Abstract

This document defines the requirements and a proposed solution for an SDP Offer/Answer exchange model for negotiating best-effort SRTP keys, i.e., in a backward-compatible manner with non-SRTP devices. The proposed solution is a trivial interpretation of the usage of the profile and the usage of SDP indication of [sdesc] and [kmgmt].
1. Introduction

The support of SRTP has been increasing recently, but its adoption has still been relatively slow. One of the reasons for this is that the currently defined mechanisms for exchanging SRTP keys are based on an all-or-nothing approach, i.e., "Always secure" or "Always not secure". If the offer indicates SRTP and the answerer cannot support SRTP, or the particular key exchange mechanism, the entire offer, or the actual session invitation, will fail. When the desired policy is "Always secure", the current established mechanism works perfectly well.

However, a need has been identified for a third policy: "Best-Effort Security". Best Effort Security means that you prefer that SRTP be used, but you are willing to use RTP if the other end does not support it. There are different reasons why one may wish to use a Best Effort policy. It could be to allow for interoperability with many devices that may not support SRTP. In other cases, it may be as a migration strategy or introducing new equipment that support SRTP,
cohabitating with devices that do not support SRTP until the older equipment is replaced.

Today, there is no generally accepted (and backward compatible) way to indicate Best Effort SRTP key negotiation. Therefore, an SRTP-capable device must either be prepared to re-attempt to establish a media stream with RTP after failing with SRTP, or simply not offer SRTP by default, and upgrade to SRTP when possible.

With the current mechanism, it may in the best case be possible to start without SRTP initially (i.e., with the AVP [rtp] profile), and then negotiate through another Offer/Answer [RFC3264] with either [kmgmt] or [sdesc] the usage of SRTP and the session keys. This is an extremely cumbersome process, and has the implication that every call will use an additional Offer/Answer exchange, and will also have the consequence that every call will start without SRTP, which is undesirable.

Also with the current mechanism, starting with SRTP (i.e., the SAVP profile) and downgrading to RTP is only achievable by rejecting the whole session. However, rejecting a session in SIP (normally with a 4XX response code), has very negative implications because of the [herfp] issue. To summarize the issue [herfp] means that the rejection sent by the UAS, when used with forking, is very unlikely to reach the UAC. Since that rejection is intended to cause a re-negotiation without SRTP, the net effect is that the call fork fails completely. In the best case scenario, another fork may answer (e.g., a voicemail system), and in the worst case scenario, the other forks also fails, which means that the calls fails entirely. This is clearly unacceptable and a great impediment to the deployment of SRTP. Note that this is an issue for both parallel and sequential forking.

Note: Some may argue that one may reject the Offer setting the port in the answer to zero as per [RFC3264], and then do a second Offer/Answer; however, since the endpoints that do not support the SAVP profile most likely do not behave this way in the first place (they will reject the whole session), this means it would not be backward compatible to use an Offer rejection mechanism. Furthermore, many UAs automatically generate a BYE if they receive an SDP answer with no accepted media lines.

This document proposes a solution to Best Effort SRTP and backward compatibility problem by introducing a third Policy to the existing ones.

The existing supported mechanisms as of today are as follows:
* SAVP (and associated keys) means secure transport only, i.e. "SRTP only"
This draft proposes a third mechanism:

* AVP with associated SRTP attributes means "Best-Effort SRTP"

The mechanism outlined in this document is fairly trivial, and is defined in order to successfully negotiate multiple mechanisms in one offer/answer exchange, even if the answerer only supports clear/non-secure RTP, and it is backward compatible. Examples are given for [kmgmt] and [sdesc]. This mechanism only applies to Offer/Answer-based applications.

The procedures described in this specification represent a technique that has already been used and deployed in the real-world. It has also been briefly mentioned in [sdp-neg], which lists many techniques used today. Of all the techniques described in [sdp-neg], it is by far the simplest one to address the "Best Effort SRTP" problem, and the only one that does not risk "breaking" any implementations.

