EXTENDED ABSTRACT: BLOCKCHAIN VOIP AUTHENTICATION OF TEXT-TO-SPEECH CONVERSATIONS

This work has been presented at Workshop on Blockchain Horizons, ECIS, Portsmouth, UK June 25th 2018

Dantas, Rafael, Lero, University of Limerick, Limerick, Ireland, rafael.dantas@lero.ie
Exton, Chris, University of Limerick, Limerick, Ireland, chris.exton@ul.ie
Le Gear, Andrew, Horizon Globex Ltd., Limerick, Ireland, andrew.legear@horizon-globex.ie

Context

Even with advances in modern GSM technologies there remains situations where increased data compression rates can be valuable (Dantas, Exton, and Le Gear, 2018). While the adoption of 4G networks is increasing, there is still a significant portion of the world’s mobile devices connected through less capable 2G or 3G networks. There are also locations where regular cell phone connections are simply not viable. Satellite connections usually fulfil this demand, however due to the higher prices associated with such technologies, bandwidth is hardly a commodity.

We propose that with recent developments in speech-to-text technology, devices are now capable of correctly transcribing speech samples with enough precision to be used for telecommunications. By converting all speech data into text streams, it is possible to achieve significantly greater compression rates than current voice codecs are capable of as demonstrated by our initial experiments described by Table 1. Such text streams could also be further compressed using traditional text compression algorithms like Huffman encoding (Huffman, 1952).

While this solution could reduce the amount of excess data being transmitted to the internet, it could not by any means be considered lossless. Although speech samples convey more information than their raw text transcripts, there is a valid trade-off to be had between losing this information and the improved clarity and precision of the message.

A remaining concern, however, would be to authenticate the identify the person on the call. Where a text-to-speech generator is being used, the speaker’s voice, which is normally used by a listener to infer the identity of the speaker, is lost. To ensure the user’s security in this scenario, the application itself would need to implement an authentication protocol to verify both users’ identities before the call proceeds. While many authentication solutions exist, most would require the users to trust the entity providing the authentication service. We propose that Blockchain technology (Nakamoto, 2008) could bring authentication functionality to a text-to-speech compressed VoIP call that would allay these concerns. The use of a smart contract-based blockchain such as Ethereum (Buterin et al., 2014) would make it possible to provide independently verified identity usable across platforms.

<table>
<thead>
<tr>
<th>Message</th>
<th>PCM Size</th>
<th>SPEEX Size</th>
<th>ASCII Size</th>
<th>Huffman Coded Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is a recording</td>
<td>12353</td>
<td>991</td>
<td>19</td>
<td>12</td>
</tr>
<tr>
<td>The quick brown fox jumps over the lazy dog</td>
<td>24000</td>
<td>1698</td>
<td>43</td>
<td>26</td>
</tr>
<tr>
<td>Yes</td>
<td>8000</td>
<td>608</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>No</td>
<td>8000</td>
<td>624</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1. Comparison between PCM, Speex, raw text and Huffman coded text

Objective

Our research aims to provide an ultra-low bandwidth text-to-speech VoIP solution while also maintaining the security of the call by bringing smart contract-based authentication of all participants. This, we envisage, addresses the apparently incompatible concerns of compression and security in such a scenario.

Approach

Speech-to-text technology has now reached a level of maturity where it can be used for real time transcription.
A one second long, 8-bit PCM, 8 kHz sample results in a 64KB (or 8 KiB) file, if uncompressed. By encoding the same sample using Speex, it is possible to reduce data usage to approximately 4.8KB (or 600 B) depending on the parameters used (Dantas, Exton, and Le Gear, 2018). While this is a substantial improvement, the message could be further compressed using speech transcription as a voice encoding mechanism. Table 1 has four examples of compression rates made possible by converting speech into text.

Extra time spent on transcription can be tolerated if some rules could be imposed on the usage of the software, especially in the bandwidth sensitive scenarios mentioned above.

Our solution is implemented using SPHINX 4 and a Huffman encoder to implement the Speech-to-text transcription phase. This library makes possible the use of the Speech-to-Text technology without reliance on Cloud-based solutions. The resulting text is then converted into a byte stream to be sent through the internet. Finally, on the counterparty, the byte stream is converted back to its UTF-8 representation and FreeTTS takes care of the conversion from text back to speech.

As mentioned, where we gain with compression, we lose the natural means to identify the caller. To this end we bring an Ethereum-based authentication mechanism to validate their identities via a blockchain contract. This mechanism is implemented as a contract based on existing interfaces such as (Vogelsteller, 2017a) and (Vogelsteller, 2017b), which standardise the ideas of claims and identities and provide a framework to work with them.

Illustration

Table 1 contains a comparison between speech and textual representations of four different sentences. For the first column, we have a headerless, 8-bit mono PCM recorded at 8 kHz sampling rate. In the second column, the speech samples were converted into an 8-bit mono Speex encoded sample, at 8 kHz. The third and fourth columns contain sizes for the ASCII and Huffman-encoded representations. This particular example is sub-optimal, as all letters were given the same weight. Higher compression could be achieved by tuning the weights according to the English language.

For the authentication process, our implementation consumes ERC-725 and ERC-735 compatible contracts. The three functions signatures are:

- **issueClaim(address holder, address claimer, address publicKey) returns (bytes32 claim)** Registers the claim with the holder entity. This entity can be a third-party claim holder, if needed.

- **removeClaim(address holder, address claimer, address publicKey, bytes32 claim) returns (bool success)** Invalidates the claim. This allows applications to build safety mechanisms such as claim expiration after N uses or time passed.

- **checkClaim(address holder, address claimer, address publicKey, bytes32 claim) view returns (bool valid)** Checks if the identity of the claimant validates against the identity recorded in said claim. It only reads from the blockchain, so it should only cost the gas price needed for the check.

This contract is published on the Kovan Ethereum test network with the contract address: 0xdd4dab1b03d4530228e4ab6a010a7dfc5c06f49c.

Conclusion

While the above is a work in progress, we can see it demonstrates a promising solution for the opposing requirements of compression versus security. We have adopted a smart contract standard, though not yet stable, that we believe will emerge as the de facto for identity management on the Blockchain. Future work will see us integrate this feature into an existing Mobile VoIP solution provided by our commercial partner, Horizon Globex Ltd., and objectively measure potential compromises in quality, versus competing solutions, using MOS scoring.

References


