Authorization via TLS

Welcome!

Simon Josefsson <simon@josefsson.org>
– Security advisor to OMIIEurope @ PDC/KTH

Middleware Security Group Meeting
Stockholm, 11 June 2007

Authorization via TLS

- Agenda
  - Authentication? Authorization?
  - Authorization Mechanisms
    - X.509 Attribute Certificates
    - SAML Assertions
  - The TLS-AUTHZ protocol
    - Protocol idea that may be applicable
  - Implementation in GnuTLS
  - Update on patent situation

Authentication?
Authorization?

- Authentication
  - Prove who you are.
  - Typically by proving something related to a digital identity.
- Authorization
  - Prove that you have access to some service.
  - Typically depends on that you have already proven who 'you' are (i.e., authentication).
- Implementation confusion
  - Often both steps are implemented by the same module. Generally not a good design, leads to confusion of the two concepts.

Authorization Mechanisms 1/2

- X.509 Attribute Certificates (“X.509AC”)
  - IETF RFC 3281.
  - Typically used together when X.509 Certificates are used for authentication.
  - May contain group membership, role, or other authorization information associated with the indicated AC holder.
  - Useless unless you know you are talking, over a secure channel, to the AC holder!

Authorization Mechanisms 2/2

- SAML Assertions (“SAMLAssert”)
  - OASIS
  - XML-based markup language

- Discussion: Are these authorization mechanisms sufficient?

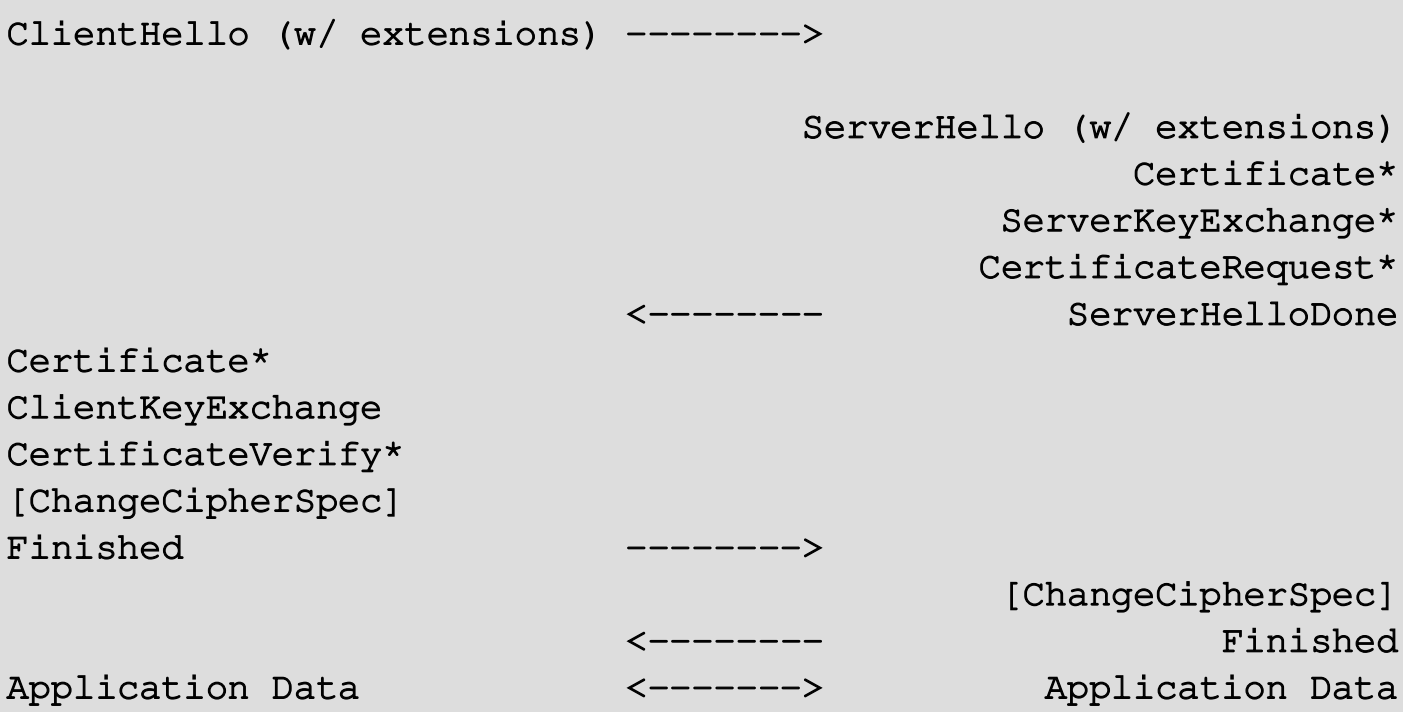
The TLS-AUTHZ Protocol

- Flexible authorization framework for TLS.
- Client and server negotiate the framework and the authorization method(s) to use
  - Allows X.509AC and SAMLAssert today, extensible typed-hole to add other authorization mechanisms for the future.
  - Supports BOTH X.509AC and SAMLAssert.
- Allows you to use X.509AC for authorization against one client, and SAMLAssert against another.
  - Allows simple transition between one technology to another.