It has been argued in [sdp-neg] that using "RTP/AVP" violates [srtp]. After reviewing [srtp], the authors could not find any justification to this claim. Rather, the authors claim that "if SAVP is indicated, we can infer that SRTP is to be used, but the reverse is not necessarily true, i.e., if AVP is used, it does not mean that SRTP will not be used". In other words, there is a well-defined encoding for using SRTP which is "SAVP", but that does not preclude an offerer from offering "AVP" and proposing SRTP dynamically. RFC 3407 [sim-cap] essentially already allows such a model, whereby the backward-compatible encoded media profile may be of one type, while the [sim-cap] offered alternate capability may change the profile for those that understand [sim-cap]. This draft essentially employs a similar model, but using the [sdesc] or [kmgmt] attributes as the explicit alternate profile offer. It should also be noted that [zrtp] already uses AVP for SRTP traffic.

A second requirement of this draft allows the offerer to indicate to the answerer to use unique payload types for SRTP packets, in order to make them distinguishable from clear RTP packets. This is mandatory if the offerer needs to render media before receiving the answer, and cannot do so until it has the keys in the answer to do so.

2. Notational Conventions

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.
terminology in this document conforms to RFC 2828, "Internet Security Glossary".

3. Applicability

This draft proposes using a [sdesc]/[kmgmt] style key exchange in a backward-compatible manner with legacy RTP devices, for Offer/Answer exchange-based applications. This mechanism should provide a smoother migration path, broader applicability, and more rapid acceptance than [sdesc] or [kmgmt] mechanism used only in the "SRTP only" mode.

The rules of this specification apply to AVP/SAVP.

OPEN ISSUE: The same technique arguably could be used with AVPF [RFC4585]/SAVPF[avpf]. Should it? We have assumed only AVP/SAVP, but it could easily be expanded to cover AVPF/SAVPF.

4. Requirements

Unlike the requirements addressed in [sdpng] and [sdp-cap-neg], this mechanism is not trying to address all the general issues with SDP capability negotiation. Instead, it is trying to provide a solution to a very short and defined set of requirements:

REQ-1: It MUST be possible to indicate and negotiate RTP vs. SRTP profiles, on a per media stream basis.

REQ-2: It MUST be possible to offer SRTP profile to an RTP-only answerer, and successfully negotiate RTP, without additional offer/answers.

REQ-3: It MUST be possible to offer SRTP without allowing a fallback to RTP, if the answerer does not support SRTP but the offerer only wishes to either use SRTP or fail the negotiation.

REQ-4: The mechanism MUST be backwards compatible for SIP RTP-only devices, without requiring them to change.

REQ-5: The mechanism MUST be media-type agnostic. (i.e., work with any media type of any codec, etc.)

REQ-6: The mechanism MUST work in the presence of SIP forking.

REQ-7: The mechanism SHOULD support RTCP-based feedback, e.g. AVPF.

OPEN ISSUE: Should REQ-7 be a requirement?
We believe the above to be the necessary and sufficient requirements set to achieve broad applicability and deployment of SRTP in the near future. Below, we provide a proposed solution meeting those requirements.

5. Solution Overview

The basic concept is:

If an SRTP-capable device wishes to *only* offer SRTP and will only accept SRTP be used, then it performs exactly the same steps as defined in \[sdesc\] or \[kmgmt\], including indicating the SAVP profile. The answerer would do the same. This is per current practice.

If a device wishes to offer RTP only, then it uses the AVP profile, and does not use \[sdesc\] or \[kmgmt\]. The answerer does the same. This is also per current practice.

If an SRTP-capable device wishes to offer SRTP but will accept an RTP answer if the far-end only supports that for a given media stream, then it indicates SRTP support as an alternative, by inserting the same media-level crypto attributes of \[sdesc\] or key management attributes of \[kmgmt\], or both, into the offer using non-secure transport profiles (e.g., "AVP" instead of "SAVP"). The offerer may also indicate SAVP support for media lines when a session-level key is used, using a new attribute. The offerer may also indicate unique Payload Type mapping values for SRTP/SRTCP packets in a new attribute, if she wishes to distinguish SRTP from RTP before the SDP answer comes back.

A legacy RTP-only device will ignore these unknown attributes and answer as if they did not exist, i.e. using the "AVP" profile without any crypto or key-mgmt attributes. The offerer then knows they don’t support it and will establish the session using regular RTP.

