PERC
Internet-Draft
Intended status: Informational
Expires: April 18, 2016

C. Groves, Ed.
W. Yang
R. Even
Huawei

October 16, 2015

Usage of CLUE with PERC
draft-groves-perc-clue-00

Abstract

This document provides an initial discussion of the relationship between PERC and CLUE. It seeks to identify any potential impacts or/and enhancement to the way that CLUE is used in the PERC architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 18, 2016.

Copyright Notice

Table of Contents

1. Introduction

OThe PERC working charter specifically mentions that the solution for PERC should:

> "be implementable by both SIP (RFC3261) and WebRTC endpoints [I-D.ietf-rtcweb-overview]. How telepresence endpoints using the protocols defined in the CLUE working group could utilize the defined security solution needs to be considered. However, it is acknowledged that limitations may exist, resulting in restricted functionality or need for additional adaptations of the CLUE protocols."

It also indicates that work for documenting the model for integrating PERC with based with the establishment of CLUE conferences needs to be performed.

This draft provides some initial information to address both these areas.

2. Requirements 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 [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

3. CLUE Background

The CLUE protocol framework [I-D.ietf-clue-framework] effectively is a means of sending metadata about media captures and encodings between a Providing Endpoint and a Consuming Endpoint. The CLUE protocol is transmitted using a WebRTC Datachannel [I-D.ietf-clue-datachannel] meaning that any SRTP based mechanisms for encrypting this metadata cannot be used.

The information that can be sent regarding media captures is summarized below:

- Spatial information, including point of capture, point on line of capture, area of capture, mobility of capture and audio capture sensitivity pattern;

- Descriptive information, including a human readable description, indication of presentation, field of view type and language;

- Person information, including the role of the person and xCard description;

- Miscellaneous information, including whether text is embedded or a relation to other captures.

It is possible for providers through the Multi-Content Capture (MCC) mechanism to provide the information about the individual contributing sources. It can provide the switching policy as well as synchronization information.

Information about the overall "Scene" may also be provided including views, human readable descriptions, xCard and scale information.

Using the CLUE protocol information the Consuming endpoint can then choose what media captures and encodings that it would like to receive through the use of CLUE and SIP/SDP signalling. Media is typically provided through SRTP. Figure 2 / [I-D.ietf-clue-framework] highlights the basic call flow. Figure 1 below provides a copy of this flow.

