

SDES

Eric Rescorla

`ekr@rtfm.com`

IETF 87

August 1, 2013

Overview

- Comparison of security properties
- DTLS and backward compatibility
- The bigger picture

Security Properties wrt Signaling Server

- In SDES, signaling server has the key
 - Passive access to the encrypted media is sufficient to recover the plaintext
- In DTLS-SRTP, signaling server authenticates endpoints
 - Can mount a MITM attack
- Key continuity or Identity allow detection of attack by signaling server
 - As well as identifying the person on the other end
 - Allows after the fact auditing as well

This is the kind of thing I mean

Activity on this account

This feature provides information about the last activity on this mail account and any concurrent activity. [Learn more](#)

This account is open in 4 other locations.
(Location may refer to a different session on the same computer.)

Concurrent session information:

Access Type [?] (Browser, mobile, etc.)	Location (IP address) [?]
Browser	United States (CA) (172.18.222.92)
Browser	United States (CA) (172.18.112.221)
Browser	United States (CA) (172.18.28.15)
Browser	United States (CA) (172.18.28.14)

Recent activity:

If the activity below doesn't look like yours, [change your password immediately](#) [Learn more](#)

Access Type [?] (Browser, mobile, POP3, etc.)	Location (IP address) [?]	Date/Time (Displayed in your time zone)
Unknown	Poland (83.17.123.186)	Mar 8 (2 days ago)
Browser	* United States (CA) (172.18.113.120)	1:03 pm (0 minutes ago)
Google Toolbar	* United States (CA) (172.18.113.120)	1:03 pm (0 minutes ago)
Browser	United States (CA) (172.18.112.221)	1:03 pm (0 minutes ago)
Browser	United States (CA) (172.18.113.120)	1:02 pm (1 minute ago)
Google Toolbar	United States (CA) (172.18.113.120)	1:02 pm (1 minute ago)

Alert preference: Show an alert for unusual activity. [change](#)

* indicates activity from the current session.

This computer is using IP address 172.18.113.120. (United States (CA))

Active vs. Passive Attack. Does it matter?

- Timescale
 - Passive attack can be mounted retrospectively
 - ... especially if you have the ability to capture media and logs
 - Active attack can only be mounted in real-time
- Visibility
 - Passive attack can be mounted invisibly
 - Active attack cannot be completely hidden from user
 - * ... though detection is not always easy
- Malice vs. incompetence
 - Easy for a site to accidentally mount a passive attack via server logs, etc.
 - Not possible to accidentally mount an active attack

DTLS vs. SDES Performance

- DTLS handshake is a trivial cost compared to audio or video encoding
 - Which you're doing if you're an endpoint
 - See Langley's talk from Velocity 2010
- Clipping is a non-issue
 - DTLS can be done in 1RTT with False Start
 - * ... small compared to ICE overhead
 - Expect new work in TLS-WG on reducing DTLS latency further for subsequent calls