A device supporting this draft and SRTP understands the attributes indicate a willingness to use the SAVP profile instead, and responds accordingly, by including a crypto or key-mgmt attribute in the answer (but still using "AVP" encoding), resulting in SRTP. If the offer included the payload type mappings, the answerer would use them for SRTP packets.

* The main difficulty with offering SRTP-attributes using non-secure transport profiles is that SRTP packets are virtually
Best Effort SRTP               October 2006

indistinguishable from RTP packets - there is no "SRTP flag". That means the offerer must either wait for the answer before knowing how to handle received RTP packets, or a distinguishing factor must be defined. This specification supports both models, by allowing the offerer to indicate its preference, and mandating the answerer support both.

6. SRTP Attribute

This draft introduces a new media-level SRTP attribute ("a=srtp"), for the purpose of indicating SRTP is desired for a media stream when it would otherwise be ambiguous, and for indicating payload type mappings for the RTP payload types in the m= media line. Examples of the new attribute are as follows:

```
a=srtp
```

```
a=srtp: map:0=96,18=97,101=102
```

The first example shows the attribute being used solely to indicate the associated media line is capable of SRTP.

The second example indicates the associated media line is capable of SRTP, and if SRTP is used, it will accept/receive SRTP on payload type 96 for SRTP in place of payload type 0 (PCMU), 97 for 18 (G729), and 102 for 101.

Following the ABNF rules used in [sdp], the new attribute is defined in ABNF as follows:

```
att-field = "srtp"
att-value = space srtp-options
srtp-options = srtp-payload-mapping | ([srtp-payload-mapping] 1*(byte-string))
srtp-payload-mapping = "map:" map-list
map-list = map *("," map)
map = rtp-pt "+=" srtp-pt
rtp-pt = 1*3(DIGIT)
srtp-pt = 1*3(DIGIT)
;typically dynamic payload types
;in the range 96-127
```

Note that, per the ABNF definition of attribute in [sdp], the att-value field is optional. Thus having both "a=srtp" and "a=srtp:..." forms is legitimate. The <srtp-payload-mapping> is optional if more byte-strings are present, in case some future extension of this attribute does not need the mapping but wishes to use the srtp-capability semantic.

The srtp attribute payload mapping list identifies which payload type numbers offered in the m= media line should be replaced with different payload type numbers if SRTP is chosen. As such, the rtp-
Best Effort SRTP               October 2006

pt (RTP payload type) values MUST correspond to fmt payload types in
the media-description line in an SDP offer. The srtp-pt (SRTP
payload type) values MUST be in the dynamic payload type range of
96-101, unless that is insufficient. The srtp-pt numbers MUST NOT
conflict with any of the offered fmt RTP payload type numbers.

7. Offer/Answer Model:

This draft is based on the offer/answer method of [RFC3264] as used
by [sdesc] and [kmgmt], except the use of secure transport (e.g.,
"SAVP") type encoding in SDP is not always used, as described above
in section 6 and detailed below. This also changes some of the
offer/answer details and RTP processing behavior as described below.

7.1. Generating the Initial Offer

If a device supporting this draft wanted to mandate use of secure
transport (i.e., SRTP) for a particular media stream they MUST
continue to use the [sdesc] or [kmgmt] prescribed secure transport
encoding, i.e. "SAVP", as before.

A device supporting this draft that wishes to use secure transport
if the answerer supports it, but is willing to accept non-secure
transport otherwise (i.e., best effort SRTP), offers the same media-
level crypto attributes and parameters as [sdesc], and/or media-
level key-mgmt attributes of [kmgmt], except that it will indicate
the "RTP/AVP" profile in SDP. The purpose of this is that, should
the receiver(s) of the offer not support SRTP, or not support it for
that particular media stream, the offer will not be rejected. The
offerer can still decide to end the session at any time should it
wish to.

An offerer MAY offer both [sdesc] crypto attributes and [kmgmt] key-
mgmt attributes in the same SDP offer, for the same media sessions.
The offerer SHOULD order them in the order it prefers - the first
type offered is the most preferred, per media stream.

