Signed Public Key and Challenge

Graham Leggett
https://datatracker.ietf.org/doc/draft-leggett-spkac/01/
What is SPKAC?

- An ASN1 structure
- Contains a public key, and a challenge
- All signed by the private key of that public key
A long time ago, in a decade far far away

• In a land before time (mid nineteen nineties), a company called Netscape invented the keygen HTML tag.

• This keygen tag allowed a browser to prove possession of a locally generated private key, and invite a certificate authority to issue a certificate to match that private key.

• The private key was generated in the browser, and never left the end user’s possession.

• Keygen eventually became part of the HTML5 specification.
The Empire Strikes Back

• In the sequel to the keygen tag, a company that existed in the 1990s called Microsoft invented a different mechanism.

• Their mechanism was based on a Certificate Sign Request, but had the same overall property:

• The private key was generated in the browser, and never left the end user’s possession.

• To this day, available in Microsoft Edge in “Internet Explorer Compatibility Mode”.
Firefox, Google Said Yeah, Nah

• In a later episode in the saga, Firefox and Google said “yeah, nah” and removed keygen from the HTML5 specification.

• Part of the justification - and fairly so - is that the SPKAC message format that keygen tags used to prove possession was not defined or standardised.

• Another part of the justification - and again fairly so - was that the keygen tag (but not SPKAC) mandated the use of obsolete MD5.

• We propose an alternative approach, standardise the SPKAC message format at the IETF, and use the standard to fix any code implementation where MD5 was hard coded.
Why bother?

• Code exists right now, today.

• That code is approaching three decades old, it is widely tested, and is widely interoperable.

• Throwing out good code is waste.

• People still want to prove they possess private keys.

• Not all private keys relate to certificates (DKIM, etc).
Goals

• Formally define the SPKAC message format as a standard at the IETF.

• Update implementations of the standard, such as those at OpenSSL and Bouncycastle, to clearly show that they follow a standard.

• Update any implementations of the standard where the MD5 message digest is hard coded (OpenSSL fixed).

• Allow people to use the SPKAC message to prove they are in possession of a private key.
Non Goals

• To conflate the SPKAC specification with the history of SPKAC and where it came from.

• Implementation details are important, however SPKAC is a message format, and we don’t want to get bogged down.

• To change SPKAC in any way, it works fine as it is.
“Where next?”