MAY add two-pair agreement schemes as found in Table 5 of the Draft Recommendation[NIST-800-56].

3. The "nonce" DH Parameter

The "nonce" parameter carries input values that an endpoint uses to derive a shared secret with the other endpoint. In order to use a secret key derived from the DH exchange to protect a particular media line, that media line SHALL contain a "nonce" parameter, which is defined in this section. RFC {sdesc} defines the "inline" parameter to convey a cryptographic key for a media stream. The nonce parameter is used in place of the inline parameter. The two parameters MUST NOT appear in the same media line. The inline parameter carries secret information when it encodes an SRTP master key and salt along with the optional lifetime and master key indicator (MKI) values. The nonce carried by the nonce parameter is passed to the DH Key Derivation Function (Section 3.2), along with the DH public key from the dhkey parameter. The format of the dhkey parameter is shown below.

nonce: salt || nonce ["|" lifetime] ["|" MKI]

The nonce parameter is structurally identical to inline in its concatenated values, though it is semantically different. (Thus, the routines that generate an inline parameter for an offer or an answer can serve with little modification to produce a nonce offer or answer.) Like the inline parameter, the nonce parameter includes two random values for an SRTP key, a 14-byte "salt" and a nonce that is 16 bytes or longer depending on the SRTP crypto suite. However, the nonce from the nonce parameter is not used directly, but instead is passed to the key derivation function; the output of that function provides the key that is associated with the salt. The inline parameter also encodes the optional lifetime and MKI; the definition and use of these fields is as defined by RFC {sdesc}. The ABNF for nonce is given next and followed by the key derivation procedure and offer/answer processing. Note that the nonce parameter MAY be public; there is no requirement to keep it confidential.

3.1. Changes to RFC {sdesc} "crypto" Grammar

The following change is made to the RFC {sdesc} crypto attribute to add the new parameter, "nonce". In the extended crypto attribute grammar shown below, nonce is an alternative (a new key-method-ext) to the inline parameter [sdesc].

```
"a=crypto:" tag 1*WSP crypto-suite 1*WSP key-params
                                      *(1*WSP session-param)

tag              = 1*9DIGIT
crypto-suite     = 1*(ALPHA / DIGIT / "_")

key-params       = key-param *(";" key-param)
key-param        = key-method ":" key-info
key-method       = "inline" / "nonce" / key-method-ext
key-method-ext   = 1*(ALPHA / DIGIT / "_")
key-info         = %x21-3A / %x3C-7E ; visible (printing) characters
                                       ; except semi-colon
```

Figure 3.1-1: Extension to RFC {sdesc} key-method

Figure 3.1-1 is the general syntax for an RFC {sdesc} a=crypto line that is extended with a new key method called "nonce." According to RFC {sdesc}, crypto is specialized according to SDP transport; only the RTP/SAVP (i.e. SRTP) transport has a crypto attribute defined in RFC {sdesc}. This definition is shown below with its nonce extension.

```
key-method          = srtp-key-method
key-info            = srtp-key-info
srtp-key-method     = "inline" / "nonce" ; nonce key is nonce
srtp-key-info       = key-salt ["|" lifetime] ["|" mki]

key-salt            = 1*(base64)   ; binary key and salt values
                                   ; concatenated together, and then
                                   ; base64 encoded [section 6.8 of
                                   ; RFC2046]

lifetime            = ["2^"] 1*(DIGIT)
mki                 = mki-value ":" mki-length
mki-value           = 1*DIGIT
mki-length          = 1*3DIGIT   ; range 1..128.
```

Figure 3.1-2: Extension to RFC {sdesc} srtp-key-method

Figure 3.1-2 is the RTP/SAVP (SRTP) syntax for a "nonce" method. The "key-salt" in nonce is interpreted to be a 16-byte nonce and SRTP salt value. The salt definition is not changed and is a 14-byte binary value. Both nonce and key are base-64 encoded. The nonce is used in a new type of SRTP master key generation as defined below.