A potential complication is that KMGMT allows for supporting either
session level key management, media level key management, or both.
When used with best-effort negotiation and in conjunction with
[sdesc], an explicit indication is needed to indicate which media
streams are potential SRTP candidates, when session-level [kmgmt] is
used. This specification requires that if Best Effort SRTP is used,
and session-level keys are offered, then a media-level srtp
attribute MUST be included in the offer for each media stream the
offerer wishes to offer SRTP for. If both session-level and media-
level attributes are offered, for example a session-level key-mgmt
and media-level crypto, the srtp attribute encoding MUST be used for
every media stream the offerer wishes to use SRTP for. Note that the media-level key will always be assumed to be more preferred than the session-level one. An offerer MAY include the srtp attribute for any media stream it wishes to offer SRTP for, even if it is not offering a session-level key type.

7.1.1 Offering Unique SRTP Payload Types

In general, the offerer cannot definitively know whether the received media is SRTP or RTP until it receives the answer, because there is no distinguishing factor in the packets. If the offerer wishes to render media before an SDP answer is received, it MUST also include the srtp attribute with the payload mapping list, defined earlier, to indicate unique SRTP payload types for the same offered payloads as the media line. This will indicate to the answerer which payload types to use for SRTP packets, if it accepts the offer using SRTP. The offerer can then safely render RTP media before the answer arrives, for payload types in the media line, while ignoring received packets using the SRTP payload types mapped in the srtp-attribute types until the answer arrives with the key to decrypt SRTP. If the key type used is such that the offerer can decrypt SRTP before the answer, the offerer MAY render it at any time. If the offerer does not wish to render media until an answer is received, it is OPTIONAL whether to include the payload mapping list.

OPEN ISSUE: do we even make it optional, or mandate that the offerer always include the mapping list?

In summary, the offerer encodes a secure transport type (SAVP) for every media stream it demands be secured, while encoding a non-secure transport type (AVP) but with the crypto and/or key-mgmt attributes for every media stream it would accept non-secure transport for. If it offers session-level keys, it includes the srtp attribute for each media line it would have previously used SAVP for. The offerer also indicates unique payload types for SRTP media in the srtp attribute, if it needs to distinguish RTP from SRTP before the answer arrives.

7.2 Generating the Initial Answer

If the offer contained both crypto and key-mgmt attributes, it is up to answerer’s local policy which key mechanism the answerer wishes to use. The answerer SHOULD accept the first one in the offer it understands and can support, per media stream. It MUST only encode the like attribute type it chose to use for SRTP per media stream, in the answer. In other words, it cannot choose both crypto and key-mgmt for the same media stream.
Regardless of local-policy preference for which particular key type to accept, the answerer MUST accept either one or the other if it can. In other words, the offerer has indicated it wishes to use SRTP, and the answerer MUST agree to do so if possible.

As per [sdesc] and [kmgmt], if the answerer chooses to accept crypto attributes or key-mgmt, it MUST use the first attribute in the offered list of attributes per media stream it can support if there are more than one offered attributes for a given key exchange type. For example if two crypto-style keys are offered for a given media stream, the answerer must select the first one it supports.

If it cannot support any of the offered crypto or key-mgmt attributes, however, it MUST treat the offer as if *no* crypto-attributes had been offered. In other words, if the answerer’s policy allows non-secure RTP, it can accept the offer as if it had been so. If the answerer’s local policy is to only allow SRTP media and not accept non-secure RTP, it MAY reject the offer. This lets a key-mgmt-only offerer successfully negotiate non-secure RTP with a crypto-only answerer, and vice-versa.

If the offer used a secure transport encoding in SDP, per [sdesc] or [kmgmt], then it MUST operate as in [sdesc] or [kmgmt] and answer using a secure transport encoding syntax if it can for the same media stream, or fail the offer. Thus an answerer supporting this draft will interoperate with an offerer supporting only legacy [sdesc] or [kmgmt].

If the offer uses best effort SRTP (using RTP/AVP profile), but offered crypto or key-mgmt attributes which were acceptable and answered, the answerer encodes its chosen key type and values and MUST continue to use the insecure transport encoding, i.e., the RTP/AVP profile. If the offer included the srtp attribute, the answerer MUST include the srtp attribute for each media stream the answerer wishes to use SRTP for.