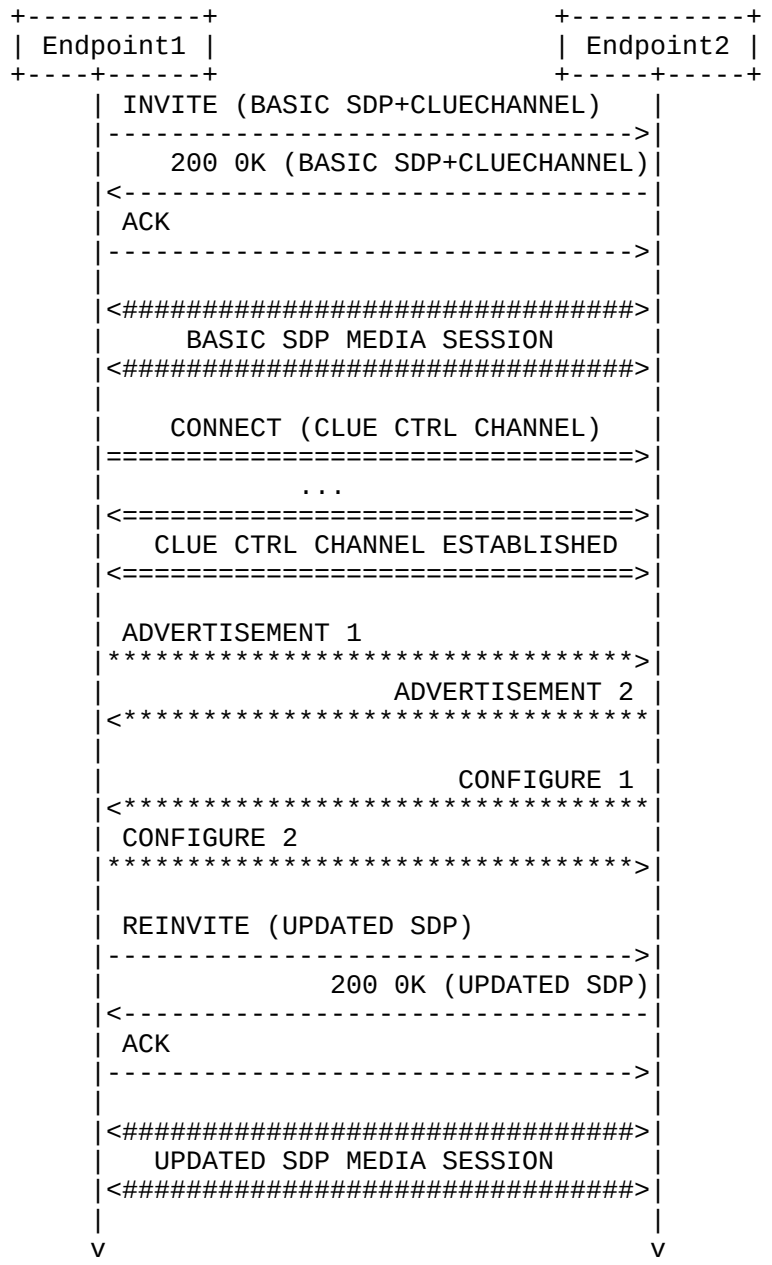

Figure 1: CLUE Basic Information Flow

Whilst CLUE is a point to point protocol it may be used in conferences containing multipoint control units (MCU)s. In this

scenario the MCU acts as an aggregation point for CLUE information. That is the MCU receives ADVERTISEMENT messages received from multiple endpoints before deciding on the contents of the ADVERTISEMENT that it wishes to send. Likewise the MCU will use received CONFIGURE messages to decide what the contents of its CONFIGURE messages will be. In doing so the MCU may apply any local policy / provisioning information to its decisions. Figure 2 illustrates this CLUE signalling. The SIP/SDP signalling is omitted for brevity.

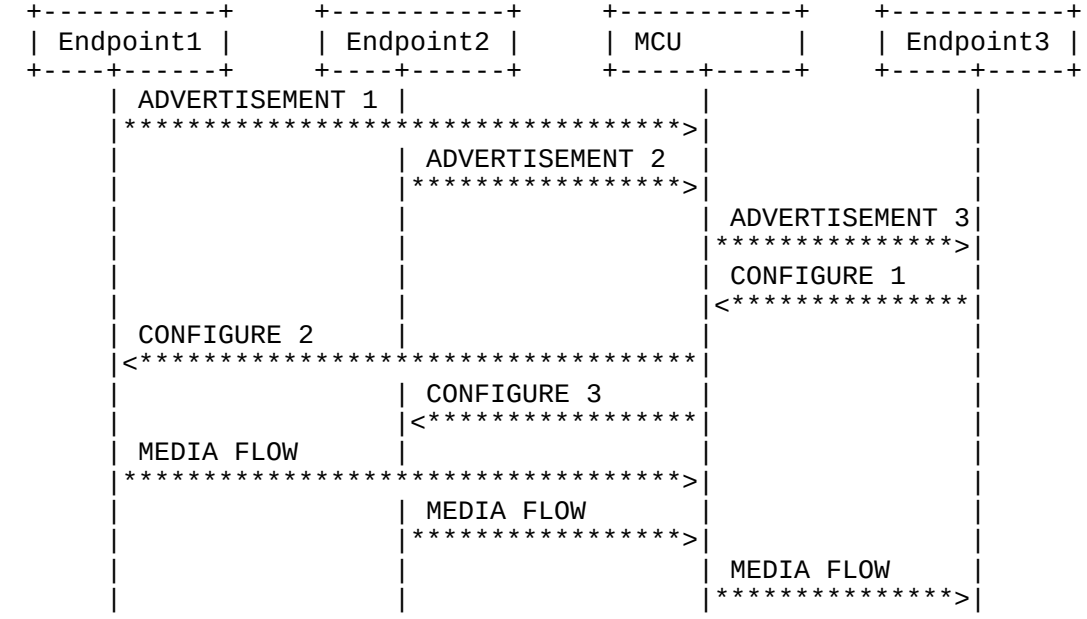

Figure 2: CLUE MCU Flow

Figure 2 shows a unidirectional media flow to Endpoint3. A bi-directional media flow would be enabled by Endpoint3 providing an ADVERTISEMENT to the MCU and the MCU providing ADVERTISEMENTS to Endpoint1 and Endpoint2. Endpoint1 and Endpoint2 would then also send CONFIGURE messages to the MCU and the MCU would send a CONFIGURE message to Endpoint3.

