More Research Directions on Digital Identity Management

The VeryIDX Team

Purdue University CS591C

October 21

Future work on DIM

Lots of research problems

- Treatment conditions for user credentials in an aggregate way
- Support for multiple registrars
  - Complexity
  - Efficiency
  - Security
- Credential delegation in VeryIDX
- And so on

Parties in communications

The VeryIDX framework.

- U - principal (user)
- SP - service provider
- R - registrar

Assumption:

- R is trusted
- The content of the service SVC provided by SP is encrypted using a symmetric algorithm with key SK.

Problem to solve

U makes a request for service SVC from SP. The service can be correctly received by U if and only if U's credential satisfies the condition Cond, specified in the policy of SP. In the same time, U's privacy is protected by not showing the details of the credential in clear.

Example: The SP, a food market, requires the customer U to present a receipt of previous purchase that was printed "on a Tuesday" and with "a value that is no less than $70.00", in order to offer a 10%-off discount. The customer U, holding a receipt printed "on last Tuesday" with a value of $85, wants to receive the discount, but does not want to let the store know either the date on the receipt, or its value.

Case Cond = "$x = x_0$"

Oblivious transfer protocol EQ-OCBE [Li and Li, 2006] can be used for VeryIDX
OCBE: Oblivious Commitment-Based Envelope

EQ-OCBE initialization

- SP and U agree on a symmetric encryption algorithm $\mathcal{E}$ and a cryptographic hash function $H(\cdot)$.
- R chooses parameters g, h from a group G of prime order p.
- After verifying the validity and ownership of a Pedersen commitment $M = g^x h^y$, R signs M and hands M together with the signature to U.
- U requests service SVC from SP and shows M and its signature signed by R.

Case Cond = "$x = x_0$" (cont'd)

EQ-OCBE communications

1. SP picks $z \in \mathbb{Z}_p^{\times}$, computes $\delta = (Mg^{-x_0})^z$, and then sends to U the pair $(\eta = h^z, C = \mathcal{E}_{H(\delta)}[SK])$.
2. Upon receiving (η, C) from SP, U computes $\delta' = \eta^y$, and decrypts C using symmetric encryption key $H(\delta')$.

If $x = x_0$, SK can be successfully recovered from C.

Example: case of equality

G: elliptic curve group $E(\mathbb{F}_q)$ order p (large); $g, h \in G$
$H(\cdot)$: SHA-1; $\mathcal{E}$: AES
Encode "STATE $=$ IN(14)" as "$x = 14$".

1. An Indiana resident U requests service from SP. SP sends its policy $\{\text{STATE} = \text{IN}(14)\}$ to U. After receiving the policy, U sends to SP its commitment $M = g^{14}h^{1234}$ signed by R. Note the value "1234" is known only to U.
2. SP picks random secret $z = 5678$, computes $\delta = (Mg^{-14})^z = (g^{14}h^{1234}g^{-14})^{5678} = (h^{1234})^{5678}$. SP sends to U the pair
$$(\eta = h^{5678}, C = \mathcal{E}_{H((h^{1234})^{5678})}[SK]).$$
3. U computes $\delta' = \eta^{1234} = h^{5678 \cdot 1234} = \delta$ and decrypts C using the key $H(\delta')$.

Multiple equality conditions: Agg-EQ-OCBE

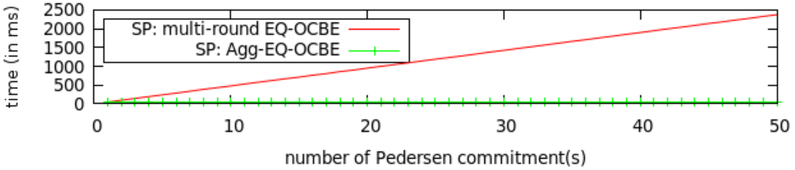
We also have extended the EQ-OCBE protocol to handle multiple equality conditions for user credentials in a secure and efficient way

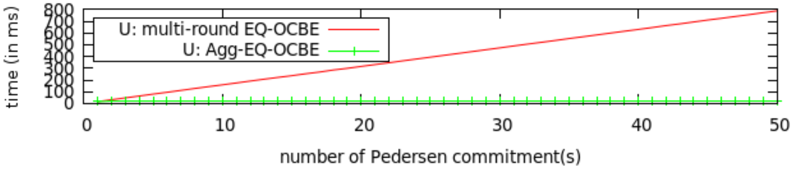
Agg-EQ-OCBE Protocol

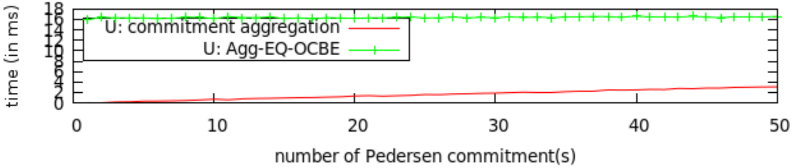
- Extension of EQ-OCBE
- Provably secure
- Much more efficient than multiple-round EQ-OCBE

Experimental Results: EQ-OCBE Vs. Agg-OCBE

Implementation with hyperelliptic curve cryptography [Shang, 2008].







Case Cond $= x \in [a, b]$

We can use the GE-OCBE and LE-OCBE protocols [Li and Li, 2006]

- Oblivious transfer protocols for inequality conditions
- Similar idea as EQ-OCBE
- More logically complex and computationally costly than EQ-OCBE
- So far no good way for aggregation

References I

Li, J. and Li, N. (2006).
Oacerts: Oblivious attribute certificates.
IEEE Transactions on Dependable and Secure Computing, 3(4):340–352.

Shang, N. (2008).
G2HEC: A Genus 2 Crypto C++ Library.
http://www.math.purdue.edu/~nshang/libg2hec.html.