
Secure Password-Based Authenticated Key Exchange for Web Services

Liang Fang
Indiana Univ.

Olivier Chevassaut
LBNL

Samuel Meder
Univ. of Chicago

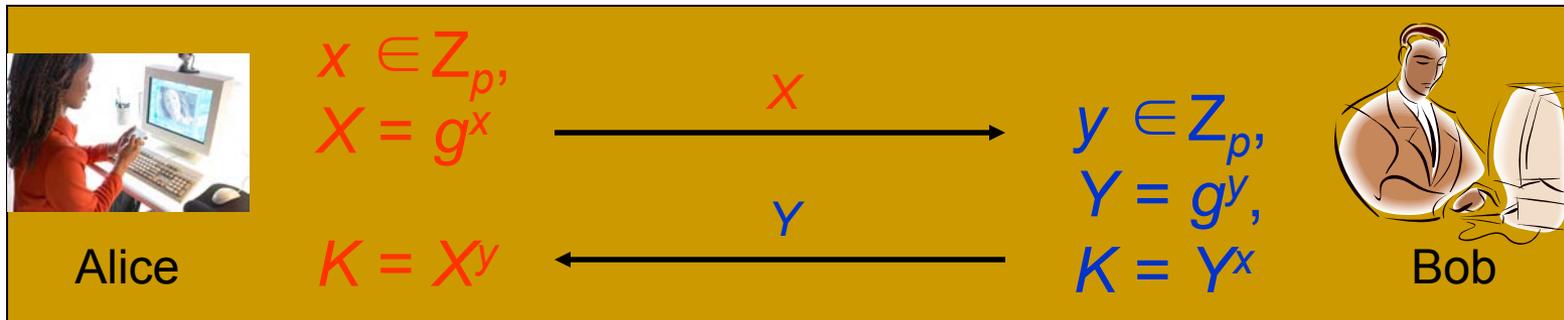
Frank Siebenlist
ANL

Summary

- Background story
 - The Contribution – brought password-based authenticated key exchange (AKE) protocols into Web services (WS)
 - Secure Password-based AKE for WS
 - Password-based AKE protocols
 - Security in Web services: WS-Trust and WS-SecureConversation
 - Implementation and integration issues
 - Conclusion and future works
-

Key Exchange Protocol

- Diffie-Hellman Key Exchange
 - $G = \langle g \rangle$, cyclic group of prime order p .



- Session key generated without being passed over the wire
- Subject to Man-in-the-middle Attack --
Authentication is needed against active adversaries

Authenticated Key Exchange (AKE)

- EKE proposed by Bellovin and Merritt in 92'
- Since then, There have been a family of variations, including AuthA, which led to their standardization under IEEE P1363 group
- Authentication Methods
 - Password: Short, low entropy common secret
 - Symmetric: Long, high entropy common secret
 - Asymmetric: (sk_A, pk_A) and possibly (sk_B, pk_B)

The AuthA Protocol



Alice (A)

Password pw



Bob (B)

$$x \in \mathbb{Z}_p, X = g^x$$

A, X

$$y \in \mathbb{Z}_p, Y = g^y$$
$$K = X^y$$

$$Y = D_{pw}(Y'), K = Y^x$$

$B, Y' = E_{pw}(Y)$ *AuthB*

$$AuthB = H_1(A, B, K)$$

Is *AuthB* correct?

One-flow encrypted key exchange

AuthA

Is *AuthA* correct?

$$AuthA = H_2(A, B, K, Y^{pw})$$

$$sk = H(A, B, X, Y, K)$$

The AuthA Protocol (cont.)

- Proven-secure Features
 - One-flow encrypted key exchange
 - Unilateral authentication (UA) and mutual authentication (MA)
 - Security against dictionary attacks
 - Semantic security
 - Forward secrecy
-

Summary

- The Contribution – brought password-based authenticated key exchange (AKE) protocols into Web services
 - Topics Coverage
 - Password-based AKE protocols
 - Security in Web services: WS-Trust and WS-SecureConversation
 - Implementation and integration issues
 - Conclusion and future works
-

Web Services Security

- Web Services Security
 - A series of emerging XML-based security standards from W3C and OASIS for SOAP-based Web services.
 - Core specs: XML Signature/Encryption, WS-Security, also called message level security.
 - Authentication
 - Integrity
 - Confidentiality
-