Thus the selection of a particular Media Capture and Encoding (Capture Encoding) by the endpoints drives what topology occurs at the MCU. However there is a caveat. The MCU could apply filtering of the CLUE metadata to provide a sub-set of the data or append its own data. For example it may decide that rather than offer the

source audio from Endpoint1 and Endpoint2 as individual streams, it will offer a mix of these two sources, as individual streams.

4. CLUE Relation to PERC

As detailed in the charter, the goal of the PERC WG is to:

> "...work on a solution that enables centralized SRTP-based conferencing, where the central device distributing the media is not required to be trusted with the keys to decrypt the participants' media. The media must be kept confidential and authenticated between an originating endpoint and the explicitly allowed receiving endpoints or other devices. The meta information provided to the central device is to be limited to the minimal required for it to perform its function to preserve the conference participant's privacy."

As described above CLUE largely provides metadata (or meta information) so the task is to identify the minimal set of CLUE data required for CLUE to still work. It also needs to be considered the limited functionality of a media distribution device (MDD) as compared to an MCU.

In broad terms the initial PERC drafts propose a solution where there are two sets of encryption keys, one for the end-to-end (e2e) session and another for the transport connection (i.e. between the endpoint and MDD). SRTP extensions are required to carry the e2e encrypted data. The concept of a key management function (KMF) is also introduced which receives information about the call and the endpoints (as per 8.1/[I-D.jones-perc-private-media-framework]). The KMF is the element that the endpoints trust it provides cryptographic keys and authenticates media content. See 6.1 / [I-D.jones-perc-private-media-reqts].

So far only SRTP media has been considered by PERC. As the MDD does not have access to the un-encrypted media stream it can only provide switching topologies (e.g. Media Switch, Selective Forwarding Unit, Transport Translator/ Transport Relay?, [I-D.ietf-avtcore-rtp-topologies-update]).

These aspects have some implications on the use of CLUE.

4.1. Topology

As noted in the background section the selection of particular captures and encodings by endpoints effectively dictates what topology occurs at the MCU/MDD. Therefore where a CLUE enabled MDD receives an indication that an encoding requires the use of PERC, the

MDD must ensure that in any subsequent ADVERTISEMENTs and SIP/SDP offers it sends, that the capture encoding is an unmixed local source (i.e. doesn't use a MCC indicating a MDD local composition of remote sources). This would be a mismatch in capabilities as the MDD is unable to mix SRTP streams.

Whilst the MDD may utilise the MCC mechanism to indicate that a particular capture encoding may represent multiple sources, the MaxCaptures attribute (section 7.2.1.1/[I-D.ietf-clue-framework]) should be set to <=1 or 1 to indicate that only switching is used. Other MaxCapture values indicate the potential use of composed (thus mixed) capture encodings.

A MCC Policy attribute (section 7.2.1.2/[I-D.ietf-clue-framework]) may also be included. It allows the indication of a "SoundLevel" policy that indicates that the content of the capture encoding is determined by a sound level detection algorithm. As a PERC enabled MDD cannot access the SRTP media the "SoundLevel" policy should not be used unless the endpoint indicates the use of an unencrypted or hop by hop mechanism (e.g. utilising [RFC6464]) for sound level detection.

4.2. Media manipulation

Given that the PERC enabled MDD cannot access the encrypted media, it cannot filter possible media content. For example an endpoint may indicate that the media capture contains embedded text (clause 7.1.1.13/[I-D.ietf-clue-framework]) information. It has no mechanism to filter out (e.g. by removing part of the image, or text signaled associated with audio) or to confirm text is being sent. Therefore the MDD can either only remove the capture from being ADVERTISED or pass the embedded text attribute without modification.

A CLUE enabled MDD has the ability of adding its own captures and encodings to ADVERTISEMENTS. PERC enabled consumers should determine if the encoding associated with the advertised captures contains the correct key/fingerprint information as distributed by the KMF before requesting the capture encoding via a CONFIGURE. This is a similar consideration as for non-CLUE endpoint responding with an SDP Answer.

4.3. Privacy

The CLUE framework allows the sending of potentially private information to the MCU. Participant and endpoint information via the xCard [RFC6351] format may be provided. xCard can contain address, contact, company, images and audio information. Whilst this information does not compromise the encrypted media it does provide information about the persons generating it.

CLUE also allows the definition of extensions so there may be proprietory extensions that may also contain potentially senstive information.