3.2. Generating SRTP Master Keys from Public Information

The key that results from the SDP DH key derivation function is used as the SRTP master key. The master salt, key lifetime and key indicator for an SRTP master key are taken from the salt field of the "nonce" parameter. These are shown in Figure 3.1-2. DH key derivation is invoked once for every SRTP master key that needs to be generated, which is once for every media-level SRTP crypto attribute. Thus, an SRTP master key is a special case of DH media-session key derivation, which is described below.

3.3. Media-session Key Derivation

The DH suites of this document use the "Concatenation Key Derivation Function" of Section 5.8.1 of the Draft Recommendation [NIST-800-56]. This key derivation function SHALL be used to derive SRTP master keys prior to SRTP key derivation, which is unchanged by this memo. The function is shown below with substitution of relevant parameters.

SHA-256(counter, Z, OtherInput),

where the values of Z and OtherInput are defined as follows.

counter:	a 32-bit long string containing the value 00000001 (hexadecimal)
Z:	Shared secret computed from the two DH pubkey values
OtherInput:	contextID || keydatalen || pubkeymat
contextID:	set to "offer"||"answer" where || is concatenation
keydatalen:	length of SRTP master key, defined by SRTP Crypto Suite
pubkeymat:	the entire pubkeymat parameter, as an octet string

For larger key sizes, please refer to [NIST-800-56].

The Draft Recommendation defines a contextID and "shared data" that depend on the particular protocol implementation. As defined here, the SDP implementation uses the nonce as shared data. The nonce is carried in the media-level nonce parameter as defined above. According to the Draft Recommendation, the contextID is the concatenation of endpoint identifiers that are also protocol specific. As defined here, the SDP implementation uses "offer" and "answer" for endpoint identifiers, which are fixed length and concatenated. The shared secret Z is the DH secret that is computed according to the Draft Recommendation. The final protocol-dependent parameter is the key derivation function (kdf), which SHALL be SHA-256. The processing of these values is given in the Draft Recommendation but copied below for the convenience of the reader.

"Process:

1. reps = upperbound(keydatalen / hashlen).
2. If reps > (232 ?1), then ABORT: output "Invalid" and stop.
3. Initialize a 32-bit, big-endian bit string counter as 0000000116.
4. For i = 1 to reps by 1, do the following:
   4.1 Compute Hashi = H(counter || Z || contextID || nonce).
   4.2 Increment counter (modulo 232), treating it as an unsigned
       32-bit integer.
5. Let Hhash be set to Hashreps if (keydatalen / hashlen) is an
   integer; otherwise, let Hhash be set to the
   (keydatalen mod hashlen) leftmost bits of Hashreps.
6. Set DerivedKeyingMaterial =
                        Hash1 || Hash2 || ... || Hashreps-1 || Hhash.

Output:

The bit string DerivedKeyingMaterial of length keydatalen bits (or "Invalid"). Any scheme attempting to call this key derivation function with keydatalen greater than or equal to hashlen*(2^32 - 1) shall output "Invalid" and stop without outputting DerivedKeyingMaterial.

Note:

The hashlen is the length in bits of the output block of the hash function, which is a 256-bit SHA-256 message digest."

3.4. Offer/Answer Processing

Use of the nonce parameter follows the Offer/Answer processing rules of RFC {sdesc} with one additional dependency: The answerer MUST reject any media-level crypto attribute with a nonce parameter if there is no session-level Diffie-Hellman secret. If there were no DH attribute in the SDP message and no accepted DH attribute in the answer, then there can be no Diffie-Hellman secret. Thus, successful processing of nonce is dependent upon the successful processing of the SDP DH attribute.

The answerer MUST use a nonce parameter in the answer for each RTP/SAVP media stream that it will source.

4. Person-to-Person Authentication

Diffie-Hellman exchanges MUST be externally authenticated to prevent active attacks in the middle. For phone calls between persons, a strong form of external authentication is for each user to validate a number that is derived from the shared secret. Both users therefore need a common means to arrive at the same output. The default method of this document is to use the DH shared secret as an HMAC key and the contextID concatenated with values of the DH Suite chosen in the answer.

```
FINGERPRINT = HMAC-SHA1 ( Z, "offeranswer" || DH_Suite
                                           || Offer_dhkey
                                           || Answer_dhkey )
DH_Suite = ( "Stat_FFDH_Group_2"   /
             "Ephem_ECDH_Group_19" /
             "Stat_ECDH_Grou_19"   /
             "Stat_FFDH_Group_14"  /
             "Ephem_FFDH_Group_14" )
```

Notes:
1. Z is the shared Diffie-Hellman secret
2. FINGERPRINT truncation and phonetic alphabet are implementation-dependent.

