

Efficient and Privacy-Preserving Enforcement of Attribute-Based Access Control

Ning Shang^{1,3} Federica Paci^{1,2} Elisa Bertino¹

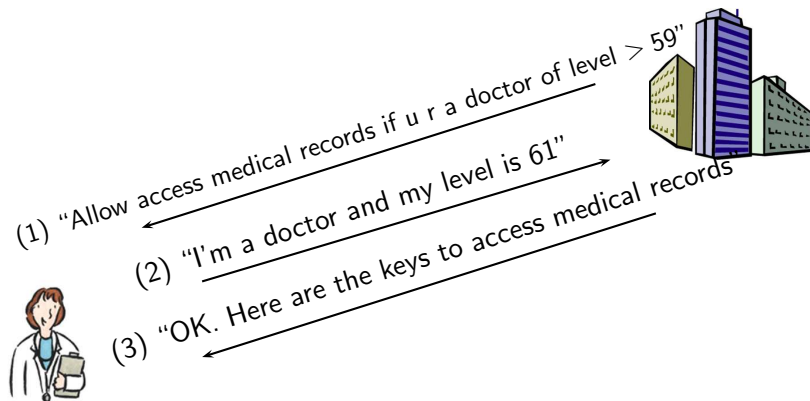
¹Purdue University, ²University of Trento, ³Microsoft

April, 2010

Attribute-based access control - Approach 0

Without privacy

Without privacy



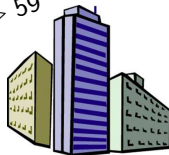
SP knows a lot about user's involved credentials

Attribute-based access control - Approach 1

Privacy-preserving via ZKPK

Privacy-preserving via ZKPK

- (1) "Allow access to medical records if u can prove u r a doctor of level > 59 "
- (2) Zero-knowledge proof protocols
- (3) "OK. Here are the keys to access medical records"




SP knows whether the user's credentials satisfy the requirements or not

Attribute-based access control - Approach 2

Privacy-preserving via OCBE

Privacy-preserving via OCBE

- 
- (1) "Allow access to medical records if u r a doctor of level > 59 "
 - (2) Commitments("I'm a doctor", "level =61")
 - (3) Envelope(keys to access medical records)



User can open the envelope iff its credentials satisfy the policy
SP does not know the outcome of envelope opening

OCBE Overview

OCBE: Oblivious Commitment-Based Envelope.¹

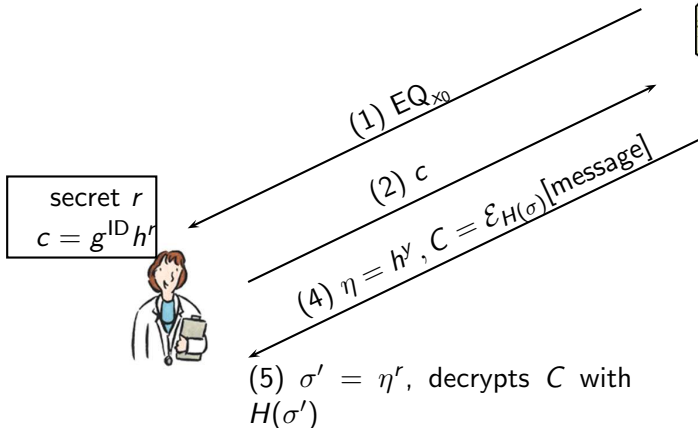
¹Jiangtao Li and Ninghui Li. OACerts: Oblivious attribute certificates.
IEEE Transactions on Dependable and Secure Computing, 3(4):340-352, 2006.

OCBE cryptographic building blocks

- $G = \langle g \rangle$: finite cyclic group of order p in which the computationally Diffie-Hellman problem is hard
- Pedersen commitment: $c = g^x h^r$, where $g, h \in G, r \xleftarrow{R} \mathbb{F}_p$
- \mathcal{E}_K : symmetric key encryption algorithm with key K
- $H(\cdot)$: cryptographic hash function

EQ-OCBE: equality predicate

Public Param = $\langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



Other OCBE's

GE-OCBE, LE-OCBE, ... are OCBE protocols for \geq, \leq, \dots predicates. They are performed in a similar fashion as EQ-OCBE, but generally more expensive.