As indicated in the PERC WG charter meta information provided to the central device is to be limited to the minimal required for the MDD to perform its function. This may potentially result in an CLUE endpoint significantly reducing the amount of metadata it sends in ADVERTISEMENTS. This would result in decreased information for CONSUMERs to decide which captures to consumer. This may lead to a decreased telepresence user experience.

4.4. Encodings

CLUE itself carries little encoding information other than encoding groups with indicate which encodings are linked (and the maxmimum bit rate) and encodingIDs of the individual encodings. The encodingIDs provide a link to the actual encoding information provided through SIP/SDP. The SDP utilizes the "a=group" and "a=mid" mechanism to reference the CLUE encodingIDs thus providing a linkage between CLUE and SDP.

It is expected that any indication of the use of PERC for SRTP streams will be signaled through SDP. Therefore a CLUE enabled endpoint is not required to change any CLUE based encoding information to use PERC.

4.5. Mapping RTP streams to CLUE media captures

In order to associate RTP media with a particular CLUE capture encoding [I-D.ietf-clue-rtp-mapping] defines a RTP header extension and a RTCP SDES item both containing a CaptureID. The draft indicates for mapping an RTP stream to a specific MC in the MCC the CLUE the media sender MUST send for MCC the captureID in the RTP header and as a RTCP SDES message.

If an MDD produces or modifies MCCs (in particular the individual source CaptureIDs) as per section 4.1 above, then it may need to potentially modify the received source RTP/RTCP captureIDs to match the CLUE MCC before sending RTP/RTCP. In the case of voice activated switching, the MDD should also send the relevant RTP/RTCP captureID.

Therefore any PERC solution should ensure that the MDD may have access to and the ability to send RTP/RTCP captureID.

4.6. Others?

TBD

5. Potential CLUE enhancements

CLUE has a defined extension mechanism (see section 8/[ID.ietf-clue-protocol]). The use of any enhancements related to PERC could be negotiated through this mechanism.

5.1. Encrypted CLUE information

In order to limit the amount of metadata available to the MDD but still allowing the full use of CLUE, CLUE could be enhanced to carry encrypted data that is associated with a capture/scene but is not available to an MDD. This would be similar to the proposed solution for SRTP. The KMF could be enhanced to provide keys to the endpoints to access this CLUE encrypted data to make decisions on which capture encodings to CONFIGURE.

In a PERC environment the endpoints are responsible for stream selection and any composition and thus they should have access to the full capture and scene metadata provided by the other endpoints in a conference. A MDD that switches streams doesn't need access to this metadata as it should not make decisions regarding the forwarding of streams based on the content/characteristics of the stream. Accordingly the MDD only strictly needs a CaptureID and the encoding information in order to switch streams. CLUE capture attributes, capture scene, simultaneous set and people information may be encrypted and passed through the MDD. This is due to the fact that a CONFIGURE only contains a CaptureID and an associated EncodingID. It's the CONFIGURE message that determine which capture encoding an endpoint sends.

The syntax below in figure 3 provides a conceptual illustration of the clear and encrypted parts of a CLUE ADVERTISEMENT utilising the example from section 10.1 / [ID.ietf-clue-protocol]:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<advertisement xmlns="urn:ietf:params:xml:ns:clue-protocol"
               xmlns:ns2="urn:ietf:params:xml:ns:clue-info"
               xmlns:ns3="urn:ietf:params:xml:ns:vcard-4.0"
               protocol="CLUE" v="0.4">
 <clueId>Napoli</clueId>
 <sequenceNr>45</sequenceNr>
 <mediaCaptures>
 <ns2:mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:type="ns2:videoCaptureType"  captureID="AC0" mediaType="video">
```