7.2.1 Answering Unique SRTP Payload Types

If the offer indicated unique payload types for SRTP, by including the srtp attribute and payload mapping list, the answerer MUST use the indicated payload types for SRTP packets it sends, if it accepts the SRTP offer. This is so that the offerer can distinguish RTP from SRTP packets arriving before the answer, which can often be the case. If the answerer used the same payload types for SRTP packets as indicated in the m= media line, the offerer would attempt to render them as RTP, which would produce a degraded user experience.
The answerer MUST answer such an offer using the actual/real payload types it is going to use for RTP or SRTP packets. Therefore, if it selects to use SRTP and the offerer indicated payload type mappings in the srtp attribute, the answerer will answer using the srtp-specific payload type values in the m= media line of its answer.

If the offer included the optional payload mappings in an srtp attribute, the answerer MUST include the srtp attribute for each media stream the answerer wishes to use SRTP for, with the same payload mappings, for those formats it is answering with.

7.3. Processing the Initial Answer

As per [sdesc] and [kmgmt], the answer is checked for matching crypto or key-mgmt attributes and key information. If the answer uses the RTP/AVP profile, and no crypto or key-mgmt attribute lines are found in the answer, however, and the originally offered transport was "AVP", then the negotiation MUST NOT be considered to have failed. Instead, non-secure RTP is used regardless if the original offer included any crypto or key-mgmt attributes to begin with. This lets a Best-Effort SRTP offerer successfully negotiate with a non-SRTP answerer, and a key-mgmt-only offerer successfully negotiate non-secure RTP with a crypto-only answerer, and vice-versa.

If a crypto attribute line is found in the answer, but does not have a matching tag, included key, or contain all of the mandatory negotiated session parameters, then the session negotiation MUST fail. If a key-mgmt line is found in the answer, but does not pass key management protocol processing, then the session negotiation MUST fail. If both a crypto attribute and key-mgmt line is found in the answer, at the media-level for the same media stream, then the session negotiation MUST fail. If an answer contains a valid crypto or key-mgmt attribute but they were not of the same key exchange type as the offer for that media stream, then the session negotiation MUST fail. These would all represent protocol failures.

If a key-mgmt line is found in the answer at the session level and a key-mgmt or crypto attribute at the media level, and such was also offered, then the media-level answers are used for each respective media stream, and the session-level one used for the remaining SAVP media streams (ones without media-level crypto or key-mgmt answers). This is the same as best current practice today.

For each media stream which an acceptable answer is received at the media level, and for all remaining SAVP media streams if an acceptable session-level answer, the offerer MUST only accept SRTP using the key and other values in the answer. It would do so as described in [sdesc] or [kmgmt], as if the original Offer and Answer
used SAVP secure transport encoding. The offerer would then begin generating SRTP based on the answer as per [sdesc] or [kmgmt] and [srtp].

8. Forked Offers and Multiple Answers

The generated Offer may be forked along the path, resulting in multiple Answers. It is typically up to local-policy how to handle such situations.

9. Clipping and Changing Transport Types

This draft does not rely on an Answer before processing RTP media, but may rely on such for SRTP media. Such is the case typically for SRTP today regardless, as the offered keys are usually the transmit keys - so an Answer has to be received to know how to decrypt received SRTP. NAT traversal using ICE has this limitation as well. [sdesc] and [mikey-rsa-r] also have this limitation. Security preconditions, as defined in [sec-pre], and/or sending the SDP answer in provisional responses as soon as possible, are RECOMMENDED for such cases.

This draft, however, provides a solution for rendering RTP media safely, without worry it is SRTP, before the answer arrives. For cases where the decrypt key is known at the time of the offer, this draft also provides the ability to render SRTP media before the answer arrives.

A second issue is changing transport types, in an updated offer/answer. Since media typically reaches the UAC before an answer, it may be difficult to know when to switch from RTP to SRTP or vice-versa. This draft provides a solution to that problem as well, by simply offering new dynamic payload types.

