```
        if trans[meth] == "reverse" then
            sig = string.reverse( sig )

        elseif trans[meth] == "slice" then
            sig = string.sub( sig, idx + 1 )

        elseif trans[meth] == "swap" then
            if idx > 1 then
                sig = string.gsub( sig, "^(.)("..string.rep( ".", idx - 1 )..")(.)(.*)$", "%3%2%1%4" )
            elseif idx == 1 then
                sig = string.gsub( sig, "^(.)(.)", "%2%1" )
            end
        else
            vlc.msg.dbg("Couldn't apply unknown youtube video URL signature transformation")
            missing = true
        end
    end
    if missing then
        vlc.msg.err( "Couldn't process youtube video URL, please check for updates to this script" )
    end
    return sig
end

-- Parse and pick our video URL
function pick_url( url_map, fmt, js_url )
    local path = nil
    for stream in string.gmatch( url_map, "[^,]+" ) do
        -- Apparently formats are listed in quality order,
        -- so we can afford to simply take the first one
        local itag = string.match( stream, "itag=(%d+)" )
        if not fmt or not itag or tonumber( itag ) == tonumber( fmt ) then
            local url = string.match( stream, "url=([^&,]+)" )
            if url then
                url = vlc.strings.decode_uri( url )

                local sig = string.match( stream, "sig=([^&,]+)" )
                if not sig then
                    -- Scrambled signature
                    sig = string.match( stream, "s=([^&,]+)" )
                    if sig then
                        vlc.msg.info( "Found "..string.len( sig )."-character scrambled signature for youtube video URL, attempting to
descramble... " )
                        if js_url then
                            sig = js_descramble( sig, js_url )
```