An Efficient Traceable Attribute-based Authentication Scheme with One-time Attribute Trees

Huihui Yang, Vladimir Oleshchuk

University of Agder, Norway

October 20, 2015

Overview

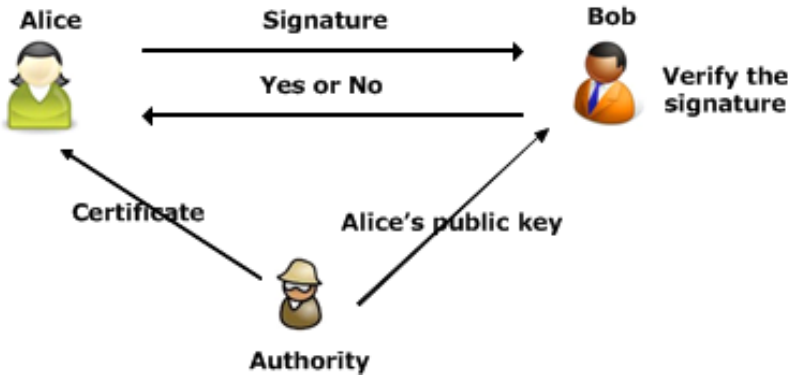
1. Motivations
2. Background Knowledge
   - Attribute-based authentication (ABA)
   - Two types of ABA Schemes
   - Attribute tree
3. The Proposed Scheme
   - ABA workflow
   - System setup
   - Signature generation, verification and opening
4. Conclusions

Motivations

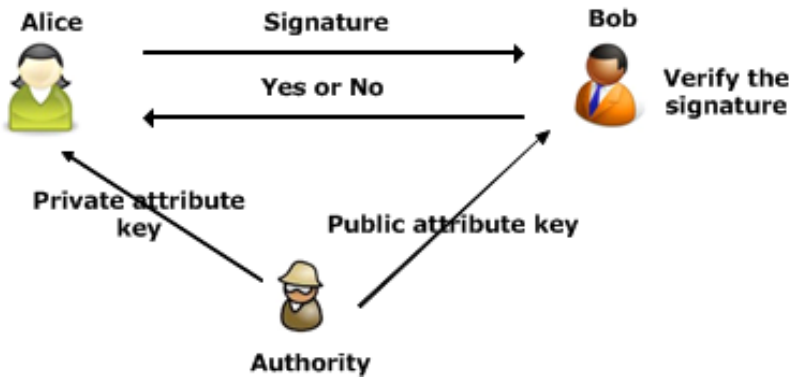
- Anonymous authentication: users may want to be authenticated without revealing their identity information to the authenticator.
- More flexibility and less cost: save cost for attribute tree and attribute key regeneration when attribute requirements change.

ABA introduction

PKI-based authentication: one public key, one private key

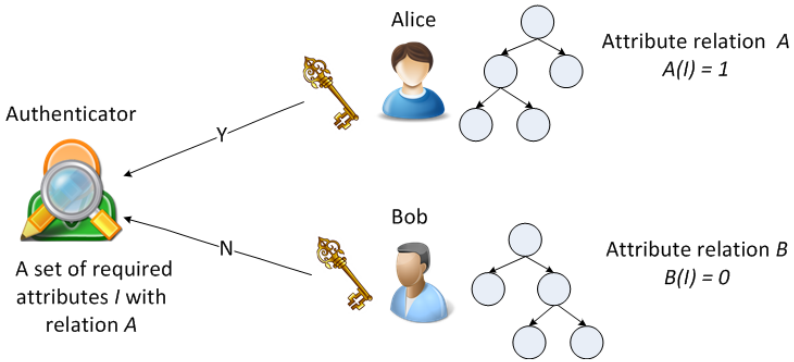


ABA: one public attribute key, one/multiple private attribute key(s)