5. Security Considerations

This document improves the security of signaling protocols that use Session Description Protocol (SDP) for SRTP cryptographic context establishment - and potentially other SDP media "transports" as well. As described in the introduction, the public key exchanges establish a shared secret between endpoints using public values in the SDP message even in the absence of integrity protection. There are vulnerabilities to this method, however, that REQUIRE use of external authentication and special consideration for bid-down attacks when the SDP message contains multiple DH and crypto offers and is not integrity-protected. Each are considered below.

5.1. Man-in-the-Middle Attacks

A Diffie-Hellman exchange that is not externally authenticated is vulnerable to a man-in-the-middle attack. In the context of an SDP offer/answer exchange, the threat is that an attacker in control of the signaling channel will substitute its DH public key for that of the offerer and forward it to the answerer. Upon receipt of the answer, the attacker will substitute its DH public key for that of the answerer and forward it on to the offerer. This attack succeeds in obtaining media data sent between the two endpoints when the attacker is also in the middle of the media channel: With its two DH secrets, the attacker has a shared secret with the answerer, another shared secret with the offerer, and can decrypt each packet from one and re-encrypt it for the other. Neither the offerer or answerer can necessarily detect this attack.

There are three defenses to the "man-in-the-middle" attack. One defense is a "person-to-person authentication procedure" where one user reads a hash of the DH secret to the other user who verifies that they have the same truncated value (as done in the AT&T Model 3600 Telephone Security Device [Schneier]). This procedure reveals a man-in-the-middle attack because each truncated hash will be different when two DH secrets are managed by the man-in-the-middle. By assumption, a man-in-the-middle cannot force the hashes of its distinct keys with each endpoint to match, nor can the attacker impersonate the voice of either side well enough to substitute the correct values over the phone. Manual and somewhat laborious, person-to-person authentication needs to be done only once to admit a static key onto the key ring so the procedure need not be repeated for each phone call. This procedure is secure and does not require pre-existing infrastructure between the calling and called parties. This procedure will only work, however, when there are people in communication over the telephone. For machine-to-machine or human-to-machine sessions, there needs to be a pre-existing relationship either from a previous run of person-to-person authentication or with

a common, trusted third party.

The second defense against a man-in-the-middle attack uses a pre-existing relationship between each person and their service provider, who attests to the fact that the caller is authorized to use the "from" address in the SIP signaling message that encapsulates the SDP. This is the "SIP Identity" solution, which establishes that the SIP "from" address is not forged. The service provider will use its private key to integrity-protect the SIP message and to speak on behalf of the domain from which the message originates. In this case, the user MUST trust the service provider to not alter the DH public key that the user has placed in the message. The user therefore trusts the service provider to not launch a man-in-the-middle attack just as the customer of a third-party certificate authority trusts that authority.

If there exists a third party certificate authority that each endpoint trusts to correctly identify the other endpoint, then this method can serve to authenticate each endpoint to the other and prevent a man-in-the-middle attack, but recent experience has shown that no such authority exists for any-to-any authorization applications such as telephony. If such a third party did exist, then the endpoint implicitly trusts the third party to not launch a man-in-the-middle's attack.

Whenever a third party or service provider is trusted to correctly identify the source domain of an endpoint, this external authentication technique can also use person-to-person authentication as a failsafe procedure especially in cases where there is no integrity protection of the SDP message.

5.2. Bid-down Attack

There is a case where integrity protection of the SDP message could thwart an attack: When multiple offers are contained in the SDP message, an attacker in control of the signaling channel might alter the message to always use the weakest DH-suite offer. The use of multiple offers serves to match security policies between the endpoints and is a convenience for periods when a transition from one DH suite to another. An example is the current transition from SHA-1 to SHA-256 hashing for certain security applications. It is a matter of security policy, however, that such a "bid-down attack" not bid down the security below a minimum threshold. In any case, it is RECOMMENDED that answers that are lower than the first (preferred) offer be logged and available to any human user.