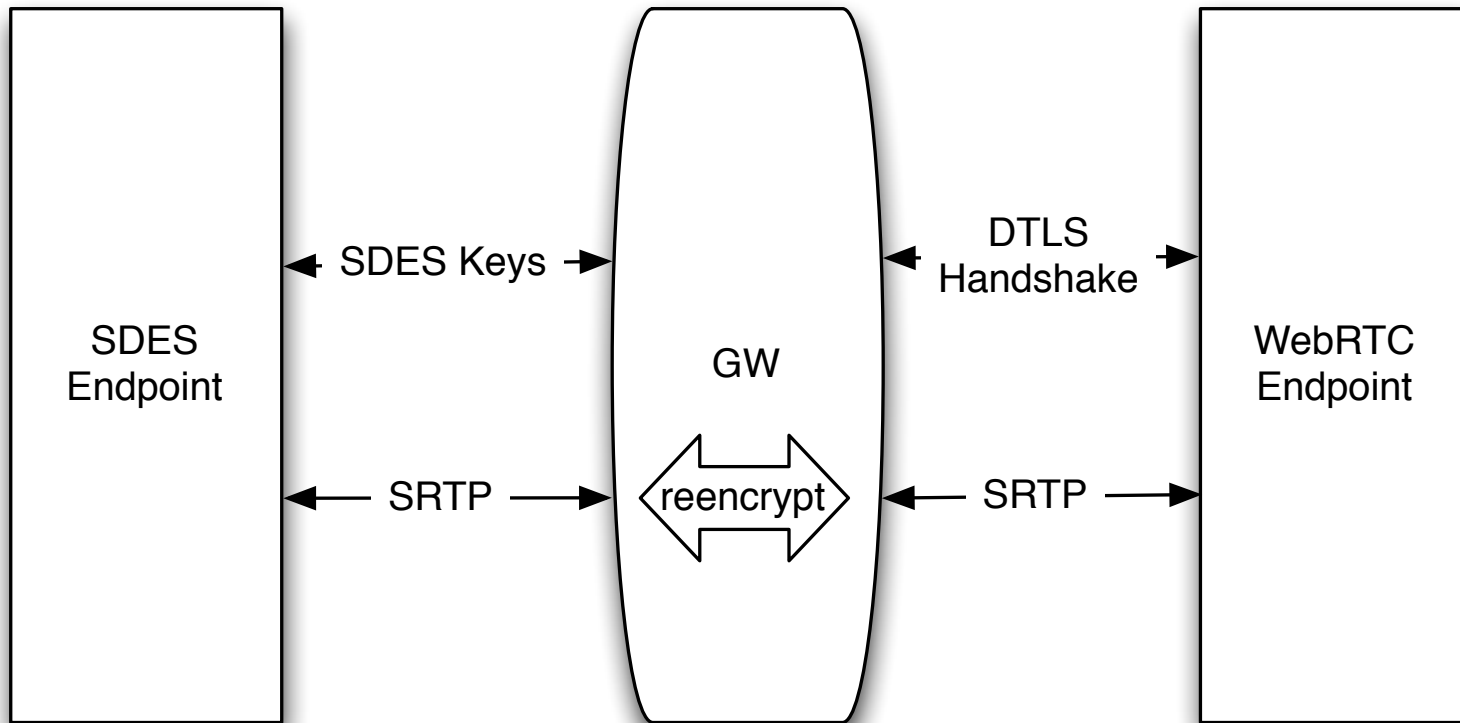
DTLS and Backward Compatibility

- The vast majority of RTP traffic isn't SRTP
- The vast majority of SRTP traffic is secured with SDES
 - The majority of legacy SRTP implementations only support SDES
- DTLS-SRTP and SDES-SRTP interop requires gatewaying

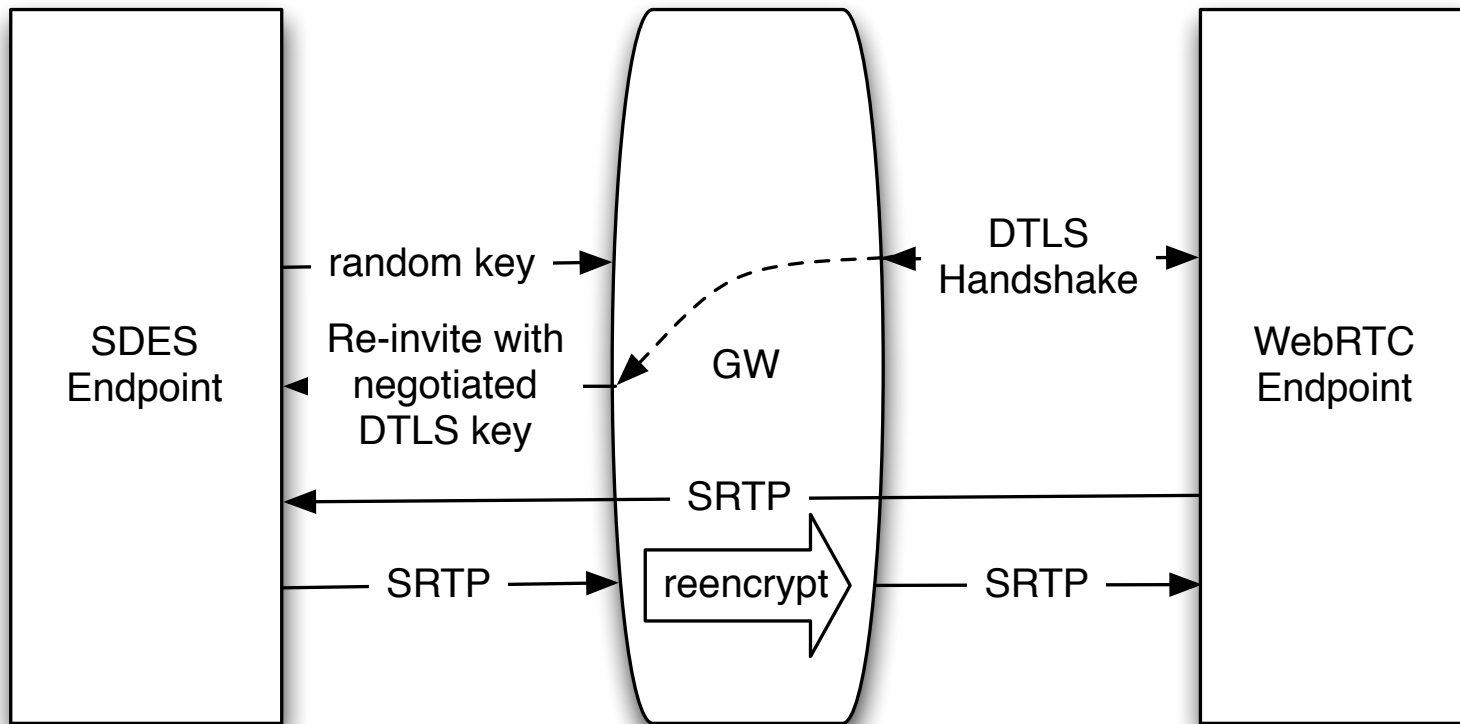
Is reencryption that big a deal?