WS-Trust

- Extension to WS-Security
 - Define syntax for security token exchanges to build up trust relationship across Web service domains.
 - How to parse the security token (together with WS-SecurityPolicy)
 - How to trust the security token
 - Frameworks
 - Negotiation and Challenge framework
 - Security token service (broker) framework
 - <RequestSecurityToken> and <RequestSecurityTokenResponse> methods
-

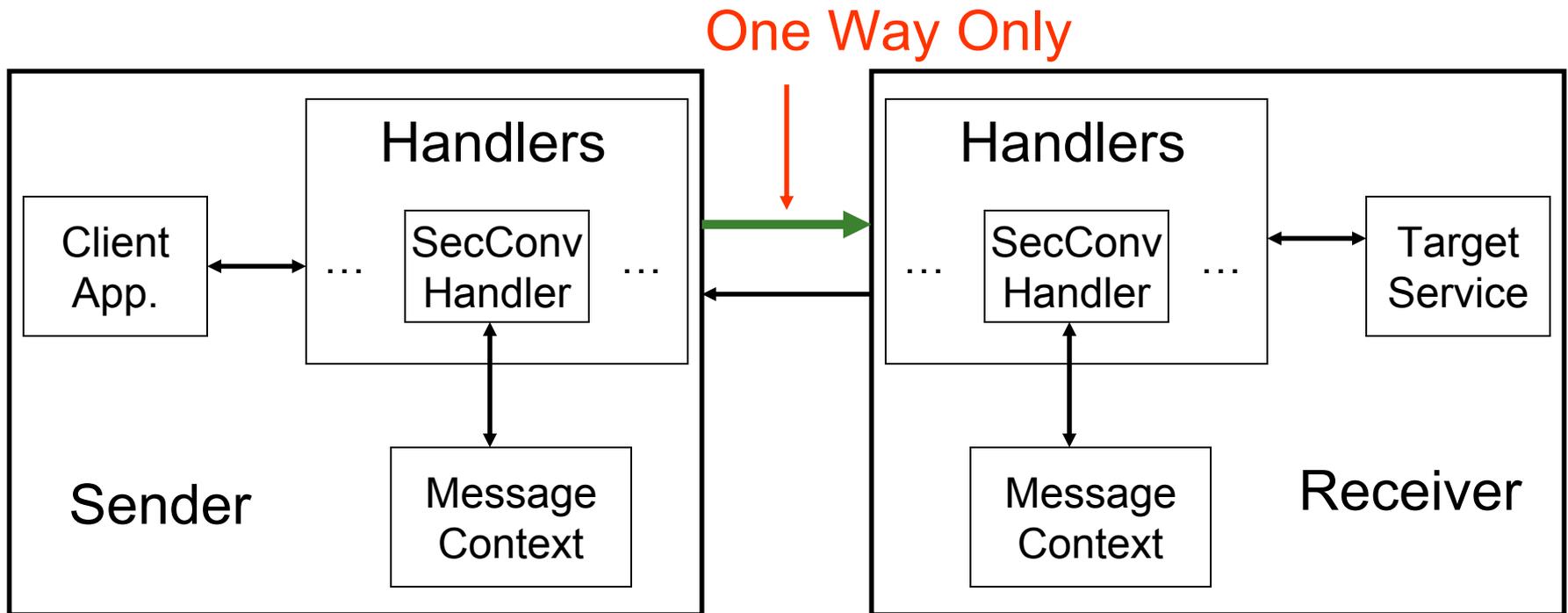
WS-SecureConversation

- Extension to WS-Security and WS-Trust
 - Allow security context establishment, sharing, and session key derivation
 - TLS-like message level solution for efficient multiple-message communication
 - Password-based AKE was integrated into TLS by Steiner et al in 2000.
 - WS-Trust and WS-SecureConversation are open for any key exchange protocols
 - Public key: expensive
 - Password-based AKE
 - <SecurityContextToken>
-