5.3. Ephemeral and Static Keys

Certain applications require perfect forward secrecy (PFS), which is a property of session keys that are generated using an ephemeral Diffie-Hellman secret. When the DH public keys and the derived DH secret are ephemeral, they are destroyed (zeroed) immediately after use along with all intermediate results. Thus, the secret used to generate the session keys is destroyed even before any session key is used. For ephemeral DH suites, an attacker gains no advantage by recording all the media streams in hope of stealing the private key of one of the communicating parties. The ephemeral secret and all intermediate computational results MUST be destroyed immediately after they are used [NIST-800-56].

Static DH suites lack the PFS property since they exist after the session terminates. Although a static key that is used for only a single multimedia session might be considered ephemeral, the fact that the keys MUST be destroyed immediately after use is not signaled between the two parties. Although one endpoint might treat its static key as an ephemeral key, there is no guarantee that the other party will do the same.

When the parties desire PFS for a multimedia session, they MUST use an ephemeral DH key. All ephemeral keying materials MUST NOT be cached, written to a file, or maintained following their use in generating session keys. Once used, any and all storage locations of keys and intermediate results MUST be set to zeros. This is the opposite of the recommended procedure for static keys: An implementation SHOULD cache static keys when system resources are available and for a period of time determined by policy.

6. IANA Considerations

IANA is requested to register a new SDP session-level attribute ("att-field") named "DH". "Static_FFDH_Group_2, "Static_ECDH_Group_19", "Ephem_ECDH_Group_19", Statid_FFDH_Group_14", "Ephem_FFDH_Group_14", and "dhkey" are DH parameter names.

IANA is further requested to register a new SDP media level att-field for crypto named "nonce".

7. Acknowledgements

The authors thank Cullen Jennings and Flemming Andreasen for their ideas, suggestions, and review.

8. References

8.1. Normative References

[FIPS-180-2]
"Secure Hash Standard (http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf)", August 2002.

[FS] Fu, D. and J. Solinas, "ECP Groups for IKE, Work in Progress", 2004.

[NIST-800-56]
Barker, E., Johnson, D., and M. Smid, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, NIST Special Publication 800-56", July 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2327] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, May 2003.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[SIPidentity]
Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", 2005.

[sdesc] Andreasen, F., Baugher, M., and D. Wing, "SDP Security Descriptions for Media Streams, IETF Work in Progress, 2006", 2006.

8.2. Informative References

[DVW] Diffie, W., van Oorschot, P., and M. Wiener, ""Authentication and authenticated key exchanges," Designs, Codes, and Cryptography, vol. 2, no. 2, pp. 107--125, 199", 1992, <Diffie, van Oorschot, Wiener>.

[IEEE1363] "IEEE P1363, Standard Specifications for Public Key Cryptography, Institute of Electrical and Electronics Engineers", 2004.

[RFC2412] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.

[RFC3156] Elkins, M., Del Torto, D., Levien, R., and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[RS] Raymond, J-F. and A. Stiglic, "Security Issues in the Diffie-Hellman Key Agreement Protocol (http://crypto.cs.mcgill.ca/~stiglic/Papers/dhfull.pdf)", 2005.

[SEC1] "SEC1: Elliptic Curve Cryptography, Standards for Efficient Cryptogrphy, Certicom Corp., 1999", April 2003.

[Schneier] Schneier, B., "Applied Cryptography", 1996.

[WE] Ellison, C. and J. Walker, "UPnP(TM) Security Ceremonies", October 2003.

[Zimmermann] "Zimmermann, Philip, The Official PGP User's Guide, The MIT Press, 1995 ISBN 0-262-74017-6 (Out of Print)", 2005.

Authors' Addresses

Mark Baugher
Cisco Systems, Inc.
800 East Tasman Drive
San Jose, CA 95164
US

Phone: (503) 245-4543
Email: mbaugher@cisco.com

David A. McGrew
Cisco Systems, Inc.
800 East Tasman Drive
San Jose, CA 95164
US

Phone: (301) 349-5815
Email: mcgrew@cisco.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.