```
            <ns2:captureSceneIDREF>CS1</ns2:captureSceneIDREF>
            <ns2:encGroupIDREF>EG1</ns2:encGroupIDREF>
           /**************** Encrypted contents *************/
 <ns2:mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="ns2:videoCaptureType" mediaType="video" captureID="VC0">
            <ns2:captureSceneIDREF>CS1</ns2:captureSceneIDREF>
            <ns2:encGroupIDREF>EG0</ns2:encGroupIDREF>
            /**************** Encrypted contents *************/
        </ns2:mediaCapture>
 <ns2:mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="ns2:videoCaptureType" mediaType="video" captureID="VC1">
            <ns2:captureSceneIDREF>CS1</ns2:captureSceneIDREF>
            <ns2:encGroupIDREF>EG0</ns2:encGroupIDREF>
            /**************** Encrypted contents *************/
        </ns2:mediaCapture>
 <ns2:mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="ns2:videoCaptureType" mediaType="video" captureID="VC3">
            <ns2:captureSceneIDREF>CS1</ns2:captureSceneIDREF>
            <ns2:encGroupIDREF>EG0</ns2:encGroupIDREF>
            /**************** Encrypted contents *************/
        </ns2:mediaCapture>
 <ns2:mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="ns2:videoCaptureType" mediaType="video" captureID="VC4">
            <ns2:captureSceneIDREF>CS1</ns2:captureSceneIDREF>
            <ns2:encGroupIDREF>EG0</ns2:encGroupIDREF>
            /**************** Encrypted contents *************/
    </mediaCaptures>
    <encodingGroups>
        <ns2:encodingGroup encodingGroupID="EG0">
            <ns2:maxGroupBandwidth>600000</ns2:maxGroupBandwidth>
            <ns2:encodingIDList>
                <ns2:encID>ENC1</ns2:encID>
                <ns2:encID>ENC2</ns2:encID>
                <ns2:encID>ENC3</ns2:encID>
            </ns2:encodingIDList>
        </ns2:encodingGroup>
        <ns2:encodingGroup encodingGroupID="EG1">
            <ns2:maxGroupBandwidth>300000</ns2:maxGroupBandwidth>
            <ns2:encodingIDList>
                <ns2:encID>ENC4</ns2:encID>
                <ns2:encID>ENC5</ns2:encID>
            </ns2:encodingIDList>
        </ns2:encodingGroup>
    </encodingGroups>
    <captureScenes>
            /**************** Encrypted contents *************/
    </captureScenes>
    <simultaneousSets>
```

```
            /**************** Encrypted contents **************/
    </simultaneousSets>
    <people>
            /**************** Encrypted contents **************/
    </people>
</advertisement>
```

Figure 3: Encrypted CLUE Advertisement

The downside of this approach is that the MDD effectively becomes unable to offer its own switched streams as multiple content captures. Whilst in theory it could offer its own MCCs utilising the unencrypted CaptureIDs, it has little metadata to decide which streams are related in order to provide synchronised switching. Therefore it could be recommended that information such as the capture area (which is unlikely to be sensitive) should be passed in the clear (unencrypted) to allow the MDD to distinguish that the captures cover different parts of the same scene. In this case the MDD could provide a MCC.

5.2. Others?

TBD

6. Summary

This draft provides a discussion of the relationship between CLUE and PERC and the potential impacts to CLUE when used with PERC streams.

7. Acknowledgements

This template was derived from an initial version written by Pekka Savola and contributed by him to the xml2rfc project.

8. IANA Considerations

None.

9. Security Considerations

This draft is about the privacy and security implications of using CLUE in a PERC environment.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

10.2. Informative References

[I-D.ietf-avtcore-rtp-topologies-update] Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-10 (work in progress), July 2015.

[I-D.ietf-clue-datachannel] Holmberg, C., "CLUE Protocol data channel", draft-ietf-clue-datachannel-10 (work in progress), September 2015.

[I-D.ietf-clue-framework] Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-23 (work in progress), September 2015.

[I-D.ietf-clue-rtp-mapping] Even, R. and J. Lennox, "Mapping RTP streams to CLUE media captures", draft-ietf-clue-rtp-mapping-04 (work in progress), March 2015.

[I-D.jones-perc-private-media-framework] Jones, P., Ismail, N., and D. Benham, "A Solution Framework for Private Media in Privacy Enhanced RTP Conferencing", draft-jones-perc-private-media-framework-01 (work in progress), October 2015.

[I-D.jones-perc-private-media-reqts] Jones, P., Ismail, N., Benham, D., Buckles, N., Mattsson, J., and R. Barnes, "Private Media Requirements in Privacy Enhanced RTP Conferencing", draft-jones-perc-private-media-reqts-00 (work in progress), July 2015.

[RFC6351] Perreault, S., "xCard: vCard XML Representation", RFC 6351, DOI 10.17487/RFC6351, August 2011, <http://www.rfc-editor.org/info/rfc6351>.

[RFC6464] Lennox, J., Ed., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <http://www.rfc-editor.org/info/rfc6464>.

Authors' Addresses

Christian Groves (editor)
Huawei
Melbourne
Australia

Email: Christian.Groves@nteczone.com

Weiwei Yang
Huawei
P.R.China

Email: tommy@huawei.com

Roni Even
Huawei
Tel Aviv
Isreal

Email: roni.even@mail01.huawei.com