- Quite likely we'll need media gateways anyway
 - Many implementations won't do ICE
 - May need to transcode audio (Opus) or video (VP8)
- Reencryption isn't that expensive (see above)
- Many MCUs are going to want to decrypt and reencrypt the media anyway
- We still have EKT if we need it

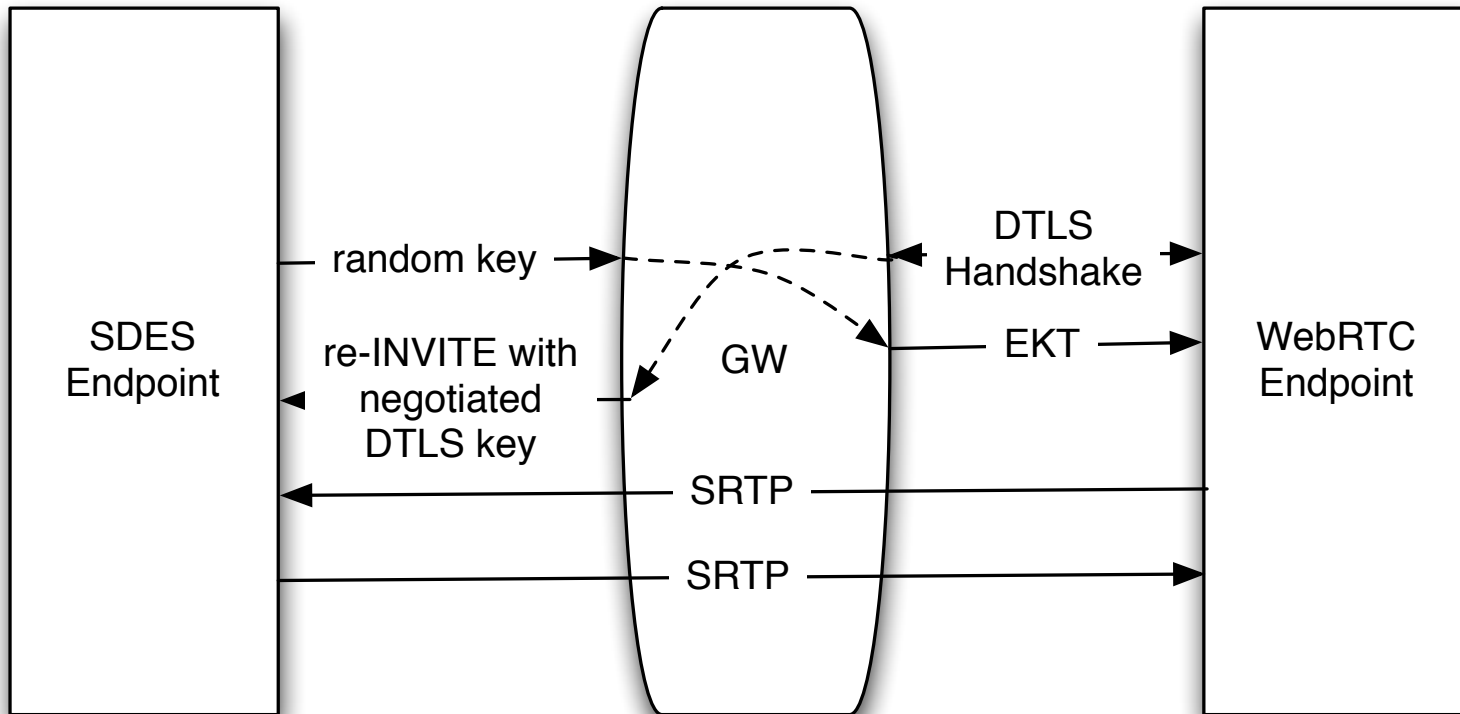
Basic Scenario



Reinvite for One-Way Media

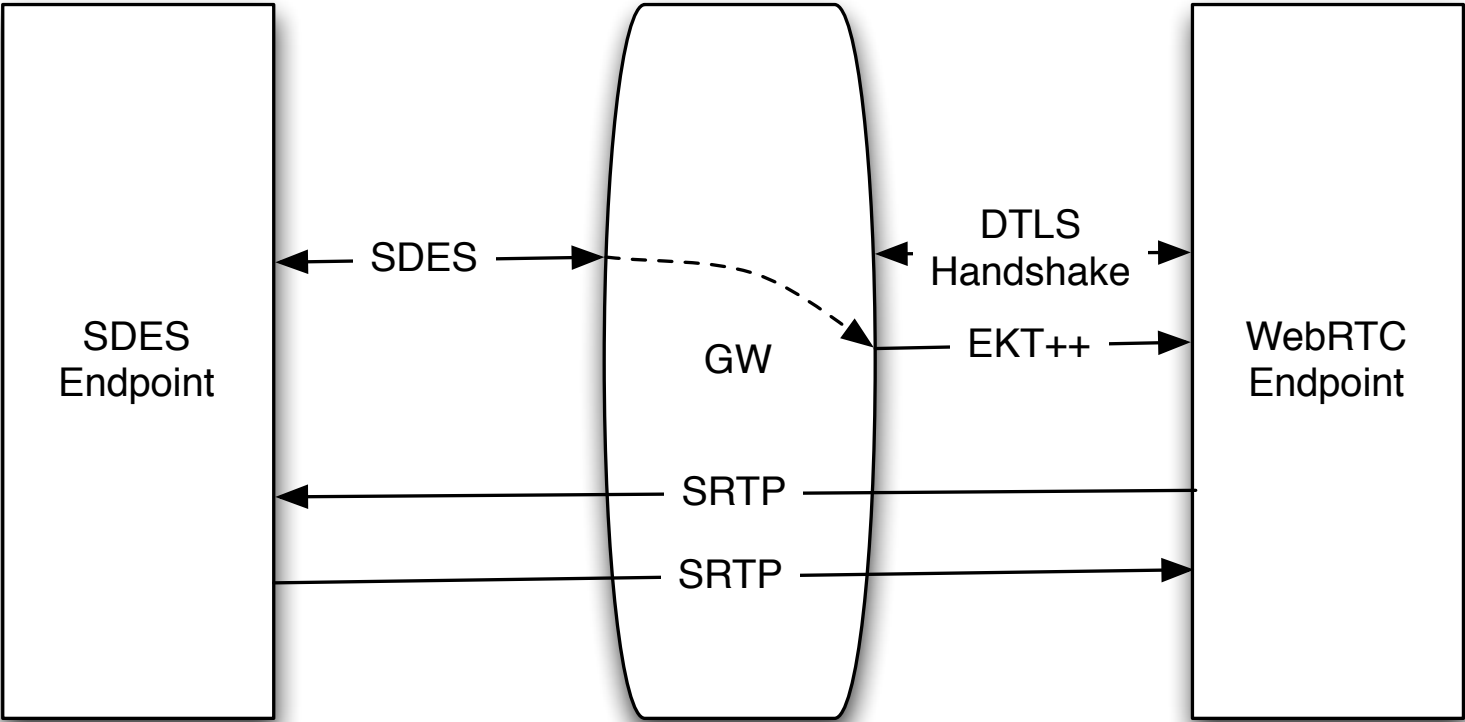


With EKT



P.S. This also works for videoconferencing

With Two-Way Key Push



Why does it matter what we allow: Incompetence

- DTLS is already going to be mandatory
 - So why shouldn't SDES be allowed?
- Because people will use it
 - Even if it means overriding defaults
 - We *know* people do stupid stuff
 - ... and someone might tell them it's faster/easier, etc.
- And the problem is that SDES is so brittle
 - Do we really believe people will remember to sanitize their logs?