OCBE features

- Security & privacy: the identity tokens (commitments) are unconditionally hiding and computationally binding
- X.509 integration: the identity tokens can be put into X.509v3 certificate extension fields

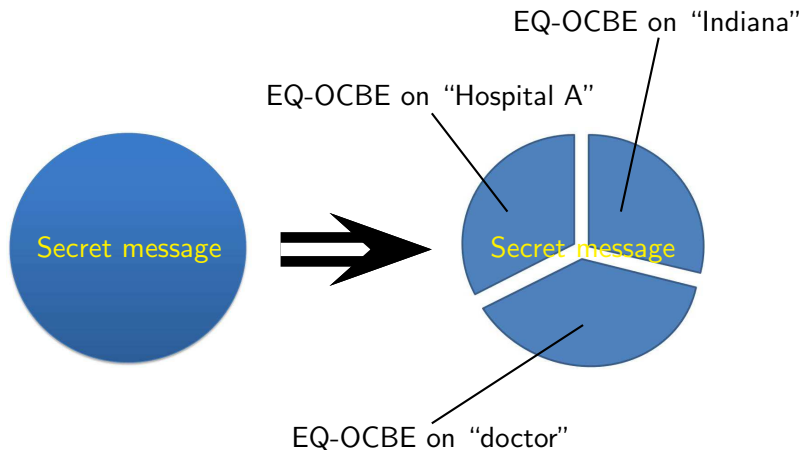
Multiple attributes specified in policy

Conjunction of conditions



“Allow access if you are a **doctor** of **Hospital A**
in **Indiana**”

Multiple attributes: a straightforward solution



This approach works, but...

It is not very efficient

communication and computation costs increase in proportion to the number of specified attributes

Question

Can we do better?

Answer

Agg-EQ-OCBE:

Aggregate OCBE protocol for equality predicates

- handles multiple equality conditions at the same time, without significantly increasing computational cost
- also requires less bandwidth

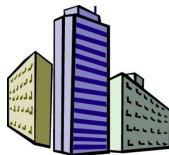
Agg-EQ-OCBE ideas

Techniques to improve the performance

- make use of the algebraic structure and operations in EQ-OCBE
- trade more expensive exponentiation operations for less costly addition and multiplication operations

Agg-EQ-OCBE illustration

Public Param = $\langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

(3) $y \xleftarrow{R} \mathbb{F}_p,$
 $c = c_1 \cdot c_2,$
 $x_0 = x_1 + x_2,$
 $\sigma = (cg^{-x_0})^y$

secret r_1, r_2
 $c_1 = g^{21} h^{r_1}$
 $c_2 = g^{35} h^{r_2}$



(5) $r = r_1 + r_2, \sigma' = \eta^r$, decrypts
 C with $H(\sigma')$ to get message

One problem



Collision

Owners of identity token sets

$$S_1 = \{c_1 = g^{21} h^{r_1}, c_2 = g^{35} h^{r_2}\} \text{ and } S_2 = \{c_3 = g^{18} h^{r_3}, c_4 = g^{38} h^{r_4}\}$$

will both open the envelope.

$$21 + 35 = 56 = 18 + 38$$

Solution

Cryptographic hash

Aggregate EQ-OCBE

Public Param = $\langle G, p, g, h \rangle, \mathcal{E}, H, H_1(\cdot)$



secret r_1, r_2
 $c_1 = g^{H_1(21)} h^{r_1}$
 $c_2 = g^{H_1(35)} h^{r_2}$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

(5) $r = r_1 + r_2, \sigma' = \eta^r$, decrypts
 C with $H(\sigma')$ to get message

(3) $y \xleftarrow{R} \mathbb{F}_p$,
 $c = c_1 \cdot c_2$,
 $x_0 = H_1(x_1) + H_1(x_2)$,
 $\sigma = (cg^{-x_0})^y$

Underlying intractability assumptions

- **Group 2nd-preimage resistant hash $\tilde{H}(\cdot)$**

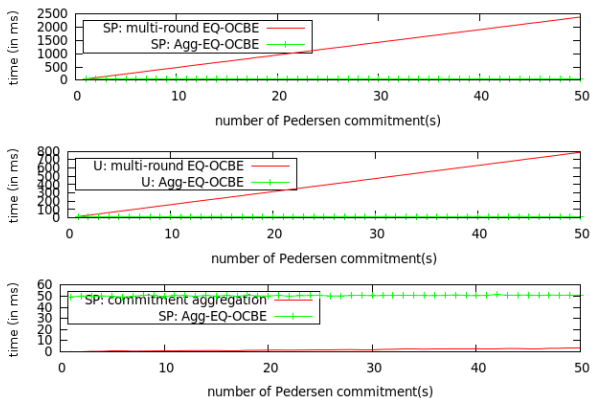
Given (x_1, \dots, x_m) , it is hard to find another tuple (y_1, \dots, y_n) such that

$$\sum_{i=1}^m \tilde{H}(x_i) = \sum_{i=1}^n \tilde{H}(y_i)$$

- **Computational Diffie-Hellman problem**

Given g^a, g^b , it is hard to compute g^{ab}

Experimental results



Future work

- **More application scenarios**
- **Aggregate GE-OCBE and other OCBE protocols**
aggregation works in certain cases, e.g., when sum of attribute values needs to be \geq a threshold value

Summary

- Privacy-preserving attribute-based access control concepts and approaches
- OCBE overview
- Aggregate EQ-OCBE
- Experimental data

The End

Thank you!

Questions?

`nshang@cs.purdue.edu`