Summary

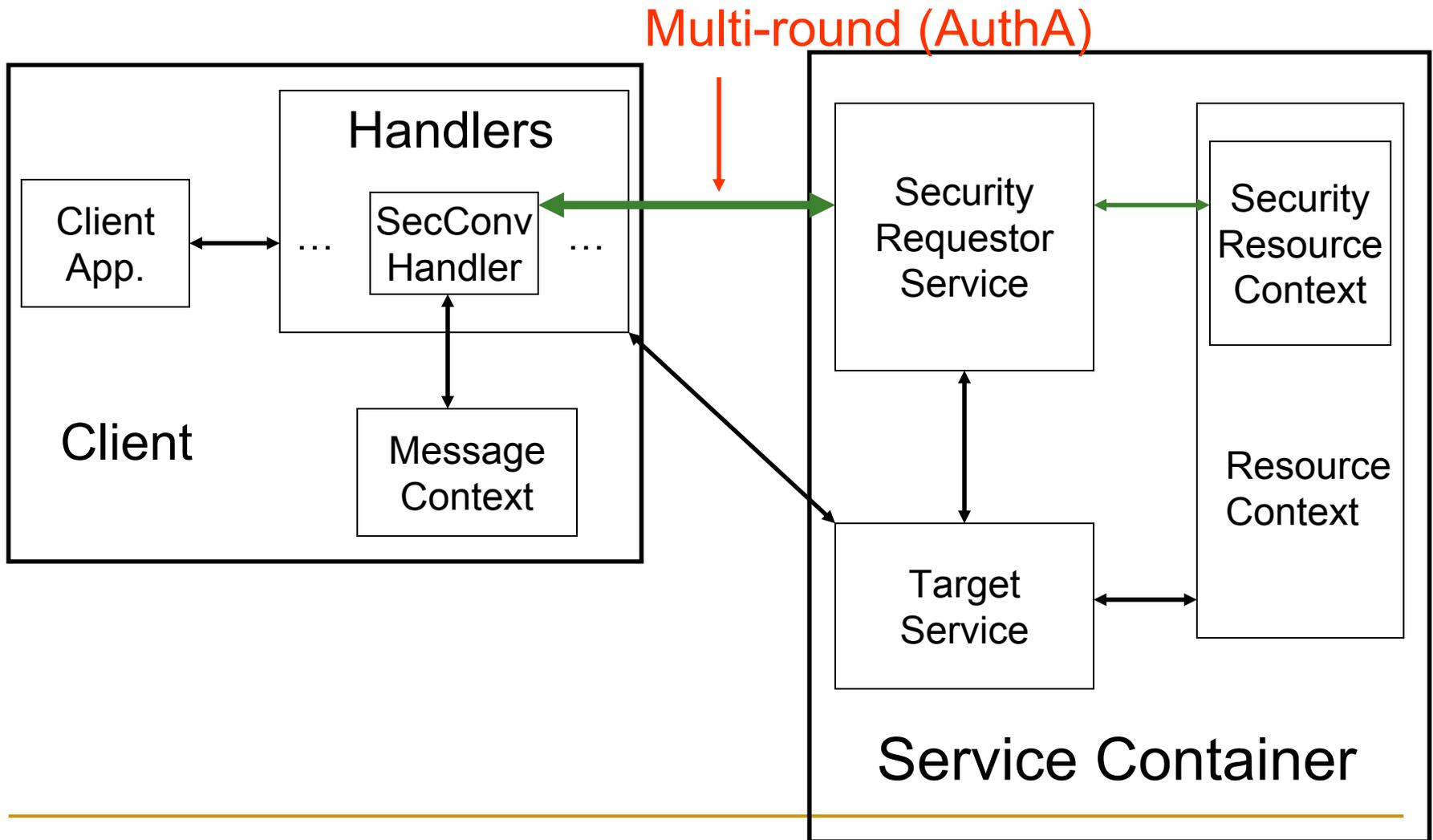
- The Contribution – brought password-based authenticated key exchange (AKE) protocols into Web services
 - Topics Coverage
 - Password-based AKE protocols
 - Security in Web services: WS-Trust and WS-SecureConversation
 - **Implementation and integration issues**
 - Conclusion and future works
-