- Let's not give people the tools to shoot themselves in the foot

Why does it matter what we allow: Malice

- If we allow SDES, negotiation will be in the SDP
- This allows for a trivial bid-down attack
 - Just pull out the fingerprint
 - ... or set the flag or whatever
- This is what you do if you want to enable monitoring
- Not possible to distinguish from
 - Laziness
 - People who want to be faster
 - Other client doesn't support DTLS
- Isolated streams + DTLS-only protect against this

They say nobody will notice if you change the JS...

Google Hangouts testing WebRTC-based, pluginless implementation?

A sharp-eyed [Toby Allen](#) recently brought the following code to my attention:

```
Qg.prototype.init=function(a,b,c,d){this.ca("pil");var
e=window.location.href.match(/.*[?&]mods=(^[^&]*)\.*/);if(e=
(e==m||2>e.length?0:/\bpluginless\b/.test(e[1]))||Z(S.Xd)){t:
{var e=new
Ad(Uc(this.e.l).location),f;f=e.K.get("lantern");if(f!=m&&
(f=Number(f),Ka(Og,f)){e=f;break t}!Fc||!
(0<=ta(dd,26))||webkitRTCPeerConnection==m?e=-1:(Pg.da())?
(f=Pg.get("mloo"),f=f!=m&&"true"==f):f=q,e=f?-
3:0==e.hb.lastIndexOf("/hangouts/_/present",0)?-4:1)}e=1==e)e?
Rg(this,q):Sg(this,a,b,c,d)};
```

That's an excerpt from the Google Hangouts javascript code. It's a bit obfuscated (either by design; or, more likely, because it's the output of another tool), and I haven't taken the time to fully dissect it. But the gist of the code appears to be to test for the presence of a "mods=pluginless" string in the URL; and, if one is present, to check whether the browser supports the use of WebRTC's RTCPeerConnection API (or, at least, Google's prefixed version of it). It then looks like it calls one of two different initialization functions based on whether such support is present.

Large scale monitoring

- Say you want to monitor a *lot* of people
 - First build a massive recording system...




Large Scale Monitoring of WebRTC

- SDES
 - Get a feed of keys from signaling server
 - Use existing traffic capture systems to record SRTP
- DTLS-SRTP
 - Reroute all traffic to your proxy
 - MITM every connection you want to monitor
 - This is not that easy to do
 - ... and not at all easy to hide
- One of these things is not like the other

Surely that would never happen...

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

HTTP Activity Client-to-Server



```
GET /search?tab=urdu&order=sortboth&q=musharraf&start=3&scope=urdu&link=next HTTP/1.1
Accept: */*
Referer: http://search.bbc.co.uk/search?tab=urdu&order=sortboth&q=musharraf&start=2&scope=urdu
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: search.bbc.co.uk
Cookie: BBC-UID=b479a5f4ad230a53063d513630203acb22684634a0e0b164c45f96efc054cf950Mozilla%2F4%2e0%20%28c
Cache-Control: max-stale=0
Connection: Keep-Alive
X-BlueCoat-Via: 66808702E9A98546
```

Search term:
Musharraf

Search on BBC

Host	URL Path	URL Args
search.bbc.co.uk	/search	tab=urdu&order=sortboth&q=musharraf&start=3&scope=urdu&link=next

Search Terms

Language	Browser	Via
musharraf	en	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
		66808702E9A98546

Referer

http://search.bbc.co.uk/search?tab=urdu&order=sortboth&q=musharraf&start=2&scope=urdu

Cookie

BBC-UID=b479a5f4ad230a53063d513630203acb22684634a0e0b164c45f96efc054cf950Mozilla%2F4%2e0%20%28c

Summary

- DTLS security properties range from somewhat better to much better
 - Doesn't make the logs a huge security risk
 - Possible to detect attacks even without identity
 - With identity/isolated streams, provides good security against the site
 - Much more resistant to large-scale monitoring
- Some legacy settings where SDES makes stuff easier
 - But not that much easier
 - And the advantage is shrinking not growing
- If we allow SDES some people will use it routinely

- And screw it up
- Hard to distinguish malice from simple laziness
- Better to just have a single secure method
- Proposed Resolution: Browser-based WebRTC implementations MUST NOT implement SDES

Questions?