10. Example Offer/Answer Exchange

In the following example, Alice is proposing to establish an unsecure RTP H.263 video channel in conjunction with a Best Effort SRTP voice channel using either G.711 or G.729, using either [kmgmt] or [sdesc]. Alice wishes to use payload type values 96 and 97 for the RTP payload types of 0 and 18. Note that the a=crypto and the a=key-mgmt lines are really 2 long lines.

```
v=0
o=alice 2890844526 2890842807 IN IP4 192.0.2.2
```
In this sample Answer below, Bob does not support SRTP, and therefore ignores the [kmgmt] and [sdesc] attributes and selects RTP for the voice channel, and also accepts the video channel.

v=0
o=bob 2890890210 807082634 IN IP4 192.0.2.4
s=Open discussion
e=bob@example.net (Bob)
c=IN IP4 192.0.2.4
t=2873397496 2873404696
m=video 4900 RTP/AVP 34
a=rtpmap:34 H263/9000
m=audio 32640 RTP/AVP 0
a=rtpmap:0 PCMU/8000

In this sample Answer below, Bob does support SRTP, selects [sdesc] for the voice channel, and also accepts the video channel. Note Bob includes the srtp attribute and payload mapping info, and will accept SRTP packets using payload type 102 for RTP of 0 (PCMU).

v=0
o=bob 2890890210 807082634 IN IP4 192.0.2.4
s=Secret discussion
e=bob@example.net (Bob)
c=IN IP4 192.0.2.4
t=2873397496 2873404696
m=video 4900 RTP/AVP 34
a=rtpmap:34 H263/9000
m=audio 32640 RTP/AVP 102
a=rtpmap:102 PCMU/8000
a=srtp: map:0=102
11. Security Considerations

Like [sdesc], SDP using the mechanism in this draft with crypto attributes is conveyed in an encapsulating application protocol which MUST provide both strong eavesdropping and authentication mechanisms. The same may be true of key-mgmt lines, depending on the key management protocol. The security requirements in [sdesc] and [kmgmt] MUST be followed for this draft as well.

11.1. Security Implications vs. [sdesc] and [kmgmt]

The best-effort mechanism proposed in this draft may be considered less secure than [sdesc] and [kmgmt] because it allows a bid-down attack to establish non-secure RTP sessions, even if both ends supported SRTP. It is however more secure than not using SRTP at all. This is by design, however, in order to facilitate interoperability and migration from RTP to SRTP. No mechanism proposed so far truly can prevent a bid-down attack. The difference is that [sdesc] and [kmgmt] would result in a failed session negotiation, whereas this mechanism would not. The authors consider that a benefit of this draft, not a drawback. This draft still mandates using SAVP if the offerer ‘only’ accepts secure transport. If the offerer would accept less anyway, then a malicious attacker can as easily bid-down [sdesc] or [kmgmt] simply by failing the session, since by definition such an offerer will re-signal using non-secure transport encoding. Therefore, this draft’s mechanism is only more susceptible to bid-down in a trivial way – namely because it will happen in fewer messages.

Using S/MIME or signing bodies using [identity] may also prevent bid-down attacks.

Neither party needs to accept a session using non-secure RTP: the offerer can simply use the secure transport encoding in SDP, and the answerer can simply reject offers which do not offer such; or either end can terminate the session or re-offer at any time. Those are local policy decisions which are available for any mechanism.
12. References

12.1. Normative References


12.2. Informative References


[identity] Peterson, Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06

[savpf] Ott, Carrara, "Extended Secure RTP Profile for RTCP-based Feedback (RTP/SAVPF)", draft-ietf-avt-profile-savpf-06


[zrtp] Zimmermann, "ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP", draft-zimmermann-avt-zrtp-01

Author’s Address

Hadriel Kaplan
Acme Packet
71 Third Ave.
Burlington, MA 01803, USA
Email: hkaplan@acmepacket.com

Francois Audet
Nortel Networks
4655 Great America Parkway
Santa Clara, CA 95054, USA
Email: audet@nortel.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.
The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgments

The authors wish to thank Alan Johnston who first suggested the idea (we believe). We also thank Flemming Andreasen, Dan Wing, Randell Jessup, Andrew Zmolek, Robert Gilman and John Elwell for their suggestions and comments.