WSS4J's Approach



Assumption: public keys previously known
Persistent support: backend database

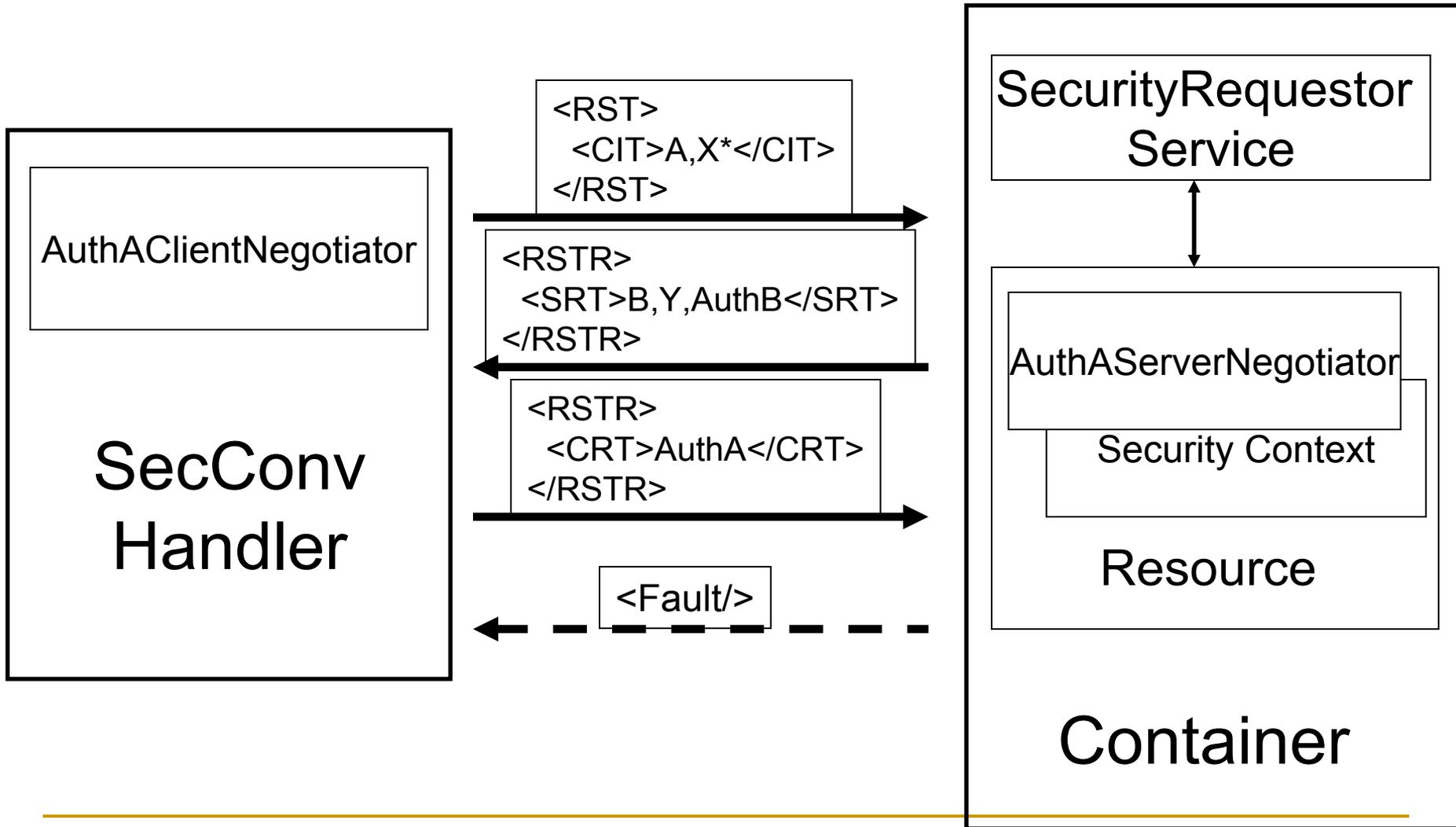
Our (Globus) Solution



Implementation Work

- The AuthA protocol
 - Implementation
 - Formalized in XML schema
 - SecurityRequestor Service in Globus container
 - Security Requestor service for multi-round session key exchange protocols, including AuthA
 - Same trust domain as the target service
 - Security context handled by Web Services Resource Framework (WSRF)
 - IBM initiated spec for stateful Web services
 - “Resource” takes the responsibility of storing the stateful information
 - It might have persistent support, but it is transparent to the services
-

Integration



Conclusion and Future Works

- Implemented Password-based AKE protocols for message level security
 - Future works
 - Optimization work
 - One time password (OTP)
 - Resistance mechanism in AuthA against Denial of Service (DoS) attacks
-