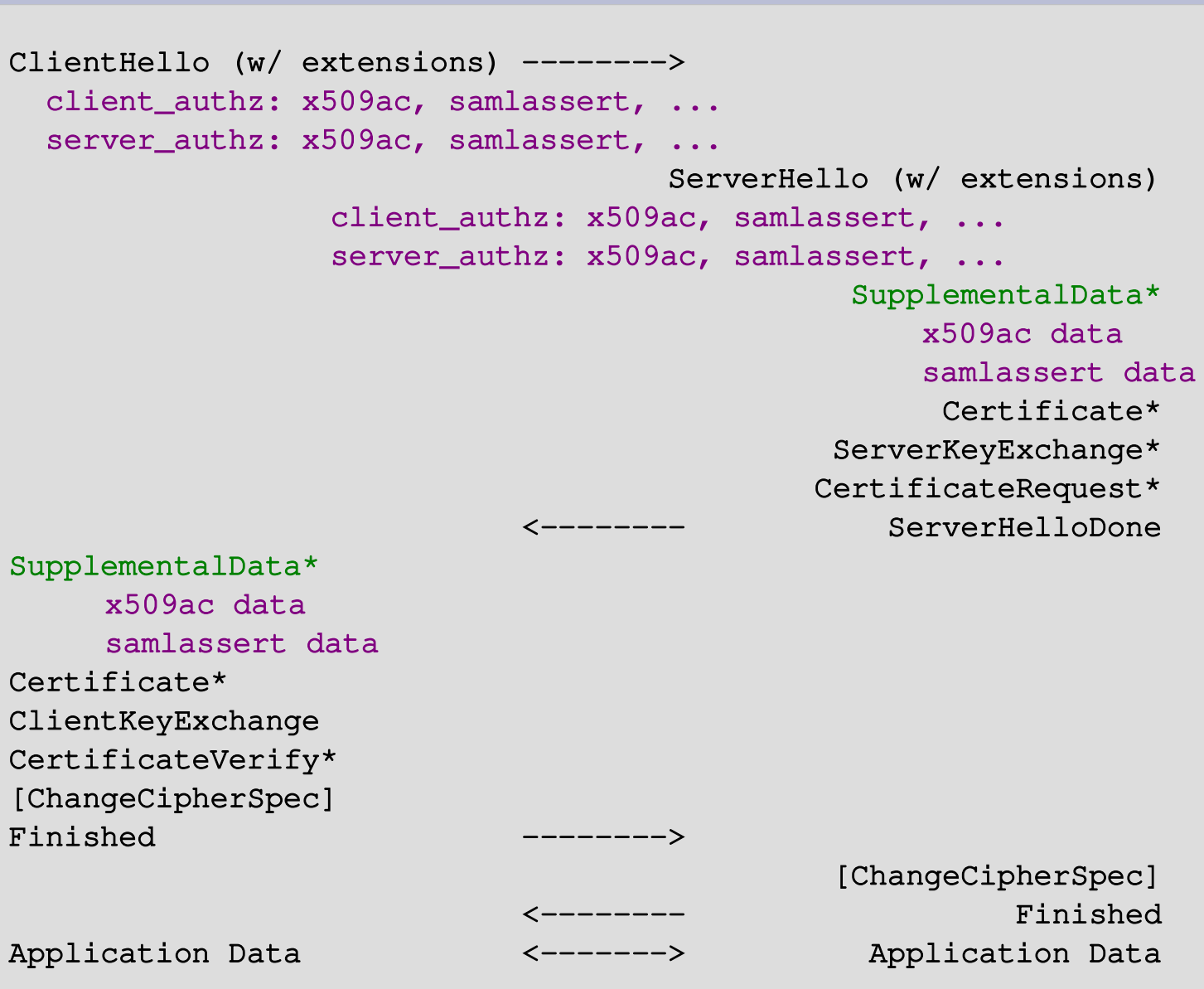
The TLS-AUTHZ Protocol

- Why?
  - Removes the need to specify a protocol on top of TLS to implement authorization services.
  - Standard method, supports any authorization framework.
  - Clearly separates authentication from authorization conceptually.

The TLS Protocol



The TLS-AUTHZ Protocol



Implementation in GnuTLS

```
/* Authorization extensions, RFC xxxx. */
typedef enum
  {
    GNUTLS_AUTHZ_X509_ATTR_CERT = 1,
    GNUTLS_AUTHZ_SAML_ASSERTION = 2,
    GNUTLS_AUTHZ_X509_ATTR_CERT_URL = 3,
    GNUTLS_AUTHZ_SAML_ASSERTION_URL = 4
  } gnutls_authz_data_format_type_t;

typedef int (*gnutls_authz_recv_callback_func) (gnutls_session_t session,
                                                const int *authz_formats, gnutls_datum_t *infos,
                                                const int *hashtypes, gnutls_datum_t *hash);
typedef int (*gnutls_authz_send_callback_func) (gnutls_session_t session,
                                        const int *client_formats, const int *server_formats);

void gnutls_authz_enable (gnutls_session_t session,
                          const int *client_formats, const int *server_formats,
                          gnutls_authz_recv_callback_func recv_callback,
                          gnutls_authz_send_callback_func send_callback);

int gnutls_authz_send_x509_attr_cert (gnutls_session_t session,
                                      const char *data, size_t len);
int gnutls_authz_send_saml_assertion (gnutls_session_t session,
                                      const char *data, size_t len);
```

Legal trouble

- Unfortunately, there is a patent application that covers these authorization ideas.

- Hopefully the application will not be approved.
- The owner has a 'patent license' on file with the IETF that gives you some rights if you abandon other rights.
  - Double and triple check with a lawyer before signing anything!

Standardization trouble

- The draft may not get published via the IETF due to the legal troubles.
- ..however, there is prior art: Stephen Farrell proposed draft-ietf-tls-attr-cert in 1998.

Way forward

- Discussion: Do you think the protocol is useful? We can propose a new document based on Stephen Farrel's older protocol.

The End

- Thank you for listening!

- Comments or questions:
  - Simon Josefsson <simon@josefsson.org>