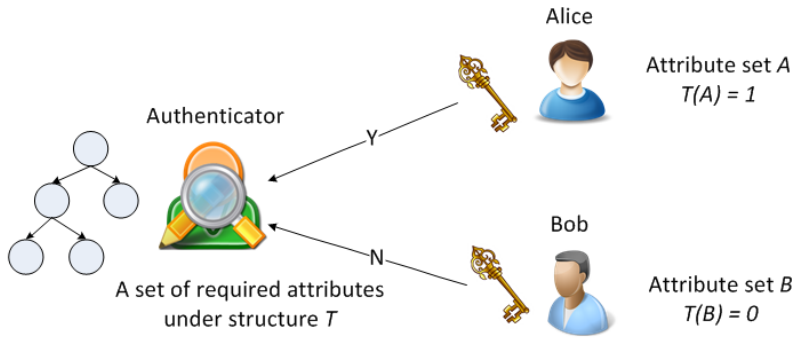


Two types of ABA schemes

Type 1:



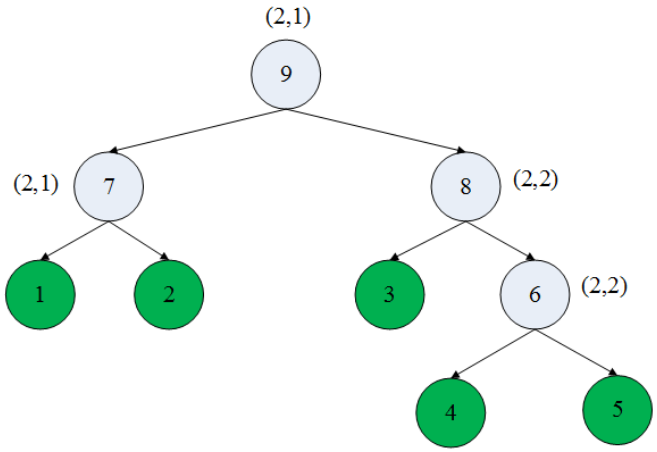
Type 2:



Attribute trees (1)

Represent logical "AND" & "OR".
Leaves represent attributes; interior nodes represent threshold.
Build related polynomials from top to down.
Interior nodes x are assigned with polynomials with degree $k_x - 1$; leaves with constant values.

1. Root: $q_{root}(0) = \alpha$.
2. Interior node x: $q_x(0) = q_{par(x)}(ind(x))$



Attribute trees (2)

Example:

$(att_1 \vee att_2) \vee (att_3 \wedge (att_4 \wedge att_5))$

$\alpha = 15 \quad q_9(0) = 15 \quad q_9(x) = 15$

$q_7(0) = q_9(7) = 15 \quad q_7(x) = 15$

$q_8(0) = q_9(8) = 15 \quad q_8(x) = x + 15$

$q_1(0) = 1_7(1) = 15$

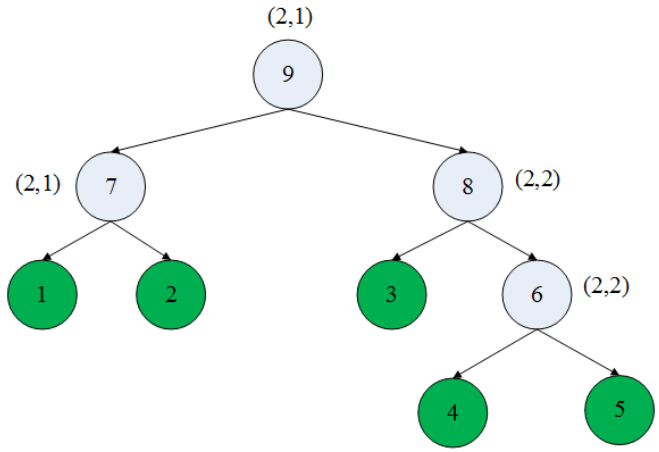
$q_2(0) = q_7(2) = 15$

$q_3(0) = q_8(3) = 18$

$q_6(0) = q_8(6) = 21 \quad q_6(x) = 2x + 21$

$q_4(0) = q_6(4) = 29$

$q_5(0) = q_6(5) = 31$



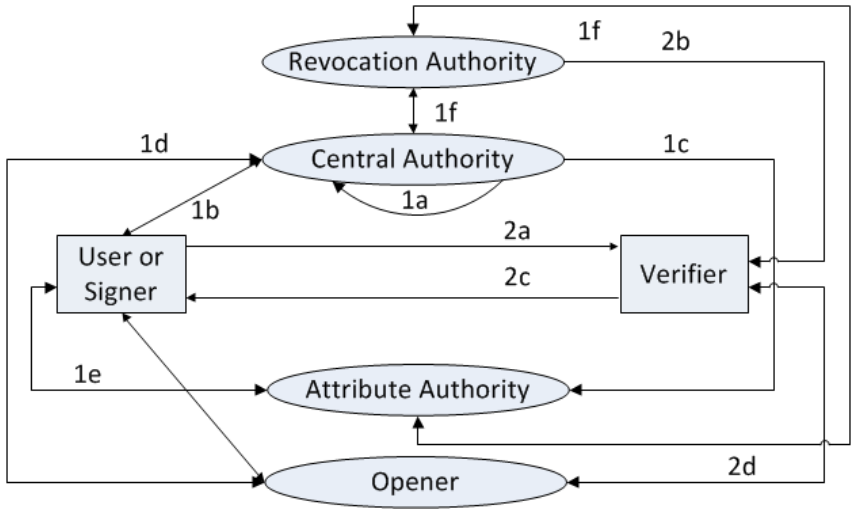
The Proposed Scheme

ABA workflow

- System setup: 1-a, 1-c, 1-d,
  - User key generation:1-b
  - Attribute key generation: 1-f
  - User attribute key generation: 1-e
- Signature generation and verification
  - Attribute tree generation
  - Signature generation: 2-a
  - Signature verification: 2-b, 2-c
  - Signature opening: 2-d

Main security requirements:

- Anonymity
- Traceability



The Proposed Scheme

System setup

- System public key set: $S_{pk} =< G, G_1, g, g_1, h, u, v, w_0 = g^{x_0} >$.
- System private key set: $S_{sk} =< x_0, tk >$, where $tk =< \xi_1, \xi_2 >$ is the tracking key.
- User U_i's key base: $< A_i, x_i >$, where $A_i = g^{1/(x_0+x_i)}$.
- System attribute set: $\Psi = \{att_1, \cdots, att_m\}$.
- System private attribute key set: $S_{Ask} =< t_1, \cdots, t_m >$.
- System public attribute key set: $S_{Apk} =< g^{t_1}, \cdots, g^{t_m} >$.
- User attribute key for attribute att_i: $T_{i_j} = g^{x_i} H(att_{i_j})^{t_{i_j}}$.

The Proposed Scheme

Signature generation, verification and opening

- Attribute selection and attribute tree generation:

$K_v = e(g, w_0)^{\alpha} = e(g, g)^{\alpha x_0}$

$\{\Gamma', g^{\alpha}, \forall y \in Leaf(\Gamma') : C_y, C'_y\}$, where $C_y = g^{q_y(0)}$ and $C'_y = H(y)^{q_y(0)}$.

- Signature generation:

Recover K_v as K_s from message $\{\Gamma', g^{\alpha}, \forall y \in Leaf(\Gamma') : C_y, C'_y\}$

Compute Signature $\sigma =< M, C_1, C_2, C_3, c, s_{\zeta}, s_{\beta}, s_{\alpha}, s_{\delta_1}, s_{\delta_2} >$.

- Signature verification:

Recover c as c' and check whether $c = c'$ holds.

- Signature opening:

$A_i = C_3/(C_1^{\varepsilon_1} C_2^{\varepsilon_2})$, using tracing key $tk =< \xi_1, \xi_2 >$.

Conclusions

- Proposed a traceable ABA scheme: flexible and efficient attribute tree generation
- The scheme provides both anonymity and traceability.
- Allow frequent attribute requirements change without increasing storage and computation costs